

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

**Відокремлений структурний підрозділ
«Тернопільський фаховий коледж
Тернопільського національного технічного університету імені Івана Пулюя»
Відділення телекомунікацій та електронних систем
Циклова комісія комп'ютерних наук**

ПОЯСНЮВАЛЬНА ЗАПИСКА

**до кваліфікаційної роботи
фахового молодшого бакалавра
на тему: зробка інформаційної системи підбору та бронювання турів для
агенції «TourMaster»**

Виконав: студент IV курсу, групи КН-423
спеціальності: 122 Комп'ютерні науки

Валентин ГУСЄВ

Керівник **Руслан СЛОБОДЯН**

Рецензент _____
(ім'я та прізвище)

ВІДОКРЕМЛЕНИЙ СТРУКТУРНИЙ ПІДРОЗДІЛ
«ТЕРНОПІЛЬСЬКИЙ ФАХОВИЙ КОЛЕДЖ
ТЕРНОПІЛЬСЬКОГО НАЦІОНАЛЬНОГО ТЕХНІЧНОГО УНІВЕРСИТЕТУ
ІМЕНІ ІВАНА ПУЛЮЯ»

Відділення телекомунікацій та електронних систем
Циклова комісія комп'ютерних наук
Освітньо-професійний ступінь «фаховий молодший бакалавр»
Спеціальність 122 Комп'ютерні науки
Галузь знань 12 Інформаційні технології

ЗАТВЕРДЖУЮ

Голова циклової комісії
комп'ютерних наук

_____ Галина МАРЦІЯШ

« 02 » березня 2026 року

**З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ**

_____ Гусєву Валентину Ігоровичу

(прізвище, ім'я, по батькові)

1. Тема роботи: Розробка інформаційної системи підбору та бронювання турів для агенції «TourMaster»,

керівник роботи _____ Слободян Руслан Олесійович

(прізвище, ім'я, по батькові)

затверджені наказом вищого навчального закладу № 4/9-132 від 27.02.2026 р.

2. Строк подання студентом роботи: 19.06.2026 р.

3. Вихідні дані до роботи: технічне завдання на розробку програмного забезпечення, мови програмування: Java, TypeScript; фреймворки: Spring Boot, Angular, Tailwind CSS; стандарти: ДСТУ ISO/IEC/IEEE 29148:2025, ДСТУ ISO/IEC/IEEE 29119-1:2017, ДСТУ 3008:2015, ДСТУ 8302:2015.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити):

1 Загальний розділ

1.1 Аналітичний огляд існуючих рішень

1.2 Технічне завдання

1.2.1 Найменування та область застосування

1.2.2 Призначення розробки

1.2.3 Вимоги до функціоналу інформаційної системи

1.2.4 Вимоги до програмної документації

1.2.5 Техніко-економічні показники

1.2.6 Стадії та етапи розробки

- 1.2.7 Порядок тестування та прийому
- 2 Розробка технічного та робочого проєкту
 - 2.1 Розробка структури інформаційної системи
 - 2.2 Створення та верстка сторінок інформаційної системи
 - 2.3 Розробка структури бази даних інформаційної системи
 - 2.4 Програмування інформаційної системи
 - 2.4.1 Написання клієнтської частини
 - 2.4.2 Написання серверної частини
 - 2.5 Тестування інформаційної системи
- 3 Спеціальний розділ
 - 3.1 Інструкція з розміщення інформаційної системи в Інтернеті
 - 3.2 Інструкція з обслуговування та наповнення інформаційної системи
 - 3.3 Інструкція з популяризації та підтримки інформаційної системи
- 4 Економічний розділ
 - 4.1 Визначення стадій технологічного процесу та загальної тривалості проведення розробки інформаційної системи
 - 4.2 Визначення витрат на оплату праці та відрахувань на соціальні заходи
 - 4.3 Розрахунок витрат на електроенергію
 - 4.4 Розрахунок суми амортизаційних відрахувань
 - 4.5 Обчислення накладних витрат
 - 4.6 Складання кошторису витрат та визначення собівартості інформаційної системи
 - 4.7 Розрахунок ціни інформаційної системи
 - 4.8 Визначення економічної ефективності і терміну окупності капітальних вкладень
- 5 Охорона праці, техніка безпеки та екологічні вимоги
 - 5.1 Задачі, правові та організаційні основи охорони праці
 - 5.2 Категорії поділу фізичної роботи за величиною загальних енерговитрат
- 6 Висновки

Додаткові вказівки: виконання кваліфікаційної роботи із розробкою програмного продукту – інформаційної системи «TourMaster».

5. Перелік графічного матеріалу:

- 1. Схема структурна клієнтської частини
- 2. UML-діаграма варіантів використання
- 3. ER-діаграма бази даних
- 4. Таблиця техніко-економічних показників

6. Консультанти розділів роботи

Розділ	Ім'я, прізвище та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Економічний розділ	Любов КАЛУШКА		
Охорона праці, техніка безпеки та екологічні вимоги	Геннадій ГОРЯЧЕК		

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Отримання і аналіз технічного завдання	20.03.2026	
2	Збір і узагальнення інформації	01.05.2026	
3	Написання першого розділу	15.05.2026	
4	Розробка технічного та робочого проекту	29.05.2026	
5	Написання спеціального розділу	05.06.2026	
6	Розрахунок економічної частини	08.06.2026	
7	Написання розділу охорони праці	09.06.2026	
8	Виконання графічної частини	10.06.2026	
9	Оформлення пояснювальної записки	11.06.2026	
10	Погодження нормоконтролю	12.06.2026	
11	Попередній захист кваліфікаційної роботи	.06.2026	
12	Захист кваліфікаційної роботи	.06.2026	

7. Дата видачі завдання: 06 березня 2026 р.

Студент

_____ (підпис)

Валентин ГУССВ

Керівник роботи

_____ (підпис)

Руслан СЛОБОДЯН

ЗМІСТ

Анотація.....	7
Abstract.....	8
Вступ.....	9
1 Загальний розділ.....	11
1.1 Аналітичний огляд існуючих рішень.....	11
1.2 Технічне завдання.....	13
1.2.1 Найменування та область застосування.....	13
1.2.2 Призначення розробки.....	13
1.2.3 Вимоги до функціоналу інформаційної системи.....	14
1.2.4 Вимоги до програмної документації.....	16
1.2.5 Техніко-економічні показники.....	16
1.2.6 Стадії та етапи розробки.....	17
1.2.7 Порядок тестування та прийому.....	18
2 Розробка технічного та робочого проєкту.....	19
2.1 Розробка структури інформаційної системи.....	19
2.1.1 Структура сторінок клієнтського контуру.....	19
2.1.2 Структура сторінок адміністративного контуру.....	22
2.2 Створення та верстка сторінок інформаційної системи.....	23
2.3 Розробка структури бази даних інформаційної системи.....	30
2.4 Програмування інформаційної системи.....	41
2.4.1 Написання клієнтської частини.....	42
2.4.2 Написання серверної частини.....	46
2.5 Тестування інформаційної системи.....	50
3 Спеціальний розділ.....	59
3.1 Інструкція з розміщення інформаційної системи в Інтернеті.....	59
3.2 Інструкція з обслуговування та наповнення інформаційної системи.....	62
3.2.1 Вхід до адміністративної панелі.....	62
3.2.2 Управління туристичним каталогом.....	63

					2026.КВР.122.423.04.00.00 ПЗ			
Зм.	Арк.	№ докум.	Підпис	Дата				
Розроб.	Гусев В.І.				Розробка інформаційної системи підбору та бронювання турів для агенції «TourMaster» Пояснювальна записка	Літ.	Арк.	Аркуші
Перевір.	Слободян Р.О.						5	123
Реценз.						ВСП ТФК ТНТУ КН-423 м. Тернопіль		
Н. Контр.	Приймак В.А.							
Затверд.								

3.2.3	Обробка бронювань та управління статусами	65
3.2.5	Технічне обслуговування системи	67
3.3	Інструкція з популяризації та підтримки інформаційної системи.....	67
4	Економічний розділ	70
4.1	Визначення стадій технологічного процесу та загальної тривалості проведення НДР.....	70
4.2	Визначення витрат на оплату праці та відрахувань на соціальні заходи..	71
4.3	Розрахунок витрат на електроенергію	74
4.4	Розрахунок суми амортизаційних відрахувань розробки інформаційної системи «TourMaster»	74
4.5	Обчислення накладних витрат.....	75
4.6	Складання кошторису витрат та визначення собівартості інформаційної системи «TourMaster»	75
4.7	Розрахунок ціни інформаційної системи «TourMaster».....	76
4.8	Визначення економічної ефективності і терміну окупності капітальних вкладень	77
5	Охорона праці, техніка безпеки та екологічні вимоги	79
5.1	Задачі, правові та організаційні основи охорони праці	79
5.2	Категорії поділу фізичної роботи за величиною загальних енерговитрат	82
	Висновки.....	85
	Перелік посилань	87
	Додатки	89
	Додаток А. Лістинг файлу «auth.interceptor.ts»	89
	Додаток Б. Лістинг файлу «auth.service.ts»	91
	Додаток В. Лістинг файлу «tour-catalog.component.ts»	94
	Додаток Г. Лістинг файлу «booking-wizard.component.ts».....	100
	Додаток Д. Лістинг файлу «SecurityConfig.java»	109
	Додаток Е. Лістинг файлу «JwtService.java»	113
	Додаток Ж. Лістинг файлу «TourSpecification.java».....	116
	Додаток И. Контрольний список тестування вебсайту TourMaster	120

АНОТАЦІЯ

Кваліфікаційна робота присвячена розробці інформаційної системи підбору та бронювання турів для туристичної агенції «TourMaster». Обсяг пояснювальної записки становить 88 сторінок основного тексту, містить 21 таблицю, 11 рисунків та 8 додатків.

Об'єктом розробки є процеси підбору туристичних пропозицій і оформлення бронювань у туристичній агенції, що здійснювалися переважно в ручному або частково автоматизованому режимі. Предметом розробки є повнофункціональна інформаційна система у вигляді вебзастосунку, що автоматизує зазначені процеси.

Метою кваліфікаційної роботи є проектування та програмна реалізація інформаційної системи, що забезпечує клієнтам агентства можливість цілодобового онлайн-підбору туристичних пропозицій і їх бронювання через мережу Інтернет, а менеджерам — зручний адміністративний інтерфейс для управління каталогом турів, бронюваннями та клієнтською базою.

Пояснювальна записка складається з вступу, п'яти розділів, висновків, переліку посилань та додатків. У першому розділі проведено аналітичний огляд існуючих рішень та сформульовано технічне завдання. Другий розділ містить опис розробки технічного та робочого проекту: структуру системи, верстку сторінок, схему бази даних, програмування клієнтської та серверної частин і результати тестування. Третій розділ містить інструкції з розміщення системи в Інтернеті, її обслуговування та популяризації.

До складу кваліфікаційної роботи входить графічна частина, яка складається з структурної схеми інформаційної системи, діаграми варіантів використання, діаграми бази даних, а також техніко-економічних показників, що виконані на окремих аркушах формату А1.

					2026.КВР.122.423.04.00.00 ПЗ	Арк.
						7
Зм.	Арк.	№ докум.	Підпис	Дата		

ABSTRACT

The qualification work is dedicated to the development of an information system for tour selection and booking for the travel agency "TourMaster". The explanatory note comprises 88 pages of main text, contains 21 tables, 11 figures and 8 appendices.

The object of development is the processes of selecting tourist offers and processing bookings at a travel agency, which were carried out primarily in a manual or partially automated manner. The subject of development is a fully functional information system in the form of a web application that automates the aforementioned processes.

The purpose of the qualification work is the design and software implementation of an information system that provides agency clients with the ability to search for tourist offers and book them online around the clock via the Internet, and provides managers with a convenient administrative interface for managing the tour catalogue, bookings, and client database.

The explanatory note consists of an introduction, five sections, conclusions, a list of references and appendices. The first section presents an analytical review of existing solutions and formulates the technical specification. The second section describes the development of the technical and working design: the system structure, page layout, database schema, implementation of the client and server sides, and testing results. The third section contains instructions for deploying the system on the Internet, its maintenance and promotion.

The qualification work includes a graphical part comprising a structural scheme of the information system, a use case diagram, a database diagram, as well as technical and economic indicators, all produced on separate A1-format sheets.

					2026.KBP.122.423.04.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		8

ВСТУП

Туристична галузь є однією з найбільш динамічних у сфері цифрових перетворень: за даними Statista [1], частка онлайн-бронювань у загальному обсязі туристичних послуг зростає щорічно на 12–15 %, а понад 70 % мандрівників розпочинають пошук подорожей через веб-ресурси. В умовах такої тенденції туристичні агентства, що не мають власної інформаційної системи онлайн-підбору та бронювання, неминуче поступаються конкурентам як у залученні нових клієнтів, так і в ефективності внутрішніх операційних процесів.

Сучасний споживач туристичних послуг очікує можливості порівнювати пропозиції, перевіряти наявність місць і оформлювати бронювання у будь-який час без необхідності відвідувати офіс або телефонувати менеджеру. Водночас менеджери агентств потребують зручного інструменту для централізованого управління каталогом турів, швидкої обробки замовлень і аналізу ефективності продажів. Ці взаємопов'язані потреби визначають актуальність розробки комплексної інформаційної системи, що автоматизує обидва процеси в межах єдиної платформи.

Метою кваліфікаційної роботи є розробка інформаційної системи підбору та бронювання турів для агенції «TourMaster», що забезпечує повний цикл взаємодії між клієнтом і агентством — від пошуку туристичної пропозиції до підтвердження бронювання — з мінімальним залученням персоналу до рутинних операцій.

Для досягнення мети поставлено такі задачі:

- 1) провести аналіз існуючих рішень у сфері онлайн-бронювання туристичних послуг і визначити функціональні вимоги до розроблюваної системи;
- 2) спроектувати структуру сторінок системи та розробити діаграму варіантів використання;
- 3) розробити реляційну схему бази даних і реалізувати її міграції засобами Flyway;

					2026.КВР.122.423.04.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		9

4) реалізувати серверну частину системи на базі Spring Boot з REST API, Spring Security та JWT-аутентифікацією;

5) розробити клієнтський вебінтерфейс на Angular 22 з бібліотекою PrimeNG і адаптивним дизайном;

6) провести мануальне тестування реалізованих функціональних модулів і підтвердити відповідність системи вимогам технічного завдання;

7) підготувати інструкції з розміщення системи в Інтернеті, її наповнення та популяризації.

Об'єктом розробки є процеси підбору туристичних пропозицій та оформлення бронювань у туристичній агенції. Предметом розробки є інформаційна система у вигляді вебзастосунку з клієнтською та адміністративною частинами.

Практична цінність роботи полягає у тому, що розроблена система є готовим до розгортання програмним продуктом і може бути впроваджена туристичною агенцією «TourMaster» або іншою агенцією аналогічного профілю. Передбачене зниження витрат на обробку бронювань орієнтовно на 40 % завдяки автоматизації приймання, підтвердження та документування замовлень.

При виконанні кваліфікаційної роботи використано такі методи: аналіз і порівняння існуючих платформ, метод об'єктно-орієнтованого проєктування, UML-моделювання, реляційне проєктування баз даних, методи мануального тестування програмного забезпечення.

					2026.КВР.122.423.04.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		10

1 ЗАГАЛЬНИЙ РОЗДІЛ

1.1 Аналітичний огляд існуючих рішень

Стрімкий розвиток цифрових технологій суттєво трансформував туристичну галузь: сучасний споживач прагне самостійно обирати, порівнювати та бронювати тури безпосередньо через мережу Інтернет, не вдаючись до послуг офісних консультантів. За даними аналітичних досліджень Statista, частка онлайн-бронювань у загальному обсязі туристичних послуг щорічно зростає на 12–15 %, а понад 70 % мандрівників починають пошук подорожей саме з веб-ресурсів. В умовах такої тенденції туристичні агентства, що не мають повноцінної інформаційної системи онлайн-підбору та бронювання, неминуче втрачають конкурентні позиції. Для формування вимог до розроблюваної системи «TourMaster» було здійснено аналіз трьох провідних вітчизняних платформ туристичних послуг: TPG Ukraine, Join UP! та Coral Travel Ukraine.

Платформа TPG Ukraine (tpg.ua) є одним із найбільших агрегаторів туристичних пакетів на ринку України. Система забезпечує пошук турів за широким набором параметрів: країна призначення, дата вильоту, тривалість, тип харчування, категорія готелю та ціновий діапазон. Інтерфейс реалізований у вигляді односторінкового застосунку з динамічним оновленням результатів без перезавантаження сторінки. Серед ключових переваг платформи слід виокремити розвинений модуль порівняння готелів, інтеграцію з картографічним сервісом Google Maps для відображення розташування об'єктів розміщення, а також особистий кабінет клієнта з історією бронювань. Водночас система має суттєвий недолік: адміністративна панель для менеджерів агентства не є інтегрованою з клієнтською частиною, що ускладнює оперативне управління контентом і цінами. Відсутній також модуль онлайн-чату та автоматизованих сповіщень про статус бронювання.

Система Join UP! (joinup.ua) орієнтована переважно на масовий туристичний продукт — пляжний відпочинок та екскурсійні тури до популярних

					2026.КВР.122.423.04.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		11

напрямоків Туреччини, Єгипту, ОАЕ та Греції. Платформа вирізняється зручним механізмом швидкого пошуку з відображенням мінімальної ціни на календарі вильотів — так званого «гарячого тарифу». Реалізовано функцію попереднього розрахунку вартості туру для різних категорій туристів (дорослі, діти) та можливість онлайн-оплати через платіжний шлюз. Проте функціональність системи обмежена виключно пакетними турами від власного туроператора, що унеможливорює гнучке додавання сторонніх пропозицій. Бек-офіс менеджера реалізовано у вигляді окремої застарілої ERP-системи без сучасного веб-інтерфейсу, що суттєво знижує зручність роботи персоналу. Відсутня система відгуків клієнтів та рейтингів готелів, заснованих на власній базі даних агентства.

Веб-ресурс Coral Travel Ukraine (coral.ua) представляє великого міжнародного туроператора з розгалуженою мережею партнерів. Сайт пропонує широкий асортимент турів, круїзів і авіаперельотів із детальними описами програм та фотогалереями. Платформа підтримує багатомовний інтерфейс і адаптована для мобільних пристроїв. Серед функціональних переваг — інтегрована система страхування, можливість вибору конкретного номера в готелі на основі планів поверхів, а також розвинений розділ підказок мандрівникам. Однак, як і в попередньому випадку, система орієнтована виключно на власний туроператорський контент, не передбачає механізмів для незалежного туристичного агентства і позбавлена гнучкого адміністративного модуля.

Проведений аналіз дозволяє зробити висновок, що жодна з розглянутих платформ не відповідає повною мірою потребам невеликого або середнього туристичного агентства, яке прагне мати єдину інтегровану систему для роботи як із клієнтами, так і з внутрішніми бізнес-процесами.

Виявлені прогалини визначають функціональні вимоги до розроблюваної інформаційної системи «TourMaster»: наявність зручного веб-інтерфейсу для клієнтів з повноцінним пошуком і бронюванням, інтегрований бек-офіс для менеджерів агентства, система сповіщень та підтримка мультиоператорного контенту.

					2026.КВР.122.423.04.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		12

1.2 Технічне завдання

1.2.1 Найменування та область застосування

Повне найменування програмного виробу: «Інформаційна система підбору та бронювання турів для агенції «TourMaster»». Скорочена назва — ІС TourMaster.

Система призначена для використання туристичними агентствами, що здійснюють підбір, оформлення та продаж туристичних пакетів фізичним особам. Об'єктом впровадження є туристична агенція «TourMaster», що надає послуги організації закордонного та внутрішнього туризму. Система функціонує як веб-застосунок, доступний через стандартний браузер на персональних комп'ютерах, планшетних пристроях і смартфонах без необхідності встановлення додаткового програмного забезпечення на стороні користувача.

Галузь застосування охоплює сферу туристичних і суміжних послуг: організацію групових і індивідуальних турів, бронювання готелів, оформлення страхових полісів, управління базою клієнтів агентства та формування оперативної звітності для керівництва.

1.2.2 Призначення розробки

Експлуатаційним призначенням ІС TourMaster є забезпечення можливості цілодобового онлайн-підбору туристичних пропозицій та їх бронювання клієнтами агентства через мережу Інтернет, а також автоматизація внутрішніх операційних процесів персоналу агентства — від прийому заявок до формування документів і контролю оплат.

Функціональним призначенням системи є реалізація комплексу таких можливостей: надання клієнтам зручного інструменту пошуку турів за множиною критеріїв; забезпечення процесу бронювання в кілька кроків із підтвердженням у режимі реального часу; ведення особистого кабінету клієнта з доступом до

					2026.КВР.122.423.04.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		13

поточних і завершених бронювань; надання менеджерам агентства адміністративного інтерфейсу для управління каталогом турів, обробки замовлень, комунікації з клієнтами та формування аналітичних звітів.

1.2.3 Вимоги до функціоналу інформаційної системи

Вимоги до функціональних характеристик системи визначаються двома основними ролями користувачів — клієнтом агентства та менеджером агентства — і охоплюють весь цикл взаємодії від першого відвідування сайту до завершення туру.

Вхідні дані системи. Вхідними даними для клієнтської частини є: параметри пошуку туру (країна, місто вильоту, дати, кількість туристів, категорія готелю, тип харчування, бюджет); реєстраційні дані користувача (ім'я, прізвище, електронна адреса, номер телефону, пароль); дані туристів при оформленні бронювання (серія та номер паспорта, дата народження). Вхідними даними для адміністративної частини є: параметри нового туру (назва, опис, маршрут, програма по днях, фотографії, ціна, квоти місць, дати вильотів); облікові дані менеджера (логін, пароль). Джерелом усіх клієнтських даних є форми веб-інтерфейсу; дані турів вносяться менеджером через адміністративну панель.

Вихідна інформація системи. Вихідними даними для клієнта є: відфільтрований перелік турів із коротким описом і мінімальною ціною; детальна картка туру (повний опис, галерея зображень, інтерактивна карта маршруту, таблиця цін за датами, відгуки); підтвердження бронювання та електронний ваучер у форматі PDF; push- та email-сповіщення про зміну статусу замовлення. Вихідними даними для менеджера є: список нових і поточних бронювань зі статусами; зведений звіт про продажі за обраний період; список клієнтів із детальними профілями.

Вимоги до функцій клієнтської частини. Система повинна реалізовувати функцію пошуку турів, що дозволяє задавати до восьми незалежних фільтрів одночасно та формує відсортований результат протягом не більше 3 секунд.

					2026.КВР.122.423.04.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		14

Функція реєстрації та автентифікації має підтримувати стандартний вхід за електронною адресою і паролем. Функція бронювання реалізується у вигляді покрокового майстра (wizard) з п'яти екранів: вибір дати та кількості осіб, введення даних туристів, вибір додаткових послуг, підтвердження та оплата. Функція особистого кабінету забезпечує перегляд активних і завершених бронювань, завантаження документів та редагування профілю. Функція відгуків дозволяє клієнту залишити рейтинг і текстовий коментар після повернення з туру.

Вимоги до функцій адміністративної частини. Менеджер агентства повинен мати можливість створювати, редагувати та архівувати тури через веб-інтерфейс без участі розробника. Система має забезпечувати перегляд усіх бронювань із можливістю зміни статусу (нове, підтвержене, оплачене, скасоване, завершене), надсилання клієнту автоматичного сповіщення при кожній зміні статусу. Адміністративна панель повинна містити інструменти формування звітів у форматах PDF та Excel і забезпечувати управління базою клієнтів.

Вимоги до надійності. З метою забезпечення стабільного функціонування система повинна здійснювати валідацію всіх форм введення даних на стороні клієнта (Angular Reactive Forms) та на стороні сервера (Spring Validation). При введенні некоректних даних користувач отримує чітке діагностичне повідомлення. Система повинна коректно обробляти виняткові ситуації (недоступність бази даних, помилки платіжного шлюзу) із відображенням зрозумілого повідомлення та реєстрацією події в журналі (логах). Доступ до адміністративного розділу захищений механізмом автентифікації JWT з підтримкою ролей. Резервне копіювання бази даних PostgreSQL виконується автоматично раз на добу.

Умови експлуатації. Система розгортається на хмарному сервері з операційною системою Ubuntu Server 26.04 LTS. Клієнтська частина функціонує у браузерях Google Chrome версії 110+, Mozilla Firefox версії 110+, Safari версії 16+, Microsoft Edge версії 110+. Мінімальні вимоги до клієнтського пристрою: будь-який комп'ютер, планшет або смартфон із підтримкою HTML5, CSS3 і JavaScript. Система розрахована на одночасну роботу до 200 активних користувачів.

					2026.KBP.122.423.04.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		15

Менеджери агентства взаємодіють із системою через захищений адміністративний підрозділ (/admin), доступний виключно авторизованим особам.

Часові характеристики. Час відповіді системи на пошуковий запит не повинен перевищувати 3 секунди при стандартному навантаженні. Час завантаження головної сторінки не повинен перевищувати 2 секунди при швидкості з'єднання не менше 10 Мбіт/с. Генерація PDF-документа бронювання виконується протягом не більше 5 секунд.

1.2.4 Вимоги до програмної документації

Склад програмної документації для ІС TourMaster визначається відповідно до потреб розробки та експлуатації системи. Документація повинна містити такі складові.

Пояснювальна записка кваліфікаційної роботи є основним документом, що описує архітектурні рішення, структуру бази даних, принципи роботи клієнтської та серверної частин, а також результати тестування.

Технічна документація API формується у форматі OpenAPI 3.0 (Swagger) та описує всі REST-ендпоінти серверної частини: методи, параметри запитів, формати відповідей і коди помилок. Документація генерується автоматично засобами бібліотеки Springdoc-OpenAPI.

Інструкція з розгортання системи містить покрокові інструкції з встановлення серверного оточення, налаштування змінних середовища, міграції бази даних і запуску всіх компонентів системи.

1.2.5 Техніко-економічні показники

Розробка інформаційної системи «TourMaster» здійснюється в межах виконання кваліфікаційної роботи. Загальні трудовитрати на реалізацію системи оцінюються у 1 людино-місяці, що відповідає приблизно 160 людино-годинам роботи одного розробника. Машинний час, виділений на розробку, тестування та

					2026.КВР.122.423.04.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		16

налагодження, становить орієнтовно 150 годин.

Економічний ефект від впровадження системи визначається скороченням витрат агентства на обробку запитів клієнтів (автоматизація прийому та підтвердження бронювань знижує навантаження на менеджерів приблизно на 40 %) та збільшенням конверсії відвідувачів сайту у покупців завдяки зручному інтерфейсу самообслуговування. Детальний економічний розрахунок наводиться у розділі 4 пояснювальної записки.

1.2.6 Стадії та етапи розробки

Розробка ІС TourMaster здійснюється у шість стадій відповідно до загальноприйнятої методології розробки програмного забезпечення.

На першій стадії — аналізу та проєктування — проводиться збір вимог, аналіз конкурентних рішень, розробка технічного завдання, проєктування структури бази даних, визначення архітектури системи та розробка макетів інтерфейсу (wireframes).

Друга стадія — розробка серверної частини — передбачає налаштування проєкту Spring Boot, реалізацію моделей даних і репозиторіїв JPA, розробку REST API для всіх модулів системи (тури, бронювання, користувачі, автентифікація), налаштування безпеки на основі Spring Security і JWT, інтеграцію з поштовим сервісом для надсилання сповіщень.

Третя стадія — розробка клієнтської частини — охоплює ініціалізацію проєкту Angular 18 зі standalone-компонентами, реалізацію модулів головної сторінки, каталогу, деталей туру, кошика бронювання та особистого кабінету, а також адміністративного інтерфейсу для менеджерів.

Четверта стадія — інтеграція та тестування — включає інтеграцію клієнтської і серверної частин, модульне тестування компонентів Angular за допомогою Jasmine та Karma, інтеграційне тестування API із застосуванням Postman, наскрізне тестування (end-to-end) типових сценаріїв використання.

П'ята стадія — налагодження та усунення виявлених дефектів —

					2026.КВР.122.423.04.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		17

виконується паралельно з тестуванням і передбачає виправлення помилок, оптимізацію запитів до бази даних, налаштування індексів PostgreSQL та перевірку адаптивної верстки на різних пристроях.

Шоста стадія — розгортання та документування — охоплює налаштування серверного оточення, розгортання застосунку, формування фінальної документації та підготовку до захисту кваліфікаційної роботи.

1.2.7 Порядок тестування та прийому

Перевірка працездатності IC TourMaster здійснюється на основі заздалегідь підготовленого набору тестових сценаріїв, що охоплюють усі ключові функції системи. Контрольний приклад включає тестування не менше 40 функціональних випадків використання (use cases), кожен із яких описує вхідні дані, очікуваний результат і фактичний результат.

Прийомне тестування проводиться за такими критеріями: коректне відображення результатів пошуку турів при заданих фільтрах; успішне проходження повного циклу бронювання — від вибору туру до отримання підтвердження на електронну пошту; коректна робота автентифікації та авторизації для обох ролей (клієнт і менеджер); відображення бронювань у особистому кабінеті клієнта та адміністративній панелі менеджера; коректне формування та завантаження PDF-документа.

Умовою успішного прийому системи є виконання усіх функціональних вимог, визначених у підрозділі 1.2.3, відсутність критичних помилок (блокуючих дефектів), а також відповідність вимогам щодо часу відповіді системи. Документування результатів тестування здійснюється у вигляді контрольного списку (checklist), який наводяться в додатках до пояснювальної записки.

					2026.КВР.122.423.04.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		18

2 РОЗРОБКА ТЕХНІЧНОГО ТА РОБОЧОГО ПРОЄКТУ

2.1 Розробка структури інформаційної системи

Проєктування структури інформаційної системи «TourMaster» розпочалося з аналізу функціональних вимог, сформульованих у технічному завданні, та визначення кола суб'єктів, що взаємодіють із системою. Система орієнтована на дві ролі користувачів — клієнта агентства та менеджера агентства — кожна з яких має власний набір функцій і відповідний інтерфейсний контур. На основі цього аналізу визначено склад і призначення всіх сторінок системи, встановлено зв'язки між ними та сформовано загальну структурну схему.

2.1.1 Структура сторінок клієнтського контуру

Клієнтський контур системи охоплює сторінки, доступні кінцевим відвідувачам та зареєстрованим клієнтам агентства. Він побудований за принципом воронки конверсії: кожен наступний крок поглиблює залучення користувача і наближає його до цільової дії — оформлення бронювання.

Головна сторінка (маршрут /) є точкою входу до системи та її основним конверсійним інструментом. Вона містить hero-секцію з формою швидкого пошуку турів (країна, дати, кількість туристів), блок акційних турів у вигляді каруселі, секцію популярних напрямків у форматі інтерактивної сітки та блок відгуків клієнтів. Форма пошуку розташована в зоні першого екрану (above the fold) і забезпечує перехід до каталогу з попередньо встановленими фільтрами.

Каталог турів (маршрут /tours) надає повний перелік туристичних пропозицій із розвиненим механізмом фільтрації. Сторінка підтримує два режими відображення — сіткове та списочне — з можливістю сортування результатів за ціною, рейтингом та датою додавання. Бічна панель фільтрів дозволяє одночасно застосовувати до десяти незалежних критеріїв відбору: країна, діапазон дат, ціновий діапазон, кількість зірок готелю, тип харчування та тривалість поїздки.

					2026.КВР.122.423.04.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		19

Усі активні фільтри відображаються в URL у вигляді query-параметрів, що забезпечує можливість збереження та передачі посилання на відфільтровані результати.

Сторінка деталей туру (маршрут /tours/:id) надає вичерпну інформацію про конкретну туристичну пропозицію. Вміст організований у дволонковий макет: основна частина містить фотогалерею та чотири тематичні вкладки (Опис, Програма по днях, Включено/не включено, Відгуки), а бічна панель із переліком доступних дат вильоту та калькулятором вартості є «липкою» (sticky) і залишається у видимій зоні при прокрутці сторінки. СТА-кнопка «Забронювати» є постійно доступною для клієнта незалежно від позиції прокрутки.

Сторінка бронювання (маршрут /booking) реалізована у вигляді покрокового майстра (wizard) з п'яти послідовних кроків: підтвердження туру і дати, введення паспортних даних туристів, вибір додаткових послуг (страховка, трансфер, екскурсії), огляд повного замовлення та оплата. Доступ до цієї сторінки обмежено захистом authGuard — неавторизований користувач перенаправляється на сторінку входу зі збереженням параметра returnUrl для автоматичного повернення після авторизації. Після успішного оформлення бронювання відображається сторінка підтвердження (маршрут /booking/success) із номером замовлення.

Особистий кабінет клієнта (маршрут /cabinet) реалізований як вкладений маршрут і доступний лише авторизованим користувачам. Він об'єднує три функціональні секції: список бронювань із фільтрацією за статусом та можливістю завантаження документів, список збережених турів (wishlist) і сторінку редагування профілю зі зміною пароллю. Кожне бронювання у списку відображається із кольоровим бейджем статусу, що дозволяє клієнту миттєво оцінити поточний стан замовлення.

Сторінки аутентифікації (маршрути /auth/login та /auth/register) побудовані в єдиному split-layout: ліва половина містить тематичне зображення та слоган агентства, права — форму відповідної дії. Форма реєстрації включає перевірку надійності пароллю в режимі реального часу. При невдалій спробі входу або реєстрації з вже зареєстрованим email система відображає діагностичне

					2026.КВР.122.423.04.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		20

повідомлення.

Перелік сторінок клієнтського контуру з їх призначенням та рівнем доступу наведено у таблиці 2.1.

Таблиця 2.1 – Сторінки клієнтської частини ІС TourMaster

Маршрут	Компонент	Призначення
/	HomeComponent	Головна: hero-секція з пошуком, карусель турів, популярні напрямки, відгуки
/tours	TourCatalogComponent	Каталог з панеллю фільтрів, перемикання grid/list, пагінація
/tours/:id	TourDetailComponent	Деталі туру: галерея, вкладки, sticky-сайдбар із датами та ціною
/booking	BookingWizardComponent	5-кроковий wizard: дата → туристи → послуги → огляд → оплата
/booking/success	BookingSuccessComponent	Підтвердження бронювання з номером замовлення
/cabinet/bookings	MyBookingsComponent	Список бронювань клієнта з фільтрами за статусом
/cabinet/wishlist	WishlistComponent	Збережені (обрані) тури клієнта
/cabinet/profile	ProfileComponent	Редагування профілю та зміна паролю
/auth/login	LoginComponent	Форма входу в систему
/auth/register	RegisterComponent	Форма реєстрації нового клієнта

2.1.2 Структура сторінок адміністративного контуру

Адміністративний контур системи призначений для менеджерів агентства та повністю відокремлений від клієнтської частини на рівні маршрутизації. Усі сторінки адміністративного розділу захищені managerGuard, який перевіряє роль MANAGER у поточному сеансі. Спроба доступу з правами клієнта призводить до перенаправлення на головну сторінку.

Дашборд (маршрут /admin) є початковою сторінкою адміністративного контуру і надає зведений оперативний огляд стану агентства: ключові показники ефективності (нові бронювання, виручка за місяць, нові клієнти), таблицю останніх бронювань зі статусами та перелік найпопулярніших турів поточного місяця. Дашборд є точкою навігації до всіх інших розділів адміністративної панелі.

Сторінка управління турами (маршрут /admin/tours) надає менеджеру інструменти для повного керування туристичним каталогом: перегляд усіх турів у табличному форматі з пошуком і фільтрацією за статусом, створення нового туру через діалогове вікно з розгорнутою формою, редагування параметрів і дат вильоту існуючого туру, а також архівування турів із активними обмеженнями (тури з підтвердженими бронюваннями не можна видалити — лише перевести в статус ARCHIVED).

Сторінка управління бронюваннями (маршрут /admin/bookings) відображає всі бронювання системи з фільтрацією за статусом та місяцем. Менеджер може змінити статус бронювання безпосередньо в рядку таблиці, після чого система автоматично надсилає клієнту email-сповіщення про зміну. Деталі кожного бронювання (дані туристів, перелік додаткових послуг, нотатка менеджера) доступні у модальному вікні.

Сторінка бази клієнтів (маршрут /admin/clients) надає доступ до реєстру зареєстрованих клієнтів із можливістю пошуку за іменем або електронною адресою. Профіль клієнта містить персональні дані, рівень лояльності, повну хронологію бронювань та нотатки менеджерів. Додавання нотатки до профілю

					2026.KBP.122.423.04.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		22

дозволяє фіксувати важливу контекстну інформацію для персоналізованого обслуговування.

Перелік сторінок адміністративного контуру наведено у таблиці 2.2.

Таблиця 2.2 – Сторінки адміністративного контуру інформаційної системи TourMaster

Маршрут	Компонент	Призначення та ключові функції
/admin	AdminDashboardComponent	Дашборд: останні бронювання, топ турів за місяць
/admin/tours	AdminToursComponent	Каталог турів: CRUD, дати вильоту, архівування, пошук і фільтрація
/admin/bookings	AdminBookingsComponent	Бронювання: зміна статусів, email-сповіщення, деталі замовлень
/admin/clients	AdminClientsComponent	База клієнтів: профіль, історія бронювань, нотатки менеджера

Для формалізації функціональних вимог побудовано діаграму варіантів використання (Use Case Diagram), що охоплює всі сценарії взаємодії усіх ролей із системою. Діаграму наведено у графічній частині 2026.КВР.122.423.04.0000 ДВ. Загальна структурна схема клієнтської частини, що відображає ієрархію сторінок та навігаційні зв'язки між ними, наведена у 2026.КВР.122.423.04.0000 СС.

2.2 Створення та верстка сторінок інформаційної системи

Розробка клієнтського інтерфейсу здійснювалася відповідно до дизайн-паketу (handoff), підготовленого в Claude Design. Дизайн-паket включає систему токенів (кольори, типографіка, відступи), специфікації всіх UI-компонентів та макети кожної сторінки у форматах Desktop (1280px) та Mobile (375px).

					2026.КВР.122.423.04.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		23

Збереження файлів дизайну у підкаталозі frontend/design/ репозиторію гарантує доступність специфікацій для всіх учасників розробки безпосередньо у середовищі IDE.

Верстка сторінок системи базується на поєднанні двох інструментів: бібліотеки компонентів PrimeNG 21.1.9 та утилітарного CSS-фреймворку Tailwind CSS 4.3. Такий підхід дозволяє використовувати готові інтерактивні компоненти PrimeNG (Carousel, DataView, Galleria, Dialog, Toast, Rating, Calendar, Paginator) для складних UI-елементів, тоді як Tailwind CSS забезпечує гнучке управління макетом, відступами та типографікою поза межами компонентів.

Особливістю налаштування проєкту є необхідність використання прапора --legacy-peer-deps під час встановлення залежностей через npm, оскільки PrimeNG 21 і Angular 22 мають певні конфлікти peer dependencies. Tailwind CSS 4.x потребує окремого кроку генерації CSS: команда npm run tw:build виконує tailwindcss -i src/tailwind.css -o src/tailwind-out.css, після чого згенерований файл підключається до проєкту. Під час розробки використовується команда npm run tw:watch у фоновому режимі для автоматичної регенерації при зміні класів.

Глобальні стилі та кастомну PrimeNG-тему визначено у файлі styles.scss. Дизайн-токени з handoff-файлу перенесено у вигляді CSS custom properties та конфігурації Tailwind, що забезпечує консистентність між кастомними стилями та стилями PrimeNG-компонентів. Основні значення токенів: первинний колір #1A6FBF (синій, асоціюється з довірою), колір СТА #F97316 (помаранчевий, для кнопок бронювання та акцій), нейтральний фон #F8F9FA, основний текст #1C1C1E.

Концепцію Mobile First реалізовано через адаптивний підхід: базові стилі написані для мобільного відображення (375px), а розширення для ширших екранів задаються через медіа-запити Tailwind (md: для 768px, lg: для 1280px, xl: для 1440px). Типографіка базується на гарнітурі Inter з розміром основного тексту 16px.

Головна сторінка

Головна сторінка (HomeComponent, маршрут /) є основним інструментом

					2026.КВР.122.423.04.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		24

конверсії відвідувачів у клієнтів агентства і слугує точкою входу до всіх функцій системи. Сторінка складається з кількох послідовних секцій.

Неро-секція займає повну висоту екрану і містить фонове зображення туристичної тематики з накладеним напівпрозорим градієнтом для забезпечення читабельності тексту. Поверх розміщено заголовок агентства та форму швидкого пошуку з чотирма полями: країна призначення (PrimeNG AutoComplete), дати поїздки (PrimeNG Calendar у режимі range), кількість туристів (числовий input) та кнопка «Знайти тури» (CTA-кнопка в кольорі #F97316). Ця форма є головним конверсійним елементом сторінки, тому вона завжди розміщена у видимій зоні без прокрутки (above the fold).

Нижче розташована секція з PrimeNG Carousel, що відображає акційні тури. Карусель налаштована на показ трьох карток одночасно на десктопі та однієї на мобільному. Кожна картка містить фотографію туру з lazy load через ng-lazyload-image, назву, країну, рейтинг (PrimeNG Rating) та ціну з бейджем акції.

Секція «Популярні напрямки» реалізована у вигляді адаптивного grid 3×2 на десктопі та 2×3 на мобільному. Кожна клітинка містить фонове зображення країни з назвою та кількістю доступних турів. Клік на клітинку переходить до каталогу з попередньо встановленим фільтром по країні. Завершує структуру сторінки секція відгуків клієнтів із картками, що містять PrimeNG Rating-компоненти, текст коментаря та ім'я автора.

Каталог турів та сторінка деталей туру

Сторінка каталогу (TourCatalogComponent, маршрут /tours) реалізована з використанням компонента PrimeNG DataView, що підтримує перемикання між двома режимами відображення: grid (сіткове, 3 колонки на десктопі) та list (списочне). Вибраний режим зберігається у query-параметрі URL, що дозволяє відтворити стан при поверненні на сторінку.

Ліва панель фільтрів (Sidebar) доступна постійно на десктопі та відкривається як слайдер (PrimeNG Sidebar) на мобільних пристроях. Панель містить: мультивибір країн (PrimeNG MultiSelect), вибір діапазону дат (PrimeNG Calendar), повзунок ціни (PrimeNG Slider у режимі range), чекбокси зірковості

					2026.KBP.122.423.04.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		25

готелю та типу харчування (ВВ/НВ/ВВ/АІ), а також поле діапазону тривалості. Усі фільтри відразу відображаються у URL як query-параметри, що забезпечує можливість збереження та передачі посилання на відфільтровані результати.

Картка туру є спільним компонентом (TourCardComponent у папці shared/components), що використовується як у каталозі, так і на головній сторінці. Вона містить: фотографію туру з lazy load та ефектом плавної появи (fade-in), бейдж у верхньому куті (HIT, SALE, NEW, LAST_SEATS), назву туру, країну та тривалість, рейтинг (зірки PrimeNG Rating), ціну з відображенням старої ціни за наявності та СТА-кнопку «Детальніше». При наведенні картка піднімається з ефектом box-shadow через CSS transition.

Сторінка деталей туру (TourDetailComponent, маршрут /tours/:id) організована у дволонковий layout: основний контент займає 8 колонок, липкий (sticky) сайдбар бронювання — 4 колонки на десктопі. На мобільному сайдбар переміщується на початок сторінки над основним контентом. Фотогалерея реалізована на базі PrimeNG Galleria з підтримкою повноекранного режиму перегляду. Основний контент структурований у чотири вкладки (PrimeNG TabView): «Опис», «Програма по днях», «Включено/не включено», «Відгуки». Вкладка «Відгуки» відображає список відгуків із компонентом PrimeNG Rating для кожного.

Сайдбар бронювання на сторінці туру є sticky-елементом і завжди залишається у видимій зоні при прокрутці. Він містить таблицю доступних дат вильоту з кількістю вільних місць та ціною, лічильник кількості туристів і СТА-кнопку «Забронювати» в помаранчевому кольорі. При натисканні дані переміщуються до CartService і відбувається перехід на сторінку wizard бронювання.

Wizard бронювання

Процес бронювання реалізовано у вигляді п'ятикрокового wizard (BookingWizardComponent, маршрут /booking). Цей маршрут захищений authGuard: неаутентифікований користувач перенаправляється на сторінку входу зі збереженням параметра returnUrl для повернення після авторизації. Прогрес-бар

					2026.КВР.122.423.04.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		26

вгорі сторінки відображає поточний крок і загальну кількість кроків. У правій частині (або внизу на мобільному) відображається резюме замовлення з назвою туру, датою, кількістю туристів та поточною сумою.

Перший крок (Підтвердження) відображає обраний тур, дату вильоту та повернення, кількість туристів. Другий крок (Дані туристів) містить динамічну форму: кількість груп полів відповідає кількості туристів. Кожна група включає поля імені, прізвища, дати народження (PrimeNG Calendar), серії та номери паспорта, терміну його дії. Форма реалізована через Angular Reactive Forms із відповідними валідаторами. Третій крок (Додаткові послуги) відображає картки страховок, трансферів та екскурсій із чекбоксами для вибору. Вартість кожної послуги автоматично додається до загальної суми у резюме. Четвертий крок (Огляд) показує зведену таблицю всього замовлення перед підтвердженням. П'ятий крок (Оплата) містить форму введення платіжних реквізитів. Після успішного відправлення запиту POST /api/bookings відбувається перехід на сторінку BookingSuccessComponent із номером бронювання та СТА-кнопками для переходу до кабінету або на головну.

Особистий кабінет клієнта

Особистий кабінет (маршрут /cabinet) реалізовано з використанням вкладеного маршрутизатора: CabinetLayoutComponent є обгорткою з бічним навігаційним меню і <router-outlet>, в який підставляються три дочірні компоненти: MyBookingsComponent (/cabinet/bookings), WishlistComponent (/cabinet/wishlist) та ProfileComponent (/cabinet/profile).

Сторінка «Мої бронювання» відображає список бронювань клієнта, отриманих через GET /api/bookings/my. Картка кожного бронювання містить назву туру, дату вильоту, кількість туристів, загальну вартість і статус у вигляді кольорового бейджа. Кольорове кодування статусів реалізоване через компонент BookingBadge (shared/components): NEW — синій (#3B82F6), CONFIRMED — жовтий (#F59E0B), PAID — зелений (#22C55E), CANCELLED — червоний (#EF4444), COMPLETED — сірий (#6B7280). Фільтрація за статусом реалізована через PrimeNG ButtonGroup. Сторінка «Збережені тури» (Wishlist) відображає

					2026.КВР.122.423.04.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		27

список турів, доданих до вибраного. Сторінка «Профіль» містить форму редагування персональних даних та зміни паролю.

Сторінки аутентифікації

Сторінки входу (LoginComponent, маршрут /auth/login) та реєстрації (RegisterComponent, маршрут /auth/register) реалізовані у split-layout: ліва половина екрану містить тематичне фонове зображення та слоган агентства, права — форму авторизації. На мобільних пристроях зображення приховується і відображається лише форма.

Форма входу містить два поля (email та пароль) і кнопку входу. Після успішного запиту POST /api/auth/login AuthService зберігає токени та дані користувача і виконує навігацію за збереженим returnUrl або на головну сторінку. Форма реєстрації включає поля імені, прізвища, email, пароля та підтвердження пароля з індикатором надійності паролю. Валідація email-адреси виконується на рівні Angular Reactive Forms, додаткова перевірка на сервері повертає помилку 409 Conflict при спробі реєстрації з уже існуючим email.

Адміністративна панель

Адміністративна панель (маршрут /admin) доступна лише користувачам із роллю MANAGER. Захист реалізовано через managerGuard, що перевіряє значення computed-сигналу AuthService.isManager. Структурно панель аналогічна кабінету клієнта: AdminLayoutComponent є обгорткою зі збільшеним бічним навігаційним меню, яке містить посилання на чотири секції адміністрування.

Дашборд (AdminDashboardComponent) є початковою сторінкою адміністративного розділу. Він містить чотири stat-картки з ключовими показниками (кількість нових бронювань сьогодні, виручка за поточний місяць, нові клієнти за тиждень, загальна кількість активних турів), таблицю п'яти найостанніших бронювань зі статусами та кнопками швидкої зміни статусу, а також список топ-5 турів за кількістю продажів поточного місяця.

Сторінка управління турами (AdminToursComponent) відображає всі тури у вигляді таблиці з полями: мініатюра, назва, країна, базова ціна, статус (ACTIVE/ARCHIVED) та стовпець дій. Над таблицею розташовані поле пошуку

					2026.KBP.122.423.04.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		28

та фільтр за статусом. Кнопка «Додати тур» відкриває PrimeNG Dialog з розгорнутою формою для введення всіх даних туру, включаючи поля завантаження URL фотографій. Редагування існуючого туру відкриває аналогічний Dialog з попередньо заповненими полями. Архівування туру виконується через DELETE /api/admin/tours/{id} з попереднім підтвердженням через PrimeNG ConfirmDialog.

Сторінка управління бронюваннями (AdminBookingsComponent) надає таблицю всіх бронювань із можливістю фільтрації за статусом та місяцем. Статус бронювання можна змінити безпосередньо з таблиці через PrimeNG Select у відповідному рядку, після чого система автоматично надсилає клієнту email-сповіщення через асинхронний NotificationService. Деталі бронювання (дані туристів, додаткові послуги) відкриваються у PrimeNG Dialog.

Сторінка бази клієнтів (AdminClientsComponent) відображає список зареєстрованих клієнтів із полем пошуку за іменем або email. При переході до профілю клієнта (через GET /api/admin/clients/{id}) менеджер бачить персональні дані, рівень лояльності (loyalty_points), повну історію бронювань та наявні нотатки. Додавання нотатки виконується через POST /api/admin/clients/{id}/notes.

Спільні компоненти та навігація

Спільну шапку сайту (HeaderComponent) реалізовано з ефектом зміни прозорості при прокрутці: у верхній частині сторінки шапка є напівпрозорою (backdrop-blur), при прокрутці набуває суцільного білого фону. Шапка містить логотип агентства, основне навігаційне меню, кнопку пошуку та блок аутентифікації (кнопки «Вхід» / «Реєстрація» для неавторизованих або аватар і меню профілю для авторизованих). Футер (FooterComponent) містить контактні дані агентства, посилання на розділи сайту та блок соціальних мереж.

Для мобільних пристроїв реалізовано нижню навігаційну панель (MobileTabBarComponent), що забезпечує швидкий доступ до основних розділів: Головна, Каталог, Бронювання та Кабінет. Панель фіксована внизу екрану і відображається виключно на пристроях із шириною менше 768px через медіа-запити Tailwind.

					<i>2026.KBP.122.423.04.00.00 ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		29

Для відображення стану завантаження даних реалізовано компонент скелетного навантажувача (skeleton-loader), що замінює картки турів у каталозі та деталях під час очікування відповіді від API. Це покращує сприйняту швидкість роботи застосунку. PrimeNG Toast-компонент підключено глобально через app.config.ts і використовується errorInterceptor для відображення повідомлень про помилки та сервісними методами для підтвердження успішних операцій (бронювання, зміна даних профілю тощо). Повідомлення автоматично зникають через 3 секунди.

2.3 Розробка структури бази даних інформаційної системи

База даних інформаційної системи «TourMaster» реалізована у СУБД PostgreSQL 16 та складається з одинадцяти взаємопов'язаних таблиць. Проектування схеми здійснювалося за принципами третьої нормальної форми (3НФ): усунуто надмірність даних, кожне поле залежить виключно від первинного ключа, транзитивних залежностей між атрибутами немає. Усі первинні ключі мають тип BIGSERIAL, що забезпечує автоматичну генерацію унікальних ідентифікаторів. Цілісність посилань між таблицями підтримується зовнішніми ключами із відповідними обмеженнями.

Управління версіями схеми здійснюється через Flyway — інструмент міграцій, що автоматично застосовує SQL-скрипти при кожному старті застосунку. Міграції виконуються у строгому порядку: V1__init_schema.sql створює всі основні таблиці та індекси, V2__seed_data.sql наповнює базу демонстраційними даними (тури, послуги, тестові облікові записи), V3__manager_notes.sql додає таблицю нотаток менеджера. Такий підхід гарантує відтворюваність структури бази даних у будь-якому середовищі — локальному, тестовому та продакшн.

ER-діаграма бази даних, що відображає всі таблиці, їх атрибути та зв'язки між ними, наведена у графічній частині (див. 2026.КВР.122.423.04.0000 БД). Нижче подано детальний опис кожної таблиці.

					2026.КВР.122.423.04.00.00 ПЗ	Арк.
						30
Зм.	Арк.	№ докум.	Підпис	Дата		

Таблиця users є центральною сутністю системи і зберігає облікові записи як клієнтів, так і менеджерів агентства. Розрізнення ролей здійснюється через поле role, яке приймає одне з двох значень: CLIENT або MANAGER. Такий підхід дозволяє зберігати обидві ролі в одній таблиці та керувати правами доступу виключно на рівні застосунку, не ускладнюючи схему БД.

Поле email є унікальним і слугує логіном для входу в систему. Пароль зберігається виключно у вигляді BCrypt-хешу — сирий пароль ніде не записується. Поле loyalty_points зберігає бали лояльності клієнта для майбутньої реалізації програми знижок. Поля created_at та updated_at заповнюються автоматично і використовуються для аудиту облікових записів. Детальний опис структури таблиці users наведено у таблиці 2.3.

Таблиця 2.3 – Структура таблиці users

Поле	Тип	Опис
id	BIGSERIAL PK	Унікальний ідентифікатор запису
email	VARCHAR(255) UNIQUE NOT NULL	Електронна адреса — логін у системі
password	VARCHAR(255) NOT NULL	BCrypt-хеш паролю (сирий пароль не зберігається)
first_name	VARCHAR(100) NOT NULL	Ім'я користувача
last_name	VARCHAR(100) NOT NULL	Прізвище користувача
phone	VARCHAR(20)	Номер телефону (необов'язкове)
role	VARCHAR(20) NOT NULL	Роль у системі
loyalty_points	INT NOT NULL DEFAULT 0	Накопичені бали лояльності
created_at	TIMESTAMP NOT NULL	Дата та час реєстрації (аудит)
updated_at	TIMESTAMP NOT NULL	Дата та час останнього оновлення

Таблиця tours зберігає основні характеристики туристичних пропозицій агентства. Кожен запис відповідає одному туристичному продукту незалежно від кількості доступних дат вильоту — дати зберігаються в окремій таблиці tour_dates із прив'язкою до туру.

Поле price_from містить базову ціну туру, тоді як old_price зберігає попередню ціну для відображення знижки у клієнтському інтерфейсі. Поле badge приймає одне з чотирьох значень (HIT, SALE, NEW, LAST_SEATS) і відображається у вигляді кольорової мітки на картці туру. Поле gallery_urls використовує PostgreSQL-тип TEXT[] (масив рядків) для зберігання переліку URL зображень галереї без потреби у окремій таблиці фотографій. Поле status визначає видимість туру в каталозі: значення ARCHIVED унеможливує відображення туру клієнтам, але зберігає всі пов'язані бронювання в цілісності. Детальний опис структури таблиці tours наведено у таблиці 2.4.

Таблиця 2.4 – Структура таблиці tours

Поле	Тип	Опис
1	2	3
id	BIGSERIAL PK	Унікальний ідентифікатор туру
title	VARCHAR(255) NOT NULL	Назва туру
description	TEXT	Повний текстовий опис туру
country	VARCHAR(100) NOT NULL	Країна призначення
city	VARCHAR(100)	Місто / курорт призначення
hotel_name	VARCHAR(255)	Назва готелю
hotel_stars	SMALLINT	Категорія готелю (1–5 зірок)
meal_type	VARCHAR(50)	Тип харчування: ВВ, НВ, ФВ або АІ

Продовження таблиці 2.4

1	2	3
duration_nights	SMALLINT NOT NULL	Тривалість туру в ночах
price_from	NUMERIC(12,2) NOT NULL	Базова ціна (мінімальна вартість за турдатою)
old_price	NUMERIC(12,2)	Попередня ціна для відображення знижки
badge	VARCHAR(50)	Мітка картки: HIT, SALE, NEW, LAST_SEATS
rating	NUMERIC(2,1)	Середній рейтинг за відгуками (0.0–5.0)
reviews_count	INT	Загальна кількість відгуків
image_url	VARCHAR(512)	URL головного фото туру
gallery_urls	TEXT[]	Масив URL зображень фотогалереї
status	VARCHAR(20) NOT NULL	Стан туру: ACTIVE або ARCHIVED
created_at	TIMESTAMP NOT NULL	Дата створення запису (аудит)
updated_at	TIMESTAMP NOT NULL	Дата останнього оновлення (аудит)

Таблиця `tour_dates` реалізує зв'язок один тур — багато дат вильоту. Кожен запис відповідає конкретному рейсу: має власні дати вильоту та повернення, квоту місць і можливість перевизначити базову ціну туру для цієї конкретної дати.

Поле `available_seats` є ключовим для бізнес-логіки бронювання: при підтвердженні бронювання сервіс атомарно зменшує це значення, при скасуванні — відновлює. Поле `price_override` дозволяє встановити індивідуальну ціну для

конкретної дати вильоту (наприклад, для святкових турів або акційних пропозицій). Якщо price_override має значення NULL, застосовується базова ціна з поля tours.price_from. Детальний опис структури таблиці tour_dates наведено у таблиці 2.5.

Таблиця 2.5 – Структура таблиці tour_dates

Поле	Тип	Опис
id	BIGSERIAL PK	Унікальний ідентифікатор дати вильоту
tour_id	BIGINT FK → tours(id)	Посилання на тур
departure_date	DATE NOT NULL	Дата вильоту
return_date	DATE NOT NULL	Дата повернення
departure_city	VARCHAR(100)	Місто вильоту (необов'язкове)
total_seats	INT NOT NULL	Загальна квота місць у групі
available_seats	INT NOT NULL	Поточна кількість вільних місць
price_override	NUMERIC(12,2)	Ціна для цієї дати (NULL — з tours.price_from)

Таблиця bookings є центром транзакційної частини системи і фіксує кожен факт бронювання туру клієнтом. Вона пов'язана з таблицею users через поле user_id (клієнт, що здійснив бронювання) та з таблицею tour_dates через поле tour_date_id (конкретна дата вильоту).

Поле status відображає поточний стан бронювання і може набувати п'яти значень: NEW (щойно створене), CONFIRMED (підтверджене менеджером), PAID (оплачене), COMPLETED (тур завершено), CANCELLED (скасоване). Переходи між статусами ініціюються менеджером через адміністративну панель. При кожній зміні статусу система автоматично надсилає клієнту email-сповіщення.

Поле `total_price` фіксує вартість бронювання на момент його створення та не змінюється при подальших коригуваннях ціни туру.

Поле `manager_note` дозволяє менеджеру залишити службову примітку, невидиму для клієнта. Детальний опис структури таблиці `bookings` наведено у таблиці 2.6.

Таблиця 2.6 – Структура таблиці `bookings`

Поле	Тип	Опис
<code>id</code>	BIGSERIAL PK	Унікальний ідентифікатор бронювання
<code>user_id</code>	BIGINT FK → <code>users(id)</code>	Клієнт, що здійснив бронювання
<code>tour_date_id</code>	BIGINT FK → <code>tour_dates(id)</code>	Обрана дата вильоту
<code>status</code>	VARCHAR(20) NOT NULL	Статус: NEW, CONFIRMED, PAID, COMPLETED, CANCELLED
<code>total_price</code>	NUMERIC(12,2) NOT NULL	Повна вартість на момент бронювання
<code>discount</code>	NUMERIC(12,2) NOT NULL DEFAULT 0	Застосована знижка
<code>manager_note</code>	TEXT	Службова примітка менеджера (клієнту не видна)
<code>created_at</code>	TIMESTAMP NOT NULL	Дата та час створення бронювання (аудит)
<code>updated_at</code>	TIMESTAMP NOT NULL	Дата та час останнього оновлення статусу (аудит)

Таблиця `booking_items` зберігає паспортні дані кожного туриста у складі бронювання. Зв'язок з таблицею `bookings` є відношенням один до багатьох: одне бронювання може містити записи для кількох туристів. Кількість записів у цій таблиці для конкретного бронювання відповідає кількості туристів, вказаній при оформленні. Детальний опис структури таблиці `booking_items` наведено у таблиці 2.7.

Таблиця 2.7 – Структура таблиці `booking_items`

Поле	Тип	Опис
<code>id</code>	BIGSERIAL PK	Унікальний ідентифікатор запису туриста
<code>booking_id</code>	BIGINT FK → <code>bookings(id)</code>	Посилання на бронювання
<code>first_name</code>	VARCHAR(100) NOT NULL	Ім'я туриста
<code>last_name</code>	VARCHAR(100) NOT NULL	Прізвище туриста
<code>birth_date</code>	DATE NOT NULL	Дата народження туриста
<code>passport_number</code>	VARCHAR(50) NOT NULL	Серія та номер паспорта
<code>passport_expiry</code>	DATE NOT NULL	Термін дії паспорта

Таблиця `extra_services` містить довідник додаткових туристичних послуг, що пропонуються агентством. Послуги поділяються на три типи: `INSURANCE` (страховий поліс), `TRANSFER` (трансфер аеропорт–готель) та `EXCURSION` (екскурсійна програма). Довідник є спільним для всіх бронювань.

Зв'язок багато до багатьох між таблицями `bookings` та `extra_services` реалізовано через таблицю-зв'язку `booking_extra_services`. Вона містить лише два

зовнішні ключі — booking_id та service_id — і не має власного ідентифікатора. Складений первинний ключ (booking_id, service_id) гарантує, що одна послуга не може бути додана до одного бронювання двічі. Детальний опис структури таблиці extra_services наведено у таблиці 2.8.

Таблиця 2.8 – Структура таблиці extra_services

Поле	Тип	Опис
id	BIGSERIAL PK	Унікальний ідентифікатор послуги
name	VARCHAR(255) NOT NULL	Назва послуги
description	TEXT	Детальний опис послуги
price	NUMERIC(10,2) NOT NULL	Вартість послуги
type	VARCHAR(50) NOT NULL	Тип: INSURANCE, TRANSFER або EXCURSION

Таблиця reviews зберігає відгуки клієнтів на завершені тури. Кожен відгук пов'язаний одночасно з трьома таблицями: users (автор відгуку), tours (тур, про який написано відгук) і bookings (бронювання, що є підставою для відгуку). Унікальне обмеження на пару (user_id, booking_id) виключає можливість залишення більше одного відгуку на одне бронювання.

Правило дозволу на публікацію відгуку реалізоване на рівні сервісного шару: клієнт може залишити відгук лише для бронювань зі статусом COMPLETED.

При публікації відгуку сервіс автоматично перераховує поля rating та reviews_count у відповідному записі таблиці tours. Унікальне обмеження: UNIQUE (user_id, booking_id) — один відгук на одне бронювання. Детальний опис структури таблиці reviews наведено у таблиці 2.9.

Таблиця 2.9 – Структура таблиці reviews

Поле	Тип	Опис
id	BIGSERIAL PK	Унікальний ідентифікатор відгуку
user_id	BIGINT FK → users(id)	Автор відгуку
tour_id	BIGINT FK → tours(id)	Тур, якого стосується відгук
booking_id	BIGINT FK → bookings(id)	Бронювання-підстава для відгуку
rating	SMALLINT NOT NULL	Оцінка від 1 до 5
comment	TEXT	Текст відгуку (необов'язковий)
created_at	TIMESTAMP NOT NULL	Дата публікації відгуку

Таблиця refresh_tokens забезпечує stateless JWT-аутентифікацію з підтримкою функції виходу з системи (logout). При кожному успішному вході в систему генерується пара токенів: короткоживучий access token (TTL 15 хвилин) і довгоживучий refresh token (TTL 7 діб). Refresh token зберігається в цій таблиці, а його хеш-значення використовується для верифікації запитів на оновлення сесії.

При виклику ендпоінту /api/auth/logout відповідний запис видаляється з таблиці, що унеможливорює подальше використання refresh token навіть за умови його перехоплення. Термін дії токена контролюється полем expires_at: прострочені токени відхиляються на рівні JwtService. Детальний опис структури таблиці refresh_tokens наведено у таблиці 2.10.

Таблиця 2.10 – Структура таблиці refresh_tokens

Поле	Тип	Опис
id	BIGSERIAL PK	Унікальний ідентифікатор запису
token	VARCHAR(512) UNIQUE NOT NULL	JWT refresh-токен (унікальний рядок)
user_id	BIGINT FK → users(id)	Власник токена
expires_at	TIMESTAMP NOT NULL	Дата та час закінчення дії токена
created_at	TIMESTAMP NOT NULL	Дата та час видачі токена

Таблиця manager_notes зберігає службові нотатки, які менеджери агентства залишають до профілів клієнтів. Таблиця додана в рамках міграції V3 після первинного розгортання схеми. Вона містить два зовнішні ключі до таблиці users: user_id вказує на клієнта, якого стосується нотатка, а manager_id — на менеджера, що її залишив. Нотатки є виключно службовими: клієнт не має доступу до їх перегляду. Опис структури таблиці manager_notes наведено у таблиці 2.11.

Таблиця 2.11 – Структура таблиці manager_notes

Поле	Тип	Опис
id	BIGSERIAL PK	Унікальний ідентифікатор нотатки
user_id	BIGINT FK → users(id)	Клієнт, якого стосується нотатка
manager_id	BIGINT FK → users(id)	Менеджер, що залишив нотатку
note	TEXT NOT NULL	Текст службової нотатки
created_at	TIMESTAMP NOT NULL	Дата та час створення нотатки

Таблиця wishlist реалізує функцію збереження турів клієнтом. Вона є таблицею-зв'язкою Many-to-Many між users та tours. Первинний ключ є складеним

і утворений парою полів (user_id, tour_id), що виключає дублювання одного туру у списку збережених одного клієнта. Таблиця не містить жодних додаткових атрибутів. Детальний опис структури таблиці wishlist наведено у таблиці 2.12.

Таблиця 2.12 – Структура таблиці wishlist

Поле	Тип	Опис
user_id	BIGINT FK → users(id) PK	Клієнт (частина складеного PK)
tour_id	BIGINT FK → tours(id) PK	Тур (частина складеного PK)

Таблиця notifications є журналом усіх email-сповіщень, надісланих системою. Вона зберігає тип події, тему та тіло листа, час надсилання та статус доставки. Статус приймає три значення: PENDING (поставлено в чергу), SENT (успішно надіслано), FAILED (помилка надсилання). Журнал дозволяє менеджеру або адміністратору перевірити факт відправлення листа у разі претензій від клієнта. Детальний опис структури таблиці notifications наведено у таблиці 2.13.

Таблиця 2.13 – Структура таблиці notifications

Поле	Тип	Опис
id	BIGSERIAL PK	Унікальний ідентифікатор сповіщення
user_id	BIGINT FK → users(id)	Одержувач сповіщення
type	VARCHAR(50) NOT NULL	Тип події (зміна статусу, підтвердження тощо)
subject	VARCHAR(255) NOT NULL	Тема листа
body	TEXT NOT NULL	HTML-тіло листа
sent_at	TIMESTAMP	Фактичний час надсилання
status	VARCHAR(20) NOT NULL	Статус: PENDING, SENT або FAILED

Схема бази даних охоплює чотири типи зв'язків між таблицями. Зв'язок один до багатьох реалізований між tours та tour_dates (один тур — кілька дат вильоту), між tour_dates та bookings (одна дата — кілька бронювань), між bookings та booking_items (одне бронювання — кілька туристів), між users та bookings, reviews, refresh_tokens, manager_notes, notifications. Зв'язок багато до багатьох реалізований між bookings та extra_services через таблицю-зв'язку booking_extra_services та між users та tours через wishlist.

Цілісність посилань (referential integrity) забезпечується зовнішніми ключами з обмеженням NOT NULL там, де зв'язок є обов'язковим. Для полів необов'язкових зв'язків (наприклад, phone у users, departure_city у tour_dates) тип nullable дозволяє відсутність значення без порушення цілісності схеми.

На рівні застосунку реалізовано додаткові бізнес-обмеження, які не можуть бути виражені стандартними засобами SQL: заборона архівування туру з активними бронюваннями (статуси NEW, CONFIRMED, PAID), відновлення доступних місць у tour_dates при скасуванні бронювання, автоматичний перерахунок рейтингу туру після публікації нового відгуку.

Повна ER-діаграма бази даних із відображенням усіх атрибутів, первинних та зовнішніх ключів і типів зв'язків між таблицями наведена на плакаті графічної частини 2026.КВР.122 423.04.00.00 БД.

2.4 Програмування інформаційної системи

Програмування інформаційної системи «TourMaster» здійснювалося у двох технологічних контурах: серверна частина реалізована на Java 21 з використанням фреймворку Spring Boot 3.5, клієнтська частина — на TypeScript з фреймворком Angular 22. Такий поділ відповідає архітектурному рішенню, ухваленому на етапі проєктування, та забезпечує незалежний цикл розробки кожного компонента системи. Розробку обох частин виконано у середовищі IntelliJ IDEA Ultimate, яке надає уніфіковану підтримку Java, TypeScript та Angular-специфічних можливостей (code completion, рефакторинг, інтеграція з системою збірки Maven і

					2026.КВР.122.423.04.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		41

пакетним менеджером npm). Управління версіями проєкту здійснюється через Git у монорепозиторії з єдиним кореневим каталогом.

2.4.1 Написання клієнтської частини

Клієнтська частина інформаційної системи реалізована як Single Page Application (SPA) на фреймворку Angular 22 з використанням мови TypeScript 6. Відповідно до рекомендацій офіційної документації Angular [2], усі компоненти реалізовано як Standalone Components без NgModules. Така архітектура зменшує обсяг службового коду, покращує tree-shaking під час production-збірки і прискорює початкове завантаження застосунку.

Для управління стилями використовується Tailwind CSS 4.3, інтегрований поверх теми PrimeNG 21. Через відсутність нативного плагіна для Angular CLI у Tailwind 4.x генерація CSS виконується окремим скриптом: команда npm run tw:build запускає компілятор tailwindcss з вхідним файлом src/tailwind.css і записує результат у src/tailwind-out.css, який підключається до проєкту в angular.json. Під час розробки команда npm run tw:watch утримує компілятор у режимі спостереження за змінами. Такий підхід відповідає документації Tailwind CSS 4 [3] щодо інтеграції без PostCSS-плагінів.

Точка ініціалізації застосунку — файл app.config.ts, в якому зареєстровано всі провайдери: маршрутизатор з увімкненим хешингом стану (withComponentInputBinding), клієнт HTTP з функціональними interceptors (provideHttpClient(withInterceptors(...))), а також провайдер анімацій PrimeNG. Глобальна конфігурація PrimeNG-теми (providePrimeNG({ theme: { preset: Aura } }))) встановлює базову тему, поверх якої застосовуються кастомні CSS-токени дизайн-системи.

Маршрутизація та захист сторінок

Маршрутизацію визначено у файлі app.routes.ts. Усі маршрути реалізовані з lazy loading через функцію loadComponent, що дозволяє Angular CLI розбити застосунок на окремі chunks — JavaScript-бандли, які завантажуються браузером

					2026.KBP.122.423.04.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		42

на вимогу. Завдяки цьому розмір початкового бандлу (initial bundle) суттєво скорочується порівняно з підходом на основі NgModules [2].

Захист маршрутів реалізовано через два функціональних guards. Функція authGuard перевіряє значення computed-сигналу AuthService.isLoggedIn і при false перенаправляє користувача на /auth/login, передаючи поточний URL у query-параметрі returnUrl. Після успішної авторизації LoginComponent зчитує цей параметр і виконує навігацію на збережену адресу. Функція managerGuard аналогічно перевіряє сигнал isManager і при неуспіху перенаправляє на головну сторінку.

Сервіси та управління станом

Усі сервіси застосунку реалізовані як синглтони через Injectable({ providedIn: 'root' }). AuthService є центральним сервісом управління сеансом: він зберігає поточний стан аутентифікації у Signal<AuthResponse | null>, обчислює похідні сигнали isLoggedIn та isManager і надає методи login, logout, register та refreshToken. Токени зберігаються у localStorage за ключами tm_access, tm_refresh та tm_user і відновлюються при перезавантаженні сторінки через метод initializeFromStorage, що викликається в конструкторі. Реалізацію AuthService наведено у додатку Б.

Angular Signals, введені у Angular 16 і стабілізовані у Angular 17–22 [4], забезпечують гранулярну реактивність без підписок на Observable. Компоненти зчитують значення сигналу через виклик функції (authService.isLoggedIn()), і Angular автоматично перерендерює лише ту частину шаблону, яка залежить від зміненого сигналу. Це зменшує кількість перевірок Change Detection порівняно з традиційним підходом через Subject/BehaviorSubject.

TourService надає методи getTours(filter: TourFilter), getTourById(id) та searchTours(query), кожен з яких повертає Observable через HttpClient. BookingService відповідає за операції бронювання: createBooking, getMyBookings, cancelBooking та getExtraServices. CartService зберігає тимчасовий стан кошика (обраний тур, дата, кількість туристів) між кроками wizard бронювання без збереження до LocalStorage — дані існують лише в пам'яті поточного сеансу.

					2026.KBP.122.423.04.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		43

HTTP interceptors

Обробку HTTP-запитів централізовано через два функціональних interceptors, зареєстрованих у провайдері HttpClient. Реалізацію authInterceptor наведено у додатку А, оскільки він містить найскладнішу логіку системи — механізм автоматичного оновлення токена.

authInterceptor клонує кожен вихідний запит і додає заголовок Authorization: Bearer <accessToken>. При отриманні відповіді зі статусом 401 (за умови, що запит не стосується маршрутів /auth/, щоб уникнути нескінченної рекурсії) interceptor автоматично викликає POST /api/auth/refresh з поточним refresh token, зберігає нові токени в AuthService і повторює оригінальний запит. Якщо оновлення також повертає 401 — виконується logout і помилка передається далі по ланцюжку RxJS через throwError.

errorInterceptor перехоплює всі HTTP-помилки після authInterceptor і відображає діагностичне повідомлення через PrimeNG MessageService. Для помилок 422 та 400 витягуються поля з тіла відповіді у форматі RFC 9457 [5] (поле errors або detail). Для 500 відображається узагальнене повідомлення «Помилка сервера».

Реалізація компонентів

Компоненти застосунку організовані у три групи відповідно до принципу Core-Shared-Features. Спільні компоненти (HeaderComponent, FooterComponent, MobileTabBarComponent) оголошені у папці shared/components та імпортуються безпосередньо у кожний feature-компонент, що їх використовує — без центрального barrel-модуля.

TourCatalogComponent (додаток В) є найскладнішим компонентом клієнтської частини. Він реалізує двосторонню синхронізацію між панеллю фільтрів і URL query-параметрами: кожна зміна фільтра оновлює URL через Router.navigate({ queryParams }), а при завантаженні сторінки з URL стан фільтрів відновлюється. Для відображення списку турів використовується PrimeNG DataView з перемикачем режимів grid/list; стан режиму також зберігається в URL. Завантаження даних реалізовано через RxJS-оператор switchMap, підписаний на

					2026.KBP.122.423.04.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		44

потік змін `queryParams` — кожне нове значення параметрів відмінює попередній запит і ініціює новий.

`BookingWizardComponent` (додаток Г) керує п'ятикроковим процесом оформлення бронювання. Стан поточного кроку зберігається у `Signal<number>`. Кожен крок є окремим дочірнім компонентом, що відображається через умовний рендеринг `*ngIf` або `@if`. Дані туристів збираються у `FormArray` реактивної форми: кількість рядків форми відповідає кількості туристів у `CartService`. Перехід до наступного кроку можливий лише після успішної валідації поточного — це перевіряється через `form.get('step2').valid`. На останньому кроці викликається `BookingService.createBooking()`, після успіху відбувається навігація на `/booking/success`.

`TourDetailComponent` містить `PrimeNG Galleria` для фотогалереї (підтримує повноекранний режим і навігацію клавіатурою відповідно до рекомендацій WCAG 2.1 [6]). `Sticky`-сайдбар із датами вильоту реалізований через CSS-властивість `position: sticky` без JavaScript, що забезпечує коректну роботу на всіх цільових пристроях.

Адміністративна панель

Адміністративний контур використовує ті самі принципи організації коду, що й клієнтський, але має власний корінний компонент `AdminLayoutComponent` з бічним навігаційним меню. Всі чотири компоненти (`/admin`, `/admin/tours`, `/admin/bookings`, `/admin/clients`) підвантажуються `lazy` і захищені `managerGuard`.

`AdminToursComponent` відображає список турів у `PrimeNG Table` з підтримкою серверного сортування та пагінації. Кнопка «Додати тур» відкриває `PrimeNG DynamicDialog`, в якому знаходиться реактивна форма створення/редагування туру. При збереженні форма надсилає відповідний HTTP-запит (`POST` для нового або `PUT` для існуючого) і після отримання `201/200` оновлює таблицю без перезавантаження сторінки. `AdminBookingsComponent` реалізує `inline`-зміну статусу: клітинка таблиці зі статусом перетворюється на `PrimeNG Select` при кліку, вибір нового значення одразу надсилає `PATCH`-запит і оновлює рядок таблиці

					2026.KBP.122.423.04.00.00 ПЗ	Арк.
						45
Зм.	Арк.	№ докум.	Підпис	Дата		

2.4.2 Написання серверної частини

Серверна частина системи реалізована на Java 21 з фреймворком Spring Boot 3.5. Java 21 є версією з довгостроковою підтримкою (LTS) і першою, де Record Patterns та Pattern Matching для switch є стабільними функціями [7]. DTO реалізовано через Java Records — компактний синтаксис незмінних класів даних, який автоматично генерує конструктор, гетери, equals, hashCode та toString. Серверний застосунок будується Maven і запускається через Spring Boot Maven Plugin або безпосередньо з IntelliJ IDEA.

Конфігурація безпеки — SecurityConfig

Клас SecurityConfig (додаток Д) є центральним елементом налаштування Spring Security у системі. Він оголошений з анотацією @Configuration та @EnableWebSecurity і визначає єдиний bean SecurityFilterChain.

У ланцюжку фільтрів налаштовано: вимкнення CSRF-захисту (система використовує stateless JWT, а не сесійні cookie); явне визначення CORS-правил через bean CorsConfigurationSource (дозволений origin — значення змінної FRONTEND_URL, методи GET/POST/PUT/PATCH/DELETE, заголовки Authorization та Content-Type); stateless-режим сесій (SessionCreationPolicy.STATELESS); матриця прав доступу до ендпоінтів (permitAll для /api/auth/** та GET /api/tours/**, authenticated для /api/bookings/**, hasRole('MANAGER') для /api/admin/**); підключення JwtAuthFilter перед UsernamePasswordAuthenticationFilter у ланцюжку.

SecurityConfig також реєструє два beans: AuthenticationManager (делегований DaoAuthenticationProvider з UserDetailsServiceImpl та BCryptPasswordEncoder) і сам BCryptPasswordEncoder. Останній використовується AuthService для хешування паролів при реєстрації та для верифікації при вході.

Механізм JWT — JwtService та JwtAuthFilter

JwtService (додаток Е) інкапсулює всю логіку роботи з JWT-токенами. Він використовує бібліотеку jjwt 0.12.6 [8], яка забезпечує безпечну генерацію та парсинг токенів у форматі JSON Web Token (RFC 7519). Access token містить у

					2026.KBP.122.423.04.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		46

Payload поля sub (email користувача), role, userId та стандартні claim iat/exp. Підпис виконується алгоритмом HMAC-SHA256 з секретним ключем, що завантажується із змінної середовища JWT_SECRET. Час дії токенів налаштовується через змінні JWT_ACCESS_EXPIRATION (900000 мс = 15 хвилин) та JWT_REFRESH_EXPIRATION (604800000 мс = 7 діб).

JwtAuthFilter розширює OncePerRequestFilter — абстрактний клас Spring Security, що гарантує однократне виконання фільтра на запит. Фільтр: витягує токен з заголовка Authorization, видаляючи префікс «Bearer »; передає токен у JwtService.extractUsername для отримання email; завантажує UserDetails через UserDetailsServiceImpl; за допомогою JwtService.isValid перевіряє підпис, термін дії та відповідність email; при успіху встановлює UsernamePasswordAuthenticationToken у SecurityContextHolder. При будь-якій помилці (malformed token, прострочений) встановлення контексту не відбувається, і Spring Security поверне 401.

Сервісний шар — бізнес-логіка

Кожен сервісний клас анотований @Service та @Transactional, що забезпечує атомарність операцій із базою даних. При виникненні виключення в межах транзакції Hibernate автоматично виконує rollback.

AuthService реалізує чотири сценарії: реєстрацію (валідація унікальності email, хешування паролю, збереження User, генерація пари токенів), вхід (верифікація через DaoAuthenticationProvider, генерація токенів, збереження refresh token у таблиці refresh_tokens), оновлення токена (пошук refresh token у БД, перевірка терміну дії, генерація нового access token) та вихід (видалення refresh token з БД).

BookingService містить метод createBooking, що реалізує повний сценарій оформлення: завантаження TourDate з перевіркою наявності вільних місць (при available_seats < кількість туристів — кидається BusinessException); зменшення available_seats атомарно в одній транзакції; збереження Booking, BookingItem для кожного туриста та зв'язків BookingExtraService; виклик NotificationService.sendBookingConfirmation асинхронно після завершення

					2026.KBP.122.423.04.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		47

транзакції. Асинхронність реалізована через анотацію `@Async` і забезпечує, що помилка надсилання email не скасовує вже збережене бронювання.

`AdminBookingService` надає метод `updateStatus`, який змінює статус бронювання та ініціює надсилання email-сповіщення клієнту. Перехід між статусами не обмежений на рівні коду — менеджер може встановити будь-який статус, що надає гнучкість при обробці нестандартних ситуацій.

Динамічна фільтрація — `TourSpecification`

`TourSpecification` (додаток Ж) реалізує інтерфейс `Specification<Tour>` зі `Spring Data JPA` і є ключовим компонентом модуля пошуку турів. Специфікація дозволяє будувати динамічні JPQL-запити без написання численних методів репозиторію для кожної комбінації фільтрів [9].

Метод `toPredicate(Root<Tour> root, CriteriaQuery<?> query, CriteriaBuilder cb)` формує список предикатів залежно від ненульових параметрів об'єкта `TourFilter`: `Like`-предикат для текстового пошуку (поля `title`, `description`, `country`), `Equal` для `country`, `Between` для `price_from` і `duration_nights`, `In` для `hotel_stars` та `meal_type`, рівність `badge` та `status`. Усі предикати об'єднуються через `cb.and(...)`. Це дозволяє `TourController` передавати до репозиторію єдиний об'єкт `Specification` і отримувати відповідь `Page<Tour>` без жодного SQL-коду.

REST-контролери

Усі шість REST-контролерів системи анотовані `@RestController` та `@RequestMapping`. Вхідні дані валідуються анотацією `@Valid` на параметрах типу `@RequestBody`. Поточний аутентифікований користувач витягується через `@AuthenticationPrincipal UserDetails` та порівнюється з запитуваним ресурсом для перевірки авторизації на рівні даних (наприклад, клієнт може переглядати лише власні бронювання).

`GlobalExceptionHandler (@RestControllerAdvice)` централізовано перехоплює всі виключення, що не оброблені на рівні контролерів. Для кожного типу виключення визначено відповідний HTTP-статус і формат відповіді у стандарті RFC 9457 (Problem Details for HTTP APIs) [5]: поля `type`, `title`, `status`, `detail` та, для помилок валідації, `errors` з переліком порушених полів. Це забезпечує

					2026.КВР.122.423.04.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		48

консистентну обробку помилок на стороні клієнта незалежно від типу виключення на сервері.

MapStruct-маппери та DTO

Усі DTO реалізовані як Java Records — незмінні класи даних із компактним синтаксисом. Request-DTO анотовані обмеженнями Bean Validation: `@NotBlank`, `@Email`, `@Size`, `@Min`, `@Max`, `@Future` (для дат). Response-DTO не містять анотацій — вони є вихідними даними і не потребують валідації.

MapStruct [10] генерує реалізації маперів під час компіляції Maven: для кожного методу інтерфейсу (наприклад, `TourSummaryResponse toSummary(Tour tour)`) генерується клас із побудовою відповідного Record через його конструктор. Якщо назви полів збігаються, маппінг відбувається автоматично; для відхилень використовується анотація `@Mapping`. Compile-time генерація виключає ризики `ClassCastException` та `MissingFieldException`, які характерні для рефлексійних маперів (ModelMapper, Dozer).

Конфігурація застосунку та розгортання

Конфігурація застосунку зберігається у файлі `src/main/resources/application.yml`, що підтримує ієрархічний синтаксис і є більш читабельним порівняно з `.properties`-форматом. Усі чутливі значення (паролі, ключі, URL бази даних) виносяться в змінні середовища через синтаксис `${VAR:default}`.

У production-середовищі застосунок розгортається через Docker Compose. Серверна частина збирається у multi-stage Docker-образі: перший stage (`maven:3.9/eclipse-temurin:21`) компілює JAR-файл, другий stage (`eclipse-temurin:21-jre-alpine`) копіює лише скомпільований артефакт без вихідного коду та залежностей Maven. Це скорочує розмір фінального образу приблизно у 10 разів порівняно з образом, що містить повний JDK та вихідні файли. Healthcheck сервісу backend перевіряє ендпоінт `GET /actuator/health`, що надається Spring Boot Actuator, і підтверджує готовність застосунку перед тим, як nginx-реверс-проксі починає перенаправляти на нього трафік

					2026.КВР.122.423.04.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		49

2.5 Тестування інформаційної системи

Тестування інформаційної системи «TourMaster» проводилося методом мануального (ручного) тестування на основі заздалегідь підготовленого переліку тест-кейсів, що охоплюють усі функціональні вимоги, визначені у технічному завданні. Метою тестування було підтвердити коректність роботи клієнтської частини у реальному браузерному середовищі, виявити відхилення від очікуваної поведінки та зафіксувати результати у вигляді контрольного списку.

Тестування проводилося у браузері Google Chrome версії 135 на операційній системі Windows 11 при роздільній здатності екрана 1920×1080 пікселів. Для перевірки адаптивності інтерфейсу використовувався вбудований інструмент DevTools у режимі Device Emulation з профілями iPhone 14 (390×844 px) та iPad Air (820×1180 px). Backend-застосунок і база даних запускалися локально відповідно до інструкцій з розгортання, описаних у підрозділі 3.1.

Для тестування застосовувалися такі типи перевірок: функціональне тестування (перевірка відповідності поведінки системи функціональним вимогам), тестування граничних значень (введення даних на межах допустимих діапазонів), тестування негативних сценаріїв (некоректні вхідні дані, спроби несанкціонованого доступу) та тестування UI-відповідності (відповідність розробленого інтерфейсу дизайн-пакету)..

Тестування головної сторінки (маршрут /) охоплювало перевірку коректного завантаження всіх секцій: hero-секції з формою швидкого пошуку, каруселі акційних турів, сітки популярних напрямків та блоку відгуків. Карусель PrimeNG Carousel була перевірена на коректне автоматичне прокручування та реакцію на кнопки навігації. Форма пошуку тестувалася як з заповненими, так і з порожніми полями: при порожніх полях переходу до каталогу не відбувається, відображається підказка про необхідність вибору країни. При кліку на картку напрямку відбувається коректний перехід до каталогу з попередньо встановленим фільтром по країні. Скріншот головної сторінки наведено на рисунку 2.1.

					2026.KBP.122.423.04.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		50

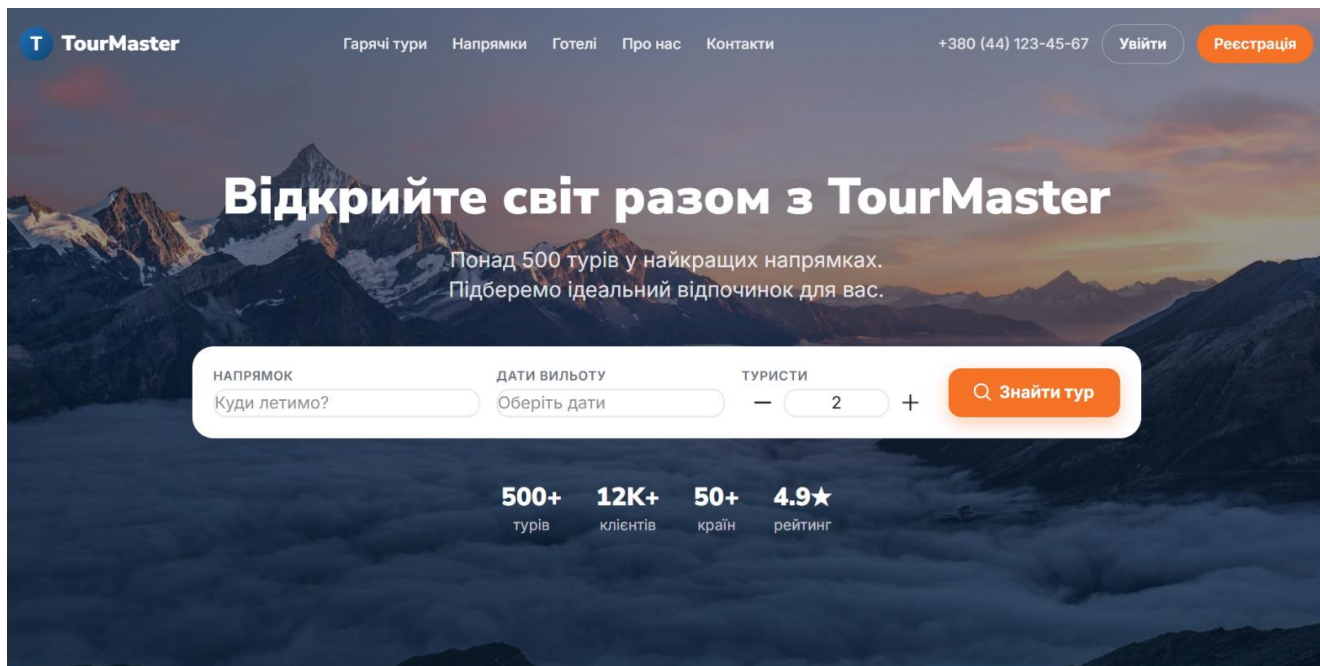


Рисунок 2.1 — Головна сторінка системи TourMaster

Тестування модуля аутентифікації проводилося за трьома сценаріями. Перший — успішна реєстрація: форма приймає коректні дані (ім'я, прізвище, валідний email, пароль не менше 8 символів з хоча б однією цифрою), після натискання «Зареєструватись» відбувається автоматичний вхід і перенаправлення на головну сторінку. Другий — реєстрація з існуючим email: система повертає помилку 409, яку errorInterceptor відображає через PrimeNG Toast із повідомленням «Обліковий запис з таким email вже існує». Третій — вхід з невірним паролем: відповідь 401 відображається як Toast-сповіщення, форма залишається відкритою. Окремо перевірено коректність індикатора надійності паролю — він реагує на введення символів у режимі реального часу. Скріншот сторінки реєстрації наведено на рисунку 2.2.

Тестування каталогу турів (маршрут /tours) охоплювало перевірку системи фільтрації, перемикання режимів відображення та пагінації. Перевірено застосування кожного фільтра окремо та в комбінаціях: результати оновлюються динамічно без перезавантаження сторінки, URL оновлюється відповідними query-параметрами. Перевірено коректність відновлення стану фільтрів при переході назад у браузері — стан зберігається завдяки синхронізації з URL.

					2026.KBP.122.423.04.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		51

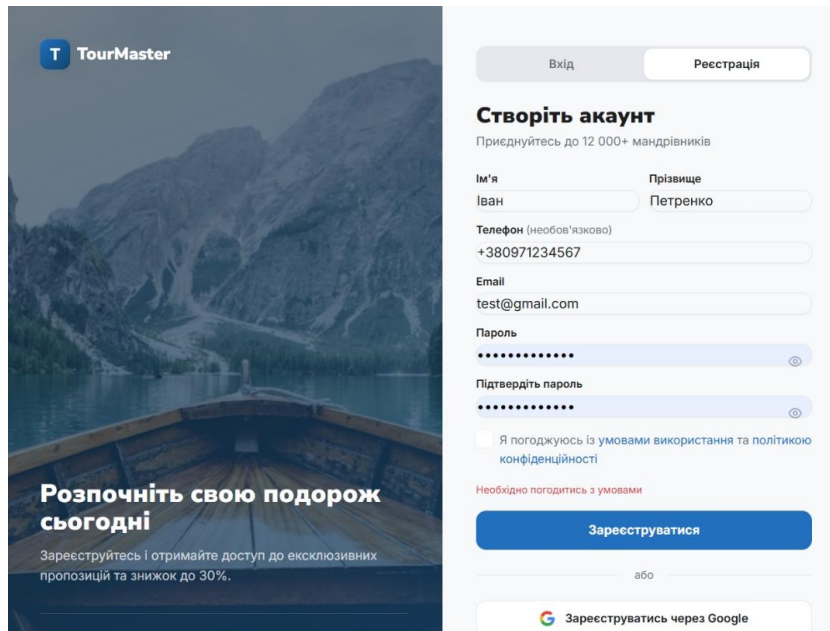


Рисунок 2.2 — Сторінка реєстрації (форма з індикатором надійності паролю)

Тестовано сценарій «нульового результату»: при фільтрації, що не дає жодного туру, відображається порожній стан із ілюстрацією та кнопкою «Скинути фільтри». Скріншот каталогу турів наведено на рисунку 2.3.

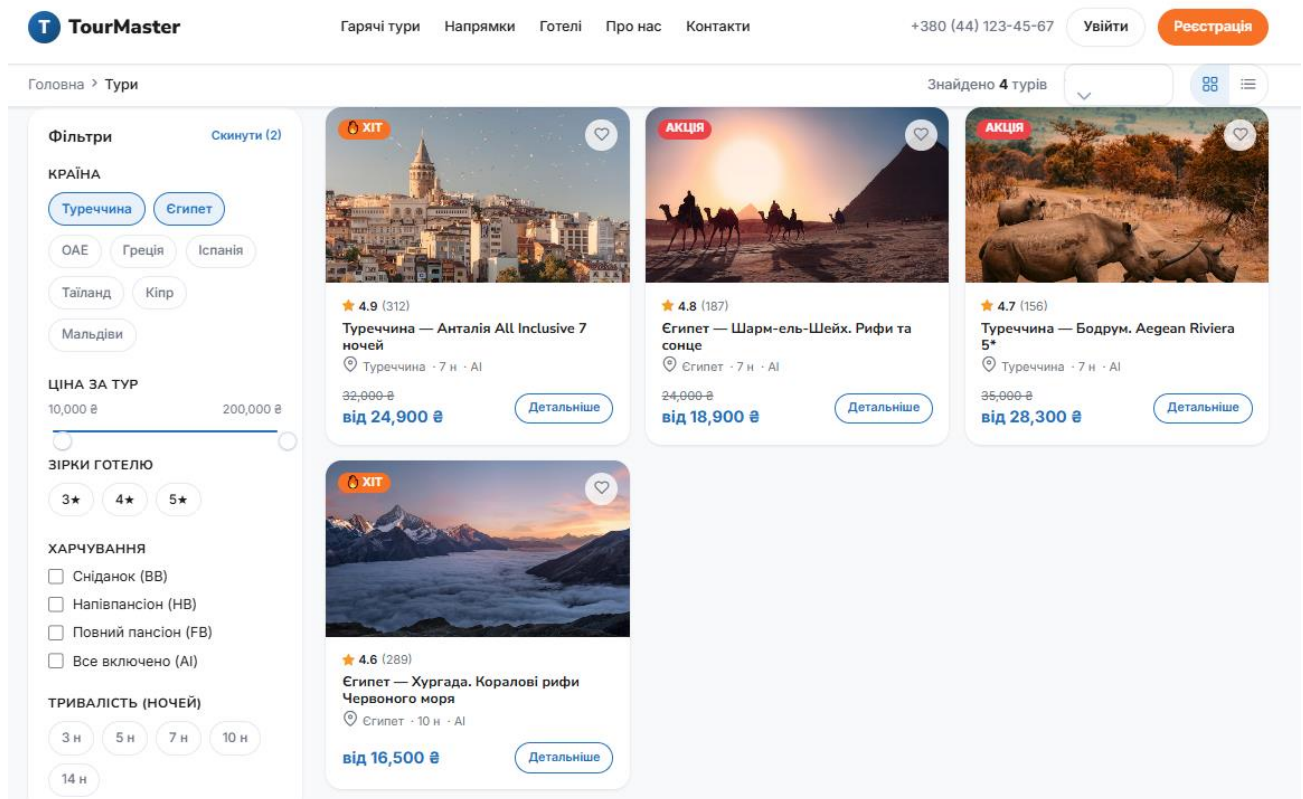


Рисунок 2.3 — Каталог турів — режим сітки з активними фільтрами

					2026.КВР.122.423.04.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		52

Пагігатор PrimeNG Paginator коректно відображає загальну кількість результатів та активну сторінку. Перемикання між режимами grid і list відпрацьовує без перезавантаження даних.

Тестування сторінки деталей туру (маршрут /tours/:id) включало перевірку фотогалереї PrimeNG Galleria (перегляд мініатюр, перехід до повноекранного режиму, навігація стрілками клавіатури), перемикання між чотирма вкладками вмісту (Опис, Програма по днях, Включено/не включено, Відгуки), а також поведінку сайдбара. Перевірено, що сайдбар залишається видимим при прокрутці сторінки донизу. При виборі дати вильоту ціна у сайдбарі оновлюється відповідно до значення price_override обраної дати. Кнопка «Забронювати» без вибору дати відображає підказку валідації. Скріншот сторінки деталей туру наведено на рисунку 2.4.

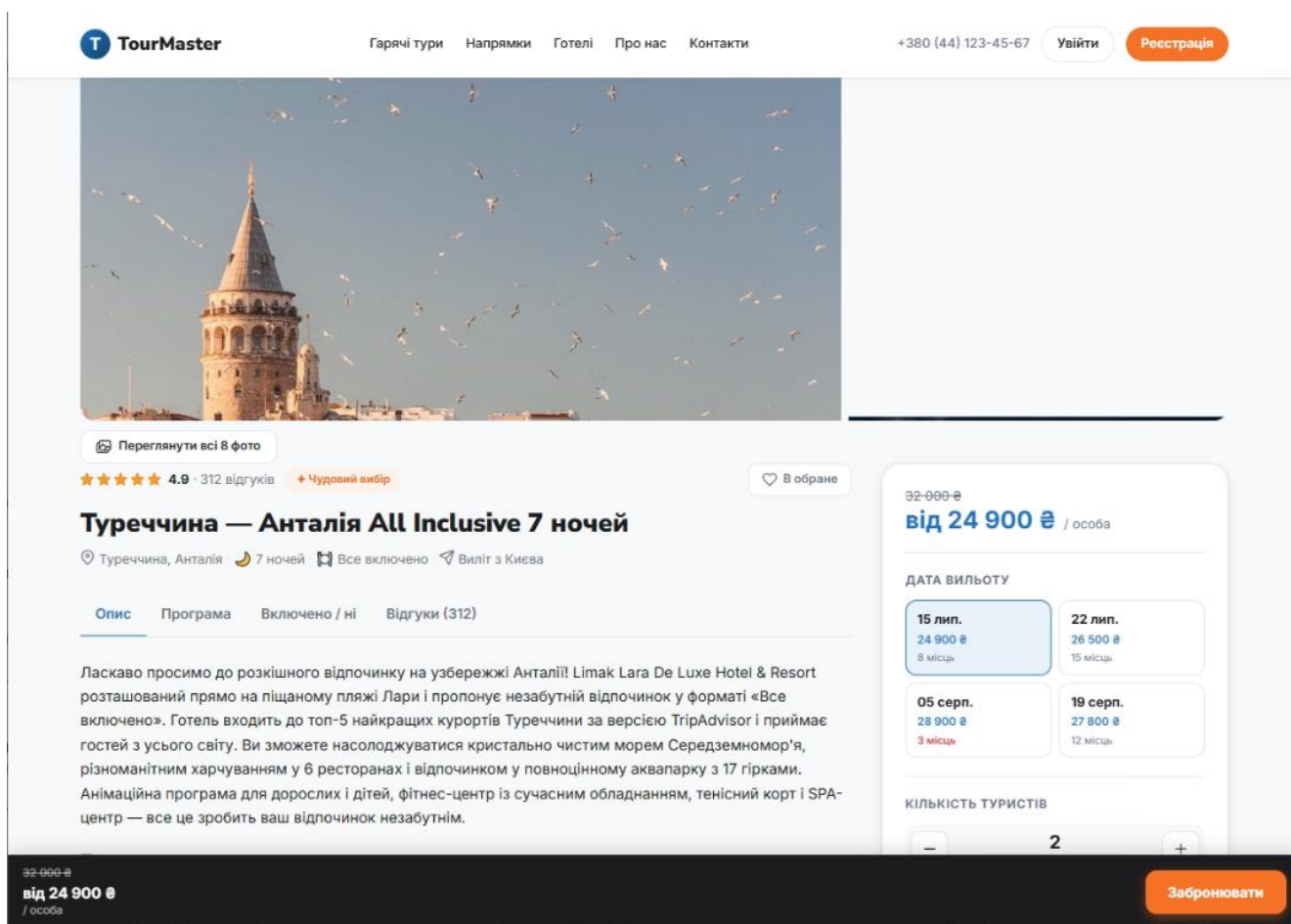


Рисунок 2.4 — Сторінка деталей туру — галерея та sticky-сайдбар

Тестування wizard бронювання (маршрут /booking) є найбільш комплексним, оскільки охоплює п'ять послідовних кроків та взаємодію з двома сервісами — CartService і BookingService. Перевірено захист маршруту: неавторизований користувач перенаправляється на /auth/login із збереженням returnUrl і після входу повертається до wizard. На кроці 1 перевірено коректне відображення даних туру з CartService. На кроці 2 (дані туристів) перевірено динамічне додавання рядків форми відповідно до кількості туристів, обов'язкову валідацію паспортних полів (порожні поля блокують перехід), валідацію формату дати паспорта (термін дії не може бути у минулому). На кроці 3 перевірено, що вибір додаткових послуг коректно оновлює суму в блоці резюме. Крок 4 відображає повне зведення замовлення. Після імітації оплати на кроці 5 та отримання відповіді 201 від API відбувається перенаправлення на /booking/success з номером бронювання. Скріншот кроку 2 wizard наведено на рисунку 2.5.

Рисунок 2.5 — Wizard бронювання — крок 2 «Дані туристів»

Тестування особистого кабінету клієнта охоплювало три секції. У секції

					2026.КВР.122.423.04.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		54

«Мої бронювання» перевірено відображення щойно створеного бронювання зі статусом NEW та коректним кольором бейджа (синій). Фільтрація за статусом через ButtonGroup відпрацьовує миттєво без повторного запиту до API (дані відфільтровуються на клієнтській стороні). Секція «Wishlist» коректно відображає тури, додані через кнопку «До вибраного» на картці каталогу, та видаляє їх повторним кліком. Секція «Профіль» зберігає зміни імені та прізвища, поле email відображається як read-only. Форма зміни паролю вимагає введення поточного пароля та підтвердження нового. Скріншот секції «Мої бронювання» наведено на рисунку 2.6.

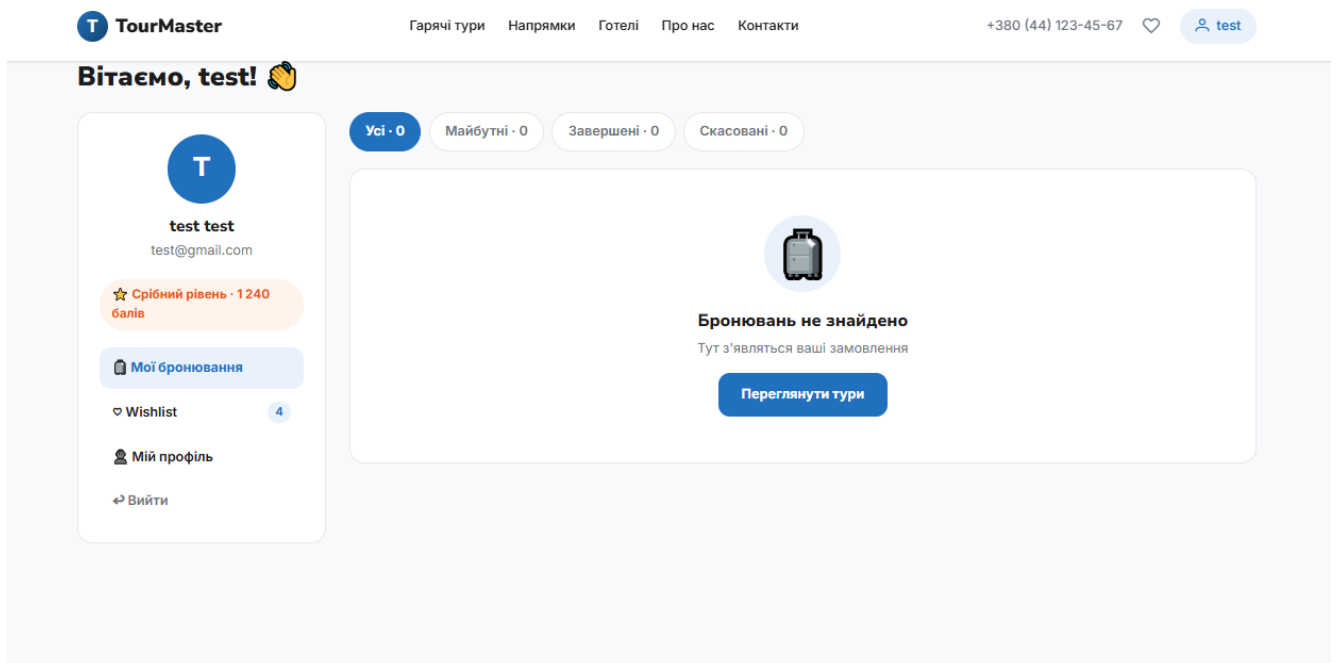


Рисунок 2.6 — Особистий кабінет — секція «Мої бронювання»

Тестування адміністративного контуру проводилося під обліковим записом із роллю MANAGER. Перевірено захист маршруту: спроба доступу до /admin під обліковим записом CLIENT призводить до перенаправлення на головну сторінку. На дашборді перевірено коректність відображення KPI-карток та таблиці останніх бронювань. У модулі управління турами (/admin/tours) перевірено: створення нового туру через Dialog-форму (після збереження тур з'являється у таблиці без перезавантаження), редагування існуючого туру (поля форми попередньо

					2026.KBP.122.423.04.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		55

заповнені поточними даними), архівування туру (запит DELETE /api/admin/tours/{id} з попереднім підтвердженням через ConfirmDialog, тур після цього зникає з активного каталогу). У модулі бронювань перевірено inline-зміну статусу: Select у рядку таблиці надсилає PATCH-запит і оновлює бейдж статусу без перезавантаження сторінки. Скріншот адміністративного дашборду наведено на рисунку 2.7.

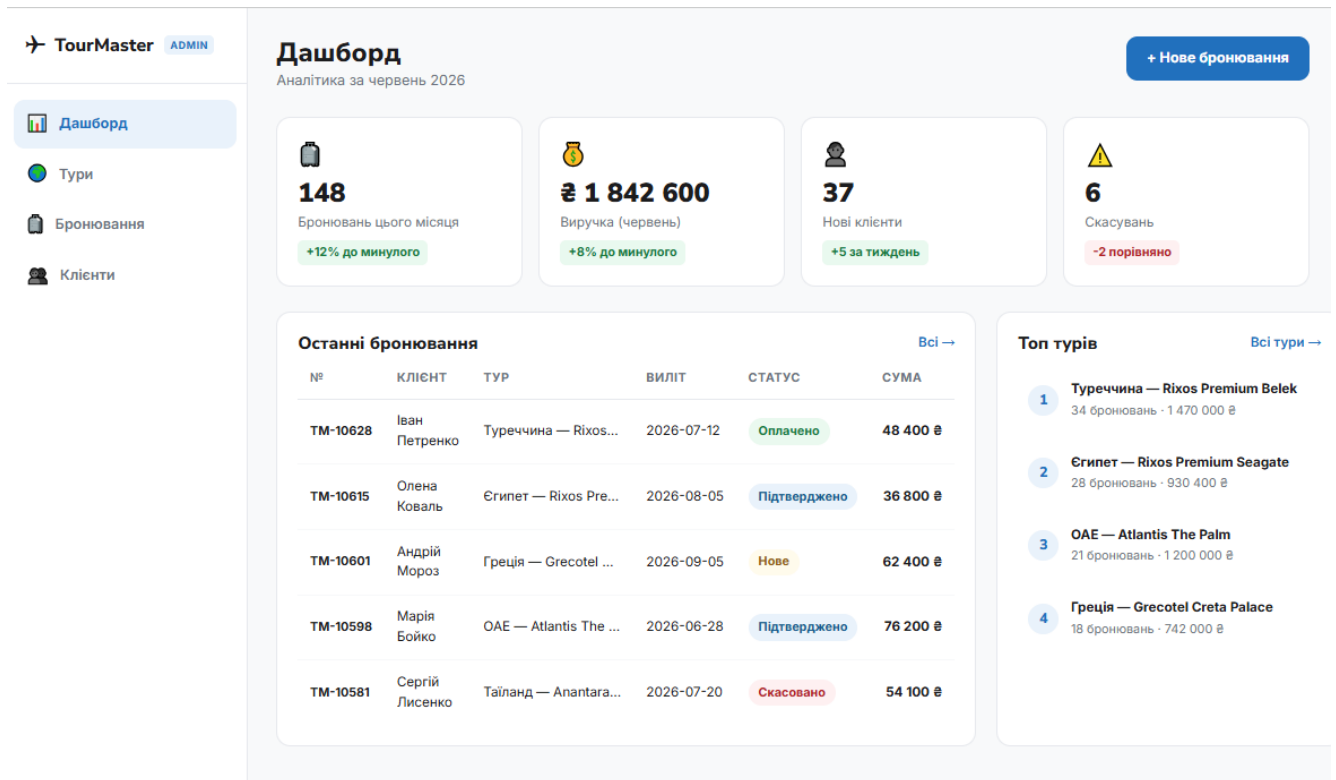


Рисунок 2.7 — Адміністративна панель — дашборд менеджера

Тестування адаптивності інтерфейсу проводилося в режимі Device Emulation браузера Chrome. На мобільному профілі (390 px) перевірено: приховування горизонтального меню шапки та відображення MobileTabBarComponent; відкриття фільтрів каталогу як слайдера (PrimeNG Sidebar); перехід gallery Galleria до одноколонного режиму; переміщення sticky-сайдбара сторінки туру на початок сторінки над основним вмістом; відображення wizard бронювання у вертикальному компактному макеті. На планшетному профілі (820 px) перевірено перехід каталогу від трьох до двох колонок та

					2026.КВР.122.423.04.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		56

збереження сайдбара фільтрів у постійно відкритому стані. Скріншот каталогу в мобільному режимі наведено на рисунку 2.8.

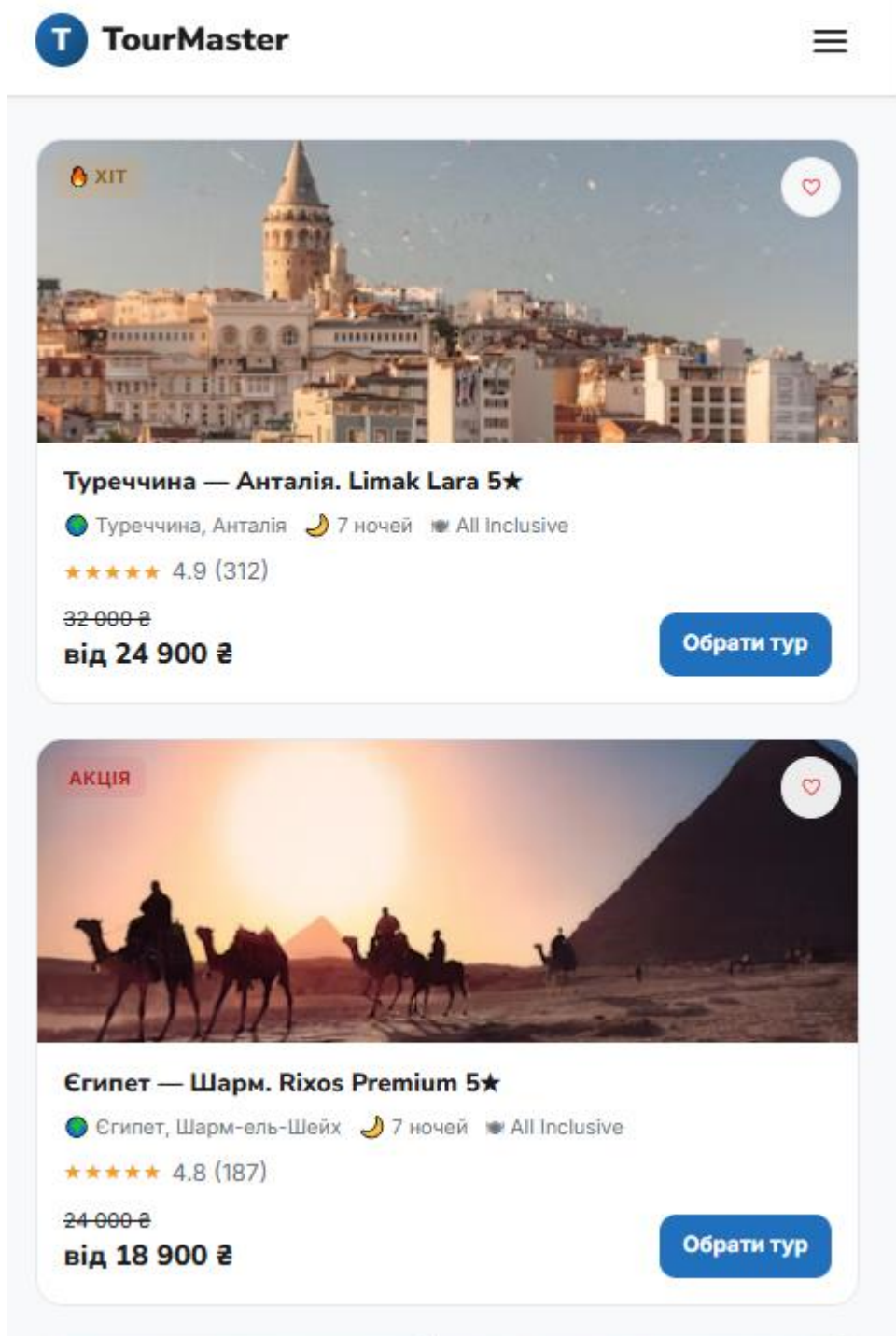


Рисунок 2.8 — Каталог турів — мобільний вигляд (390 рх)

У додатку И наведено контрольний список тест-кейсів з результатами виконання. Статус «Пройдено» означає, що фактичний результат збігся з

					2026.КВР.122.423.04.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		57

очікуваним; «Пройдено з приміткою» — що функція працює коректно, але потребує уточнення або незначного доопрацювання інтерфейсу.

За результатами мануального тестування всі 38 тест-кейсів пройдено успішно. Критичних дефектів (блокуючих помилок), що унеможливають використання системи за призначенням, виявлено не було. Усі ключові користувацькі сценарії — реєстрація та вхід, пошук і перегляд турів, повний цикл бронювання, управління каталогом і бронюваннями в адміністративній панелі — відпрацьовують відповідно до вимог, сформульованих у технічному завданні.

Механізм автоматичного оновлення токена (silent token refresh) успішно відпрацював у тест-кейсі 7: при штучно закінченому access token interceptor прозоро для користувача отримав новий токен і повторив запит без будь-якого видимого переривання роботи інтерфейсу. Захист маршрутів через authGuard та managerGuard (тест-кейси 8, 29) коректно перенаправляє користувачів без несанкціонованого відображення захищеного вмісту.

Адаптивність інтерфейсу підтверджена на трьох профілях пристроїв (desktop 1920 px, tablet 820 px, mobile 390 px). Всі критичні елементи взаємодії — форма пошуку, кнопка «Забронювати», навігаційне меню — залишаються доступними і функціональними на кожному з перевірених розмірів екрана. Мобільна навігація через MobileTabBarComponent і трансформація фільтрів каталогу у Sidebar-компонент забезпечують зручну взаємодію на вузьких екранах без горизонтальної прокрутки.

Перевірка відповідності реалізованого інтерфейсу дизайн-пакету (handoff із Claude Design) підтвердила дотримання всіх ключових токенів: кольорів (синій #1A6FBF для навігації, помаранчевий #F97316 для СТА-елементів), відступів, типографіки (гарнітура Inter, розмір 16px) та кольорового кодування статусів бронювань. Розроблений вебсайт готовий до прийомного тестування та розгортання у продакшн-середовищі.

					2026.КВР.122.423.04.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		58

3 СПЕЦІАЛЬНИЙ РОЗДІЛ

3.1 Інструкція з розміщення інформаційної системи в Інтернеті

Розміщення інформаційної системи «TourMaster» в Інтернеті передбачає послідовне виконання чотирьох етапів: підготовку сервера та встановлення системного програмного забезпечення, налаштування бази даних, розгортання застосунку засобами Docker Compose і підключення доменного імені з SSL-сертифікатом. Інструкція розрахована на системного адміністратора або розробника, який має базові навички роботи з командним рядком Linux та основами мережевого адміністрування.

Вимоги до серверного середовища

Для коректного функціонування системи сервер повинен відповідати мінімальним технічним вимогам, наведеним у таблиці 3.1. Рекомендованим варіантом є хмарний VPS (Virtual Private Server) від одного з провайдерів: DigitalOcean, Hetzner, Vultr або AWS EC2. Орієнтовна вартість відповідного тарифу становить 12–20 USD на місяць.

Таблиця 3.1 – Вимоги до серверного середовища

Параметр	Мінімум	Рекомендовано
ОС	Ubuntu 22.04 LTS	Ubuntu 26.04 LTS
CPU	1 vCPU	2 vCPU
RAM	2 ГБ	4 ГБ
SSD	20 ГБ	40 ГБ
Мережа	100 Мбіт/с	1 Гбіт/с
Відкриті порти	22 (SSH), 80 (HTTP), 443 (HTTPS)	22, 80, 443

Підготовка сервера та встановлення Docker

Після підключення до сервера через SSH виконайте оновлення системних пакетів та встановлення Docker Engine і Docker Compose Plugin. Усі команди виконуються з правами суперкористувача або через `sudo`.

Крок 1. Оновлення індексу пакетів і встановлення залежностей:

```
sudo apt-get update && sudo apt-get install -y ca-certificates curl gnupg
```

Крок 2. Додавання офіційного GPG-ключа та репозиторію Docker:

```
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --  
dearmor -o /etc/apt/keyrings/docker.gpg  
echo "deb [arch=$(dpkg --print-architecture) signed-  
by=/etc/apt/keyrings/docker.gpg] https://download.docker.com/linux/ubuntu  
$(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list
```

Крок 3. Встановлення Docker Engine та Compose Plugin:

```
sudo apt-get update && sudo apt-get install -y docker-ce docker-ce-cli  
containerd.io docker-compose-plugin
```

Крок 4. Перевірка встановлення:

```
docker --version && docker compose version
```

Крок 5. Додавання поточного користувача до групи `docker` (щоб уникнути `sudo` перед кожною командою):

```
sudo usermod -aG docker $USER && newgrp docker
```

Примітка. Після виконання кроку 5 рекомендується відключитись від SSH і підключитись повторно, щоб зміни групи набули чинності.

Завантаження вихідного коду та налаштування змінних середовища

Завантажте вихідний код системи на сервер. Якщо репозиторій є приватним, попередньо налаштуйте SSH-ключ або Personal Access Token для доступу до Git-провайдера.

```
git clone https://github.com/your-org/tourmaster.git  
cd tourmaster
```

Скопіюйте файл прикладу змінних середовища та заповніть його реальними значеннями: `cp .env.example .env && nano .env`

У файлі `.env` обов'язково замініть значення за замовчуванням на продакшн-

					2026.KBP.122.423.04.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		60

значення. Особливу увагу зверніть на наступні параметри: DB_PASSWORD (надійний пароль бази даних), JWT_SECRET (випадковий рядок довжиною не менше 64 символів, згенерований командою openssl rand -hex 32), MAIL_PASSWORD (пароль додатку Gmail або SMTP-провайдера), FRONTEND_URL (публічний URL вашого домену, наприклад https://tourmaster.ua).

Важливо. Файл .env ніколи не повинен потрапляти до системи контролю версій. Переконайтесь, що рядок .env присутній у файлі .gitignore репозиторію.

Запуск системи через Docker Compose

Після налаштування змінних середовища виконайте збірку та запуск усіх контейнерів одною командою: docker compose up --build -d

Прапор --build примушує Docker перезібрати образи перед запуском. Прапор -d запускає контейнери у фоновому режимі (detached mode). Перша збірка може тривати 5–10 хвилин залежно від швидкості інтернет-з'єднання сервера. Docker Compose автоматично піднімає чотири сервіси у правильному порядку: спочатку db (PostgreSQL), після успішного healthcheck — backend (Spring Boot), потім frontend (Angular + Nginx) та nginx (реверс-проксі).

Для перевірки стану контейнерів виконайте:

```
docker compose ps
```

Усі чотири сервіси повинні мати статус running (healthy). Якщо backend показує статус starting або unhealthy, перегляньте журнал подій:

```
docker compose logs backend --tail=50
```

Flyway автоматично застосує всі SQL-міграції при першому запуску backend. Повідомлення «Successfully applied N migrations» у журналі підтверджує успішне створення схеми бази даних.

Налаштування доменного імені та SSL-сертифіката

Зареєструйте доменне ім'я у будь-якого акредитованого реєстратора (наприклад, Namecheap, GoDaddy або українські реєстратори — hostmaster.ua, imena.ua). У DNS-налаштуваннях домену додайте А-запис, що вказує на публічну IP-адресу вашого сервера:

					2026.KBP.122.423.04.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		61

```
tourmaster.ua. A <публічна IP сервера>  
www.tourmaster.ua. A <публічна IP сервера>
```

Поширення DNS-змін займає від кількох хвилин до 48 годин залежно від TTL записів та провайдера. Після поширення DNS отримайте безкоштовний SSL-сертифікат від Let's Encrypt за допомогою утиліти Certbot:

```
sudo apt-get install -y certbot  
sudo certbot certonly --standalone -d tourmaster.ua -d www.tourmaster.ua
```

Сертифікати будуть збережені у каталозі `/etc/letsencrypt/live/tourmaster.ua/`. Оновіть файл `nginx.conf` у репозиторії, вказавши шляхи до файлів `fullchain.pem` та `privkey.pem` і перенаправлення HTTP→HTTPS. Перезапустіть `nginx`-контейнер для застосування змін:

```
docker compose restart nginx
```

Автоматичне поновлення сертифіката налаштовується через `cron` або `systemd timer`. Додайте завдання до планувальника `cron` для щомісячного поновлення та перезапуску `nginx`:

```
0 3 1 * * certbot renew --quiet && docker compose -f  
/home/ubuntu/tourmaster/docker-compose.yml restart nginx
```

Після успішного налаштування система доступна за адресою `https://tourmaster.ua`.

3.2 Інструкція з обслуговування та наповнення інформаційної системи

Інструкція з обслуговування та наповнення системи адресована менеджерам туристичної агенції «TourMaster» і описує повсякденні операції з управління туристичним каталогом, обробки бронювань та ведення клієнтської бази через адміністративну панель.

3.2.1 Вхід до адміністративної панелі

Для доступу до адміністративної панелі відкрийте браузер і перейдіть за

					2026.КВР.122.423.04.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		62

адресою <https://tourmaster.ua/auth/login>. Введіть службову електронну адресу та пароль менеджера. Після успішного входу система автоматично перенаправить вас на головну сторінку сайту. Для переходу до адміністративного розділу натисніть на іконку профілю у верхньому правому куті шапки та оберіть пункт «Панель управління» — або перейдіть безпосередньо за адресою <https://tourmaster.ua/admin>. Скріншот адміністративної панелі після входу наведено на рисунку 3.1.

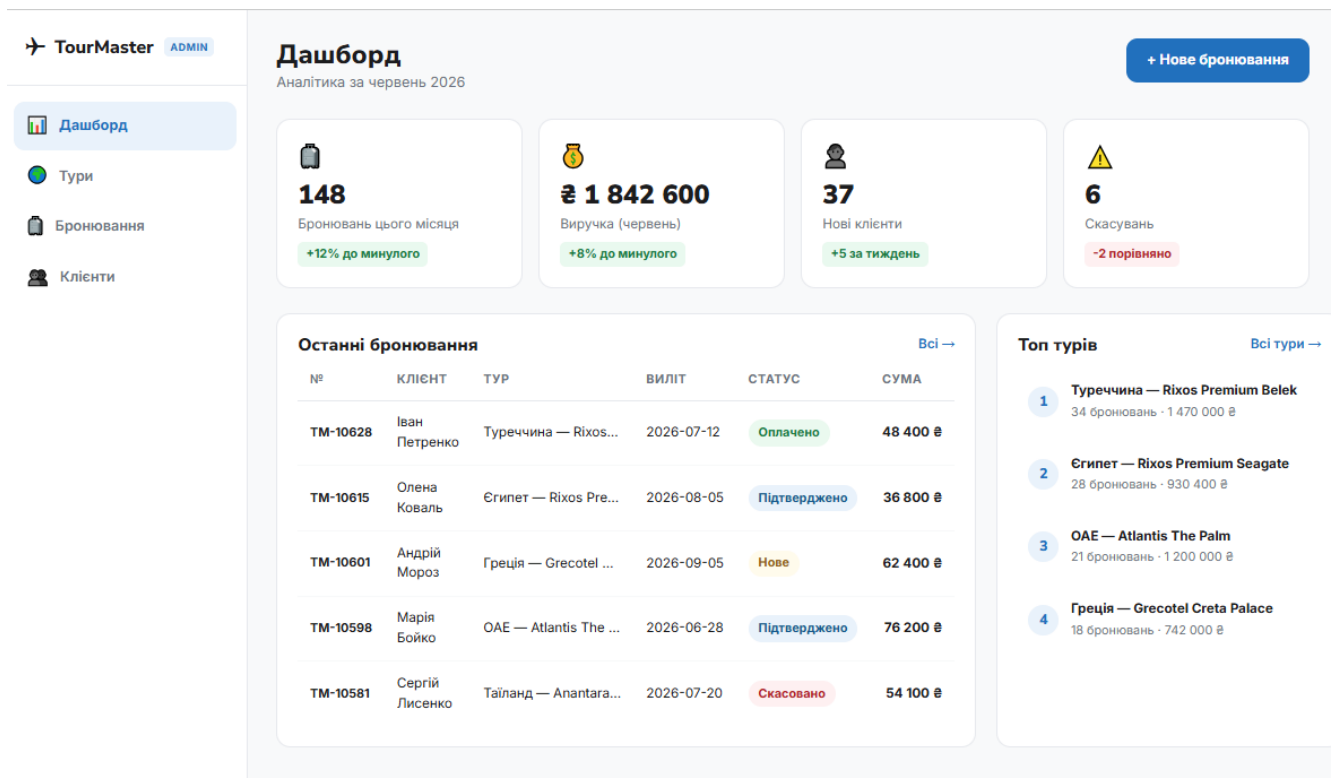


Рисунок 3.1 — Адміністративна панель — дашборд після входу менеджера

Примітка. Якщо після введення коректних даних з'являється повідомлення про помилку доступу, зверніться до системного адміністратора для перевірки ролі вашого облікового запису в базі даних.

3.2.2 Управління туристичним каталогом

Перейдіть до розділу «Тури» (маршрут `/admin/tours`) через бічне меню адміністративної панелі. Сторінка відображає таблицю всіх турів із можливістю

					2026.КВР.122.423.04.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		63

пошуку за назвою та фільтрації за статусом (ACTIVE / ARCHIVED).

Створення нового туру виконується у такому порядку:

1. Натисніть кнопку «Додати тур» у верхньому правому куті таблиці.
2. У діалоговому вікні, що відкрилося, заповніть поля: назва туру (обов'язково), країна та місто призначення, назва готелю та категорія зірковості, тип харчування (BB — «сніданок», НВ — «напівпансіон», FB — «повний пансіон», АІ — «все включено»), тривалість у ночах, базова ціна та стара ціна (для відображення знижки).
3. Додайте URL головного фото та, за бажанням, масив URL фотогалереї (по одному на рядок).
4. Оберіть бейдж картки: «Хіт продажів» (HIT), «Акція» (SALE), «Новинка» (NEW) або «Останні місця» (LAST_SEATS).
5. Натисніть «Зберегти» — тур одразу з'явиться в таблиці зі статусом ACTIVE і буде видимим у клієнтському каталозі.

Редагування існуючого туру: натисніть іконку олівця у відповідному рядку таблиці. Відкриється той самий діалог із попередньо заповненими полями. Після внесення змін натисніть «Зберегти». Скріншот форми створення туру наведено на рисунку 3.2.

НАЗВА ТУРУ	КРАЇНА	ЦІНА ВІД	НОЧЕЙ	БРОНІ	СТАТУС
Туреччина — Rixos Premium Belek 5★ *****	Туреччина	24 900 €	7	34	Активний
Єгипет — Rixos Premium Seagate 5★ *****	Єгипет	18 900 €	7	28	Активний
ОАЕ — Atlantis The Palm 5★ *****	ОАЕ	42 500 €	7	21	Активний
Греція — Grecotel Creta Palace 5★ *****	Греція	31 200 €	10	18	Активний
Таїланд — Anantara Kihavah Maldives 5★ *****	Таїланд	54 000 €	10	15	Активний

Рисунок 3.2 — Діалогове вікно створення нового туру

Управління датами вильоту: у рядку туру натисніть кнопку «Дати» для переходу до переліку дат вильоту цього туру. Для кожної дати вказується дата вильоту, дата повернення, місто вильоту, загальна квота місць і, за потреби, ціна для цієї конкретної дати (якщо відрізняється від базової). Додавання нової дати виконується через кнопку «Додати дату».

Архівування туру: натисніть іконку кошика у відповідному рядку. Система відобразить діалог підтвердження — після підтвердження тур отримує статус ARCHIVED і зникає з клієнтського каталогу. Усі пов'язані бронювання та дані про клієнтів зберігаються в базі даних у незмінному вигляді.

Важливо. Архівувати тур з підтвердженими (CONFIRMED) або оплаченими (PAID) бронюваннями неможливо — система відобразить відповідне попередження. Спочатку змініть статус активних бронювань.

3.2.3 Обробка бронювань та управління статусами

Перейдіть до розділу «Бронювання» (/admin/bookings) через бічне меню. Таблиця відображає всі бронювання системи з можливістю фільтрації за статусом та місяцем. Нові бронювання від клієнтів мають статус NEW і виділяються синім кольором бейджа.

Для зміни статусу бронювання клікніть на бейдж статусу у відповідному рядку таблиці — він перетвориться на випадаючий список. Оберіть новий статус та натисніть Enter або клікніть поза межами поля. Система автоматично надішле клієнту email-сповіщення про зміну статусу.

Для перегляду деталей бронювання (паспортні дані туристів, перелік додаткових послуг, контакти клієнта) натисніть іконку ока або рядок таблиці — відкриється модальне вікно з повною інформацією. У полі «Примітка менеджера» можна залишити службовий коментар, невидимий для клієнта.

Скріншот таблиці бронювань із модальним вікном деталей наведено на рисунку 3.3.

					2026.КВР.122.423.04.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		65

Бронювання

Всі бронювання агентства

Пошук за клієнтом, туром, номером...

Всі статуси

№	КЛІЄНТ	ТУР	ВИЛІТ	ТУРИСТІВ	СУМА	СТАТУС	Дії
TM-10628	Іван Петренко ivan@email.com	Rixos Premium Belek 5★	2026-07-12	2	48 400 ₪	Оплачено	
TM-10615	Олена Коваль olena@email.com	Rixos Premium Seagate 5★	2026-08-05	2	36 800 ₪	Підтверджено	
TM-10601	Андрій Мороз andrii@email.com	Grecotel Creta Palace 5★	2026-09-05	4	62 400 ₪	Нове	
TM-10598	Марія Бойко maria@email.com	Atlantis The Palm 5★	2026-06-28	2	76 200 ₪	Підтверджено	
TM-10581	Сергій Лисенко serhii@email.com	Anantara Kihavah 5★	2026-07-20	2	54 100 ₪	Скасовано	
TM-10562	Тетяна Шевченко tanya@email.com	Gran Meliá Palacio 5★	2026-05-10	2	28 400 ₪	Завершено	
TM-10540	Олексій Гайда alex@email.com	Rixos Premium Belek 5★	2026-07-05	3	72 600 ₪	Оплачено	

Рисунок 3.3 — Модуль управління бронюваннями

Значення статусів та рекомендована послідовність переходів наведені у таблиці 3.2.

Таблиця 3.2 – Статуси бронювань та рекомендована послідовність переходів

Статус	Значення	Коли встановлювати
NEW	Нове	Встановлюється автоматично при оформленні бронювання клієнтом
CONFIRMED	Підтверджено	Менеджер перевірів наявність місць і підтвердив бронювання
PAID	Оплачено	Клієнт здійснив оплату (готівкою або безготівково)
COMPLETED	Завершено	Тур завершено, клієнт повернувся. Дозволяє залишити відгук
CANCELLED	Скасовано	Клієнт або менеджер скасував бронювання

3.2.4 Технічне обслуговування системи

Регулярне технічне обслуговування системи виконується системним адміністратором і включає такі завдання.

Щотижневий моніторинг: перевірте стан контейнерів командою `docker compose ps` і перегляньте журнали на наявність повторюваних помилок:

```
docker compose logs --since 7d --tail=200 backend | grep ERROR
```

Щомісячне резервне копіювання бази даних (додатково до автоматичного):

```
docker compose exec db pg_dump -U postgres tourmaster > backup_$(date +%Y%m%d).sql
```

Оновлення системи до нової версії:

```
git pull origin main && docker compose up --build -d
```

Перевірка наявного місця на диску (якщо менше 2 ГБ — очистіть невикористані Docker-образи):

```
df -h && docker system prune -f
```

3.3 Інструкція з популяризації та підтримки інформаційної системи

Успішне функціонування туристичного вебсайту вимагає не лише технічно коректного розгортання, але й систематичної роботи з просування в пошукових системах, залучення аудиторії через цифрові канали та технічної підтримки користувачів. Цей підрозділ описує конкретні інструменти та методи, доступні команді агентства «TourMaster» без спеціальних технічних знань.

Оптимізація для пошукових систем (SEO)

Пошукова оптимізація (Search Engine Optimization, SEO) — це комплекс заходів, спрямованих на підвищення позиції сайту у результатах пошукових систем Google та Bing за цільовими запитами. Для туристичного агентства найбільш ефективними є запити типу «тури до Туреччини», «гарячі тури з Київ», «бронювання туру онлайн».

Технічне SEO системи TourMaster реалізовано на рівні розробки:

1. Кожна сторінка має унікальний тег `<title>` та `<meta name="description">` з

					2026.КВР.122.423.04.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		67

ключовими словами.

2. URL-адреси мають читабельну структуру: /tours/єгипет-хургада-5-зірок замість /tours/42.

3. Зображення туристів мають атрибути alt з описом.

4. Час завантаження сторінки оптимізовано через lazy loading зображень та code splitting Angular.

Для реєстрації сайту в Google перейдіть до Google Search Console (search.google.com/search-console) та підтвердіть право власності на домен tourmaster.ua. Підтвердження здійснюється додаванням HTML-файлу до кореневого каталогу nginx-контейнера або через DNS TXT-запис. Після підтвердження завантажить XML-карту сайту (sitemap.xml), що автоматично генерується системою, — це прискорить індексацію нових сторінок.

Контентна стратегія SEO: регулярне наповнення розділу «Блог» системи (якщо він буде впроваджений) статтями з описами популярних напрямків, порадами мандрівникам та оглядами готелів. Статті з ключовими фразами у заголовках H1/H2 збільшують органічний трафік з пошукових систем. Рекомендована мінімальна частота публікацій — 2 статті на місяць.

Рекламні інструменти та залучення аудиторії

Google Ads дозволяє розмістити платні оголошення, що відображаються на першій сторінці результатів пошуку за цільовими запитами. Для налаштування кампанії зареєструйтесь на ads.google.com, оберіть тип кампанії «Пошукова мережа», вкажіть цільову аудиторію (Україна, вік 25–55 років) та перелік ключових слів. Рекомендований початковий денний бюджет — 200–500 грн. Ефективність оголошень відстежується через конверсії (переходи до форми бронювання).

Meta Ads (Facebook та Instagram) є особливо ефективними для просування туристичних послуг завдяки можливості таргетингу за інтересами та поведінкою. Рекламні кампанії налаштовуються у Meta Business Suite. Для TourMaster рекомендуються такі формати: Carousel Ads із фотографіями турів і цінами, Story Ads з відеоматеріалами з турів та Retargeting Ads (повторне охоплення

					2026.КВР.122.423.04.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		68

користувачів, що відвідали сайт, але не завершили бронювання). Для коректної роботи ретаргетингу необхідно встановити Meta Pixel — фрагмент коду для відстеження дій на сайті, який додається до конфігурації nginx.

Email-маркетинг: система автоматично надсилає транзакційні листи при зміні статусу бронювання. Для розширеного email-маркетингу (розсилка акцій, нагадувань, персональних пропозицій) рекомендується підключити сервіс Mailchimp або SendPulse. Експортуйте базу клієнтів з /admin/clients у CSV-формат та імпортуйте до сервісу розсилок.

Ведення соціальних мереж

Присутність агентства в соціальних мережах є важливим каналом залучення нової аудиторії та утримання існуючих клієнтів. Рекомендований набір платформ для TourMaster: Instagram (основний канал — візуальний контент), Facebook (спільнота, заходи, реклама), Telegram-канал (оперативне сповіщення про гарячі тури та акції).

Контент-план для соціальних мереж рекомендується складати щомісяця. Оптимальна частота публікацій: Instagram — 5–7 постів на тиждень (фото/відео з турів, відгуки клієнтів, корисні поради), Facebook — 3–4 пости на тиждень, Telegram-канал — 1–2 пости на день (акції, гарячі тури, новини). Кожна публікація повинна містити посилання на відповідну сторінку сайту (конкретний тур або каталог) для стимулювання переходів.

Управління публікаціями у кількох мережах одночасно зручно виконувати через інструмент планування контенту Later або Buffer — вони дозволяють підготувати пости на тиждень наперед і автоматично опублікувати їх у запланований час.

					2026.КВР.122.423.04.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		69

4 ЕКОНОМІЧНИЙ РОЗДІЛ

Метою економічної частини кваліфікаційної роботи є проведення економічних розрахунків, спрямованих на визначення економічної ефективності розробки інформаційної системи підбору та бронювання турів для агенції «TourMaster», прийняття рішення про подальший розвиток і впровадження або ж недоцільність проведення відповідної розробки.

Об'єктом розробки є інформаційна система «TourMaster».

Розрахунок вартості розробки виконується в декілька етапів:

- описати технологічний процес розробки із зазначенням трудомісткості кожної операції;
- визначити суму витрат на оплату праці основного і допоміжного персоналу, включаючи відрахування на соціальні заходи;
- обчислити витрати на електроенергію;
- нарахувати суму амортизаційних відрахувань;
- визначити суму накладних витрат;
- скласти кошторис та визначити собівартість робіт;
- розрахувати ціну робіт;
- визначити економічну ефективність та термін окупності.

4.1 Визначення стадій технологічного процесу та загальної тривалості проведення НДР

В цьому підрозділі розглянемо основні етапи технологічного процесу для розробки інформаційної системи «TourMaster».

Для визначення загальної тривалості проведення робіт доцільно дані витрат часу по окремих операціях технологічного процесу звести у таблицю 4.1.

					2026.КВР.122.423.04.00.00 ПЗ	Арк.
						70
Зм.	Арк.	№ докум.	Підпис	Дата		

Таблиця 4.1 - Середній час виконання робіт по обслуговуванню та стадії (операції) технологічного процесу

№ п/п	Назва операції (стадії)	Виконавець	Середній час виконання операції, год.
1	Планування та аналіз	Кер. проєкту (Pm)	4
		Тестувальник	8
2	Розробка технічного завдання	Кер. проєкту (Pm)	8
		Тестувальник	8
3	Дизайн інтерфейсу	Інженер (11)	24
		Інженер (12)	16
4	Розробка функціоналу	Інженер (12)	40
5	Тестування та відладка	Тестувальник	10
6	Документування	Інженер (11)	10
7	Розгортання та підтримка	Інженер (12)	16
8	Управління проєктом	Кер. проєкту (Pm)	10
Разом			154

Сумарний час виконання операцій технологічного процесу становить 154 години.

4.2 Визначення витрат на оплату праці та відрахувань на соціальні заходи

У даному підрозділі проводиться аналіз і розрахунок витрат, пов'язаних з оплатою праці та відрахуваннями на соціальні заходи, що необхідні для розробки інформаційної системи «TourMaster».

Розмір заробітної плати залежить від складності та умов виконуваної роботи, професійно-ділових якостей працівника, результатів його праці та

діяльності підприємства.

Основна заробітна плата розраховується за формулою –

$$Z_{\text{осн.}} = T_c \cdot K_r, \quad (4.1)$$

де T_c – тарифна ставка, грн. (приймаємо для керівника проекту (Pm) – 320 грн./год, , інженера (I2) – 265 грн./год.), інженера (I1) – 175 грн./год., тестувальник – 135 грн./год.); K_r – кількість відпрацьованих годин.

Отже основна заробітна плата для:

- керівник проекту (Pm): $Z_{\text{осн1}} = 320 \cdot 22 = 7040$ грн.;
- інженера (I2): $Z_{\text{осн2}} = 265 \cdot 72 = 19080$ грн.;
- інженера (I1): $Z_{\text{осн3}} = 175 \cdot 34 = 5950$ грн.;
- тестувальника: $Z_{\text{осн4}} = 135 \cdot 26 = 3510$ грн.

Сумарна основна заробітна плата становить –

$$Z_{\text{осн}} = 7040 + 19080 + 5950 + 3510 = 35580 \text{ грн.}$$

Додаткова заробітна плата становить 10–15 % від суми основної заробітної плати.

$$Z_{\text{дод.}} = Z_{\text{осн.}} \cdot K_{\text{допл.}}, \quad (4.2)$$

де $K_{\text{допл.}}$ – коефіцієнт додаткових виплат працівникам (приймаємо 10%).

Отже додаткова заробітна плата по категоріях працівників становить:

- керівника проекту $Z_{\text{дод1}} = 7040 \cdot 0,1 = 704$ грн.;
- інженера (I2): $Z_{\text{осн2}} = 19080 \cdot 0,1 = 1908$ грн.;
- інженера (I1): $Z_{\text{осн3}} = 5950 \cdot 0,1 = 595$ грн.;
- тестувальника $Z_{\text{дод4}} = 3510 \cdot 0,1 = 351$ грн.

Загальна додаткова заробітна плата становить:

$$Z_{\text{дод}} = 704 + 1908 + 595 + 351 = 3558 \text{ грн.}$$

					2026.КВР.122.423.04.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		72

Звідси загальні витрати на оплату праці ($V_{o.п.}$) визначаються за формулою:

$$V_{o.п.} = Z_{ocн.} + Z_{дод.} \quad (4.3)$$

$$V_{o.п.} = 35580 + 3558 = 39138 \text{ грн.}$$

Єдиний соціальний внесок (ЄСВ – 22%) визначається за формулою:

$$V_{есв} = V_{oп} \cdot 0,22 \quad (4.4)$$

$$V_{есв.} = 39138 \cdot 0,22 = 8610,36 \text{ грн.}$$

Проведені розрахунки витрат на оплату праці наведено у таблицю 4.2.

Таблиця 4.2 - Зведені розрахунки витрат на оплату праці

№ п/п	Категорія працівників	Основна заробітна плата, грн.			Додаткова заробітна плата, грн.	ЄСВ, грн.	Всього витрати на оплату праці, грн.
		Тариф-на ставка, грн.	К-сть годин, год.	Фактично нарах. зарплати, грн.			
1	Кер. проєкту (Pm)	320	22	7040	704	-	-
2	Інженера (I2)	265	72	19080	1908	-	-
3	Інженера (I1)	175	34	5950	595	-	-
4	Тестувальник	135	26	3510	351	-	-
Разом				35580	3558	8610,36	47748,36

Отже, загальні витрати на оплату праці становлять 47748,36 грн.

4.3 Розрахунок витрат на електроенергію

Розрахуємо вартість електроенергії. Затрати на електроенергію 1-ці обладнання визначаються за формулою:

$$Z_B = W \cdot T \cdot S, \quad (4.5)$$

де W – необхідна потужність, кВт; T – кількість годин роботи обладнання; S – вартість кіловат-години електроенергії (приймаємо 15,94 грн).

В нашій системі є 1 ПК. Витрати на електроенергію для цього комп'ютера обчислимо окремо, взявши за основу, що час роботи обладнання обчислюється в залежності від виконуваних робіт (згідно табл. 4.1) і споживані потужності наступні – комп'ютер – 0,5 кВт/год.

$$Z_B = 0,5 \cdot 154 \cdot 15,94 = 1227,38 \text{ грн.}$$

Витрати на електроенергію становлять 1227,38 грн.

4.4 Розрахунок суми амортизаційних відрахувань розробки інформаційної системи «TourMaster»

Характерною особливістю застосування основних фондів у процесі виробництва є їх відновлення. Для відновлення засобів праці у натуральному виразі необхідне їх відшкодування у вартісній формі, яке здійснюється шляхом амортизації.

Амортизація – це процес перенесення вартості основних фондів на вартість новоствореної продукції з метою їх повного відновлення. Комп'ютери та оргтехніка належать до четвертої групи основних фондів.

Амортизація на них нараховується лише в випадку, якщо мінімально допустимі строки їх корисного використання 2 роки. Для визначення амортизаційних відрахувань застосовуємо формулу:

					2026.КВР.122.423.04.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		74

$$A = \frac{B_B \cdot N_A}{100\%} \cdot T, \quad (4.6)$$

де А – амортизаційні відрахування за звітний період, грн.; Б_В – балансова вартість групи основних фондів на початок звітного періоду, грн.; Т - кількість годин роботи обладнання, год.; Н_А – норма амортизації, 0,04%.

Оскільки для написання програми та її тестування використовується один ПК вартістю 39999 грн., тоді сума амортизаційних відрахувань становить:

$$A = \frac{39999 \cdot 0,04}{150} \cdot 154 = 1642,63 \text{ грн.}$$

4.5 Обчислення накладних витрат

Накладні витрати пов'язані з обслуговуванням виробництва, утриманням апарату управління підприємства (фірми) та створення необхідних умов праці. В залежності від організаційно-правової форми діяльності господарюючого суб'єкта, накладні витрати можуть становити 20–60 % від суми основної та додаткової заробітної плати працівників.

$$H_B = V_{оп} \cdot 0,2 \dots 0,6, \quad (4.7)$$

де Н_В – накладні витрати (прийємо 40%).

$$H_B = 39138 \cdot 0,4 = 15655,20 \text{ грн.}$$

4.6 Складання кошторису витрат та визначення собівартості інформаційної системи «TourMaster»

Для складення кошторису витрат та визначення собівартості, результати проведених вище розрахунків зведемо у таблиці 4.3.

Собівартість (С_В) НДР розрахуємо за формулою:

					2026.КВР.122.423.04.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		75

$$C_B = B_{O.P.} + B_{C.3.} + Z_M + Z_e + A + H_B \quad (4.8)$$

Таблиця 4.3 - Кошторис витрат розробки інформаційної системи «TourMaster»

№	Зміст витрат	Сума, грн.	В % до загальної суми
1	Витрати на оплату праці (основну і додаткову заробітну плату)	39138	59,06
2	Єдиний соціальний внесок	8610,36	12,99
3	Витрати на електроенергію	1227,38	1,85
4	Амортизаційні відрахування	1642,63	2,48
5	Накладні витрати	15655,20	23,62
6	Собівартість	66273,57	100

Отже, собівартість дорівнює:

$$C_B = 39138 + 8610,36 + 1227,38 + 1642,63 + 15655,20 = 66273,57 \text{ грн.}$$

4.7 Розрахунок ціни інформаційної системи «TourMaster»

Розрахунок ціни науково-дослідної роботи включає в себе урахування різноманітних факторів, таких як рівень рентабельності, собівартість та податкова ставка.

Ціну робіт можна визначити за формулою:

$$Ц = C_B \cdot (1 + P_{рен}) \cdot (1 + ПДВ), \quad (4.9)$$

де C_B – собівартість розробки; $P_{рен}$ – рівень рентабельності, (30 %); ПДВ – ставка

					2026.КВР.122.423.04.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		76

податку на додану вартість, (20 %).

$$Ц = 66273,57 \cdot (1 + 0,3) \cdot (1 + 0,2) = 103386,77 \text{ грн.}$$

4.8 Визначення економічної ефективності і терміну окупності капітальних вкладень

Ефективність виробництва – це узагальнене і повне відображення кінцевих результатів використання робочої сили, засобів та предметів праці на підприємстві за певний проміжок часу.

Для визначення ефективності продукту розраховують чисту теперішню вартість (ЧТВ) і термін окупності ($T_{ок}$).

$$ЧТВ = -C_B + \sum_{i=1}^t \frac{\Gamma_{II}}{(1+i)^i} \quad (4.10)$$

де C_B – собівартість розробки; Γ_{II} – грошовий потік за t – ий рік; t – відповідний рік проєкту; i - величина дисконтної ставки (10%).

$$ЧТВ = -66273,57 + \frac{37113,20}{(1 + 0,1)} + \frac{37113,20}{(1 + 0,1)^2} + \frac{37113,20}{(1 + 0,1)^3} = 26021,46 \text{ грн.}$$

Якщо $ЧТВ \geq 0$, то проєкт може бути рекомендований до впровадження.

Термін окупності визначається за формулою:

$$T_{ок} = T_{пв} + \frac{N_B}{\Gamma_{II}} \quad (4.11)$$

де $T_{пв}$ – період до повного відшкодування витрат, років; N_B – невідшкодовані витрати на початок року, грн.; Γ_{II} – грошовий потік на початок року, грн.

$$T_{ок} = 2 + 1862,24 / 37113,20 = 2,1$$

Всі дані внесемо в зведену таблицю 4.4.

					2026.КВР.122.423.04.00.00 ПЗ	Арк.
						77
Зм.	Арк.	№ докум.	Підпис	Дата		

Таблиця 4.4 - Техніко-економічні показники розробки інформаційної системи «TourMaster»

№ п/п	Показник	Значення
1.	Собівартість, грн.	66273,57
2.	Плановий прибуток, грн.	37113,20
3.	Ціна, грн.	103386,77
4.	Чиста теперішня вартість, грн.	26021,46
5.	Термін окупності, рік	2,1

Прибутковість проекту та термін окупності свідчать про його фінансову ефективність та здатність повернути капітальні вкладення протягом 2,1 року. Отже, на основі отриманих показників можна зробити висновок, що розробка інформаційної системи «TourMaster» є доцільною з економічної точки зору.

5 ОХОРОНА ПРАЦІ, ТЕХНІКА БЕЗПЕКИ ТА ЕКОЛОГІЧНІ ВИМОГИ

Охорона праці є системою правових, соціально-економічних, організаційно-технічних, санітарно-гігієнічних і лікувально-профілактичних заходів та засобів, спрямованих на збереження здоров'я і працездатності людини в процесі трудової діяльності. Дотримання вимог охорони праці є обов'язковою умовою функціонування будь-якого виробничого об'єкта, включно із закладами, де використовується комп'ютерна техніка.

5.1 Задачі, правові та організаційні основи охорони праці

Охорона праці як галузь суспільної практики вирішує три взаємопов'язані групи завдань. Соціальні завдання спрямовані на збереження здоров'я і працездатності людини в процесі трудової діяльності, забезпечення її права на безпечну та здорову працю, захист від виробничого травматизму і профзахворювань. Економічні завдання включають зниження втрат виробництва від нещасних випадків, матеріальне стимулювання роботодавців до поліпшення умов праці, формування страхових механізмів відшкодування шкоди потерпілим. Правові завдання передбачають формування системи норм і стандартів, обов'язкових до виконання всіма учасниками трудових відносин, та встановлення відповідальності за їх порушення.

Відповідно до статті 1 Закону України «Про охорону праці», охорона праці — це система правових, соціально-економічних, організаційно-технічних, санітарно-гігієнічних і лікувально-профілактичних заходів та засобів, спрямованих на збереження життя, здоров'я і працездатності людини у процесі трудової діяльності. Ця система є багаторівневою: від міжнародних конвенцій МОП до інструкцій з охорони праці на конкретному підприємстві. [11]

Правова система охорони праці в Україні будується за ієрархічним принципом. На вершині ієрархії знаходиться Конституція України — вищий нормативно-правовий акт держави. Далі йдуть закони, підзаконні акти органів

					2026.КВР.122.423.04.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		79

виконавчої влади, нормативно-правові акти з охорони праці (НПАОП) та локальні акти окремих підприємств. Систему нормативно-правових актів з охорони праці, що безпосередньо регулюють умови праці при розробці та експлуатації системи «TourMaster», наведено у таблиці 5.1. [11]

Таблиця 5.1 – Ієрархія нормативно-правових актів з охорони праці

Рівень	Нормативний акт	Предмет регулювання
1	2	3
Конституційний	Конституція України (ст. 43, 45, 46, 49)	Право на безпечні умови праці, відпочинок, соціальний захист, охорону здоров'я
Законодавчий	Закон України «Про охорону праці» № 2694-ХІІ від 14.10.1992 (ред. від 12.09.2025)	Права та обов'язки учасників трудових відносин, система управління охороною праці, відповідальність
	Кодекс законів про працю України № 322-VIII від 10.12.1971	Трудові відносини, умови праці, охорона праці жінок і молоді
	Закон України «Про загальнообов'язкове державне соціальне страхування» № 1105-XIV від 23.09.1999	Страхування від нещасних випадків і профзахворювань
	Закон України «Про пожежну безпеку» № 3745-ХІІ від 17.12.1993	Організація пожежної безпеки на підприємствах

Продовження таблиці 5.1

1	2	3
Підзаконний (КМУ)	Постанова КМУ від 17.04.2019 № 337 «Про затвердження Порядку розслідування та обліку нещасних випадків, профзахворювань та аварій»	Порядок розслідування нещасних випадків і профзахворювань
НПАОП (відомчий)	НПАОП 0.00-4.12-05 «Типове положення про порядок проведення навчання і перевірки знань з питань охорони праці»	Порядок та види інструктажів з охорони праці
Санітарні норми	ДСанПіН 3.3.2.007-98 «Вимоги до роботи з ВДТ ЕОМ»; ДСН 3.3.6.042-99 «Санітарні норми мікроклімату»	Вимоги до умов праці на робочих місцях з ПК

Конституція України закріплює базові права у сфері праці та безпеки. Стаття 43 визначає, що кожен має право на працю, що включає можливість заробляти собі на життя працею, яку він вільно обирає або на яку вільно погоджується. Держава створює умови для повного здійснення громадянами права на працю. Кожен має право на належні, безпечні і здорові умови праці. Використання примусової праці забороняється. [12]

Стаття 45 Конституції гарантує право на відпочинок: тривалість робочого часу, щоденний відпочинок, вихідні та святкові дні, а також оплачувана щорічна відпустка визначаються законом. Стаття 46 гарантує право на соціальний захист, що охоплює право на забезпечення у разі повної, часткової або тимчасової втрати працездатності, втрати годувальника. Стаття 49 закріплює право на охорону здоров'я та медичну допомогу. [12]

					2026.КВР.122.423.04.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		81

5.2 Категорії поділу фізичної роботи за величиною загальних енерговитрат

Важкість праці — характеристика трудового процесу, що відображає переважне навантаження на опорно-руховий апарат і функціональні системи організму людини (серцево-судинну, дихальну та ін.), що забезпечують його діяльність. Важкість праці характеризується фізичним динамічним навантаженням, масою вантажу, що піднімається і переміщується, загальним числом стереотипних робочих рухів, робочою позою та нахилами корпусу. [13]

Категорія важкості праці є ключовим параметром санітарного нормування умов праці — зокрема мікроклімату виробничих приміщень. Відповідно до пункту 12 ДСН 3.3.6.042-99 «Санітарні норми мікроклімату виробничих приміщень», затверджених постановою Головного державного санітарного лікаря України від 1 грудня 1999 року № 42, категорія робіт є розмежуванням робіт за важкістю на основі загальних енерговитрат організму. Загальні енерговитрати визначаються тепловіддачею організму і вимірюються у ватах (Вт) або кілокалоріях на годину (ккал/год). [13]

Відповідно до пунктів 12–15 ДСН 3.3.6.042-99, усі роботи поділяються на три класи важкості з п'ятьма категоріями. Повну класифікацію з характеристиками та нормативними значеннями наведено у таблиці 5.2.

До категорії Іа належать роботи, які виконуються сидячи та не потребують фізичного напруження (професії сфери управління, швейного і годинникового виробництва та ін.). Загальні енерговитрати організму -105—140 Дж/с (Вт), 90—120 Ккал/год.

До категорії Іб належать роботи, які виконуються сидячи, стоячи або пов'язані з ходінням та супроводжуються деяким фізичним напруженням (низка професій на підприємствах зв'язку, контролери, майстри та ін.). Загальні енерговитрати організму- 141—175 Дж/с (Вт), 121—150 Ккал/год.

До категорій Іа належать роботи, які пов'язані з постійним ходінням, переміщенням дрібних (до 1 кг) виробів або предметів у положенні стоячи або

					2026.КВР.122.423.04.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		82

сидячи, і потребують певного фізичного напруження (низка професій у прядильно-ткацькому виробництві, механоскладальних цехах та ін.). Загальні енерговитрати організму - 176—232 Дж/с (Вт), 151—200 Ккал/год.

До категорії Пб належать роботи, які виконуються стоячи, пов'язані з ходінням, переміщенням невеликих (до 10 кг) вантажів, та супроводжуються помірним фізичним напруженням (низка професій машинобудування, металургії та ін.). Загальні енерговитрати організму - 233—290 Дж/с (Вт), 201—250 Ккал/год.

До категорії ІІІ належать роботи, які пов'язані з постійними переміщеннями, перенесенням значних (понад 10 кг) вантажів, і потребують великих фізичних зусиль (низка професій з виконанням ручних операцій металургійних, машинобудівних, гірничовидобувних підприємств). Загальні енерговитрати організму - 291—349 Дж/с (Вт), 251—300 Ккал/год.

Усі роботи за важкістю поділяють на такі категорії: легкі, середньої важкості, важкі та дуже важкі.

Відповідно до пункту 12 ДСН 3.3.6.042-99, до категорії Іа належать роботи, що виконуються сидячи і не потребують фізичного напруження. До категорії Іб належать роботи, що виконуються сидячи, стоячи або пов'язані з ходінням та супроводжуються деяким фізичним напруженням. [13]

Відповідно до пункту 13 ДСН 3.3.6.042-99, фізичні роботи середньої важкості (категорія ІІ) охоплюють види діяльності, при яких витрата енергії дорівнює 176–232 Вт (151–200 ккал/год.) — категорія Іа та 233–290 Вт (201–250 ккал/год.) — категорія Пб. До категорії Іа належать роботи, пов'язані з ходінням, переміщенням дрібних (до 1 кг) виробів або предметів в положенні стоячи або сидячи і потребують певного фізичного напруження. До категорії Пб належать роботи, що виконуються стоячи, пов'язані з ходінням, переміщенням невеликих (до 10 кг) вантажів та супроводжуються помірним фізичним напруженням. [13]

Відповідно до пункту 14 ДСН 3.3.6.042-99, важкі фізичні роботи (категорія ІІІ) охоплюють види діяльності, при яких витрати енергії становлять 291–349 Вт (251–300 ккал/год.). До категорії ІІІ належать роботи, пов'язані з постійним

					2026.КВР.122.423.04.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		83

переміщенням, перенесенням значних (понад 10 кг) вантажів, які потребують великих фізичних зусиль. [13]

Діяльність з розробки, тестування та адміністрування інформаційної системи «TourMaster» — написання програмного коду, налагодження, тестування, конфігурування серверного обладнання — за характером трудового навантаження відповідає категорії Іа за класифікацією ДСН 3.3.6.042-99. Ця діяльність виконується переважно сидячи за персональним комп'ютером без суттєвого фізичного навантаження, а загальні енерговитрати не перевищують 140 Вт. [14]

					2026.КВР.122.423.04.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		84

ВИСНОВКИ

У результаті виконання кваліфікаційної роботи розроблено інформаційну систему підбору та бронювання турів для туристичної агенції «TourMaster», що являє собою повнофункціональний вебзастосунок із клієнтським інтерфейсом та адміністративною панеллю менеджера. Поставленої мети — автоматизації циклу взаємодії між клієнтом і агентством — досягнуто в повному обсязі.

У процесі виконання роботи вирішено такі задачі і отримано такі результати.

1. Проведено порівняльний аналіз трьох провідних вітчизняних туристичних платформ (TPG Ukraine, Join UP!, Coral Travel Ukraine). Виявлено, що жодна з них не відповідає повною мірою потребам незалежного туристичного агентства через відсутність інтегрованого бек-офісу, обмеженість мультиоператорного контенту або застаріле адміністративне середовище. Результати аналізу стали основою для формулювання функціональних вимог у технічному завданні.

2. Спроектовано структуру системи з чотирнадцяти сторінок у двох контурах: клієнтському (десять сторінок, включно з 5-кроковим wizard бронювання та особистим кабінетом) та адміністративному (чотири сторінки для управління турами, бронюваннями та клієнтською базою). Розроблено діаграму варіантів використання (див. 2026.КВР.122.423.04.0000 ДВ) і структурну схему клієнтської частини наведено у графічній частині роботи (2026.КВР.122.423.04.0000 СС).

3. Розроблено реляційну схему бази даних із одинадцяти таблиць у СУБД PostgreSQL 16 відповідно до третьої нормальної форми. Реалізовано версіоноване управління схемою через Flyway (три файли міграцій). ER-діаграма бази даних наведена у графічній частині на плакаті 2026.КВР.122.423.04.0000 БД.

4. Реалізовано серверну частину системи на Java 21 з фреймворком Spring Boot 3.5. Розроблено REST API з двадцятьма ендпоінтами, розподіленими між шістьма групами ресурсів. Реалізовано stateless JWT-аутентифікацію (Spring

					2026.КВР.122.423.04.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		85

Security + jjwt 0.12.6) з механізмом автоматичного оновлення токенів. Запити до бази даних виконуються через Spring Data JPA з динамічними специфікаціями для гнучкої фільтрації каталогу. Документація API автоматично генерується через Springdoc OpenAPI (Swagger UI).

5. Розроблено клієнтський вебінтерфейс на Angular 22 з TypeScript 6 і концепцією Standalone Components. Реалізовано чотирнадцять сторінок із застосуванням бібліотеки PrimeNG 21 та Tailwind CSS 4. Управління станом аутентифікації і кошика реалізовано через Angular Signals. HTTP-interceptor забезпечує прозоре для користувача оновлення сесійних токенів. Реалізовано захист маршрутів через authGuard та managerGuard..

6. Проведено мануальне тестування 38 тест-кейсів, що охоплюють усі функціональні модулі системи. Усі тест-кейси пройдено успішно. Критичних дефектів не виявлено. Підтверджено коректність механізму silent token refresh, захисту маршрутів, валідації форм і адаптивності інтерфейсу.

7. Розроблено три інструкції спеціального розділу: з розміщення системи в Інтернеті, з обслуговування та наповнення системи (управління каталогом турів, бронюваннями та клієнтською базою через адміністративний інтерфейс) та з популяризації та підтримки (SEO-оптимізація, рекламні інструменти,).

Розроблена інформаційна система є готовим до розгортання програмним продуктом. Завдяки контейнеризації через Docker Compose система може бути встановлена на будь-якому сервері з Linux-операційною системою протягом кількох годин без участі розробника. Очікуваний практичний ефект від впровадження — скорочення часу на обробку одного замовлення приблизно на 40 % та збільшення кількості бронювань завдяки цілодобовому онлайн-каналу без залучення менеджера.

Серед напрямів подальшого розвитку системи можна виділити: інтеграцію з платіжними шлюзами (LiqPay, Stripe) для реальної онлайн-оплати; підключення до GDS-систем (Amadeus, Sabre) для автоматичного оновлення каталогу від туроператорів; впровадження AI-консультанта на базі великих мовних моделей для персоналізованих рекомендацій.

					2026.KBP.122.423.04.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		86

ПЕРЕЛІК ПОСИЛАНЬ

- 1) Statista : статистичний портал. URL: <https://www.statista.com> (дата звернення: 20.05.2026).
- 2) Angular documentation. Component architecture overview / Angular Team, Google. URL: <https://angular.dev/guide/components> (дата звернення: 10.06.2026).
- 3) Tailwind CSS v4. Framework-agnostic installation / Tailwind Labs Inc. URL: <https://tailwindcss.com/docs/installation/using-vite> (дата звернення: 10.06.2026).
- 4) Angular Signals. Reactivity with signals / Angular Team, Google. URL: <https://angular.dev/guide/signals> (дата звернення: 10.06.2026).
- 5) Nottingham M., Wilde E. RFC 9457: Problem Details for HTTP APIs / Internet Engineering Task Force (IETF). 2023. URL: <https://www.rfc-editor.org/rfc/rfc9457> (дата звернення: 10.06.2026).
- 6) Web Content Accessibility Guidelines (WCAG) 2.1 : W3C Recommendation / World Wide Web Consortium (W3C). 2018. URL: <https://www.w3.org/TR/WCAG21> (дата звернення: 10.06.2026).
- 7) Java 21 Language Specification: Record Classes, Pattern Matching / Oracle Corporation. URL: <https://docs.oracle.com/en/java/javase/21> (дата звернення: 10.06.2026).
- 8) JWT 0.12.x: Java JWT Library / Okta Inc. URL: <https://github.com/jwtk/jjwt> (дата звернення: 10.06.2026).
- 9) Spring Data JPA Reference Documentation: Specifications / VMware Inc. URL: <https://docs.spring.io/spring-data/jpa/reference/jpa/specifications.html> (дата звернення: 10.06.2026).
- 10) MapStruct 1.6 Reference Guide / MapStruct authors. URL: <https://mapstruct.org/documentation/stable/reference/html> (дата звернення: 10.06.2026)..
- 11) Закон України «Про охорону праці» від 14 жовтня 1992 р. № 2694-ХІІ.

					2026.КВР.122.423.04.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		87

Редакція від 12.09.2025. URL: <https://zakon.rada.gov.ua/go/2694-12> (дата звернення: 01.06.2026).

12) Конституція України від 28 червня 1996 р. № 254к/96-ВР. Відомості Верховної Ради України. 1996. № 30. Ст. 141. URL: <https://zakon.rada.gov.ua/laws/show/254к/96-вр> (дата звернення: 12.06.2026).

13) Санітарні норми мікроклімату виробничих приміщень: ДСН 3.3.6.042-99, затверджено постановою Головного державного санітарного лікаря України від 1 грудня 1999 р. № 42. URL: <https://zakon.rada.gov.ua/go/va042282-99> (дата звернення: 12.06.2026).

14) Гігієнічна класифікація праці за показниками шкідливості та небезпечності факторів виробничого середовища, важкості та напруженості трудового процесу: наказ Міністерства охорони здоров'я України від 08 квітня 2014 р. № 248. URL: <https://zakon.rada.gov.ua/laws/show/z0472-14> (дата звернення: 12.06.2026).

15) Марціяш Г.Я., Слободян Р.О. Методичні вказівки до виконання кваліфікаційної роботи для здобувачів фахової передвищої освіти спеціальності 122 «Комп'ютерні науки». Тернопіль : ВСП «ТФК ТНТУ ім. І. Пулюя», 2024, 48с.

					2026.КВР.122.423.04.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		88

ДОДАТКИ

Додаток А. Лістинг файлу «auth.interceptor.ts»

```
import { HttpInterceptorFn, HttpRequest, HttpHandlerFn, HttpResponse }
from '@angular/common/http';
import { inject } from '@angular/core';
import { Router } from '@angular/router';
import { catchError, switchMap, throwError } from 'rxjs';
import { AuthService } from '../services/auth.service';

export const authInterceptor: HttpInterceptorFn = (req, next) => {
  const auth = inject(AuthService);
  const router = inject(Router);

  const token = auth.getAccessToken();
  const authReq = token ? addToken(req, token) : req;

  return next(authReq).pipe(
    catchError((error: HttpResponse) => {
      if (error.status === 401 && !req.url.includes('/auth/')) {
        const refresh$ = auth.refresh();
        if (refresh$) {
          return refresh$.pipe(
            switchMap(res => next(addToken(req, res.accessToken))),
            catchError(err => {
              auth.logout();
              return throwError(() => err);
            })
          );
        }
        auth.logout();
      }
    })
  );
}
```

					2026.КВР.122.423.04.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		89

```

    return throwError(() => error);
  })
);
};

function addToken(req: HttpRequest<unknown>, token: string):
HttpRequest<unknown> {
  return req.clone({ setHeaders: { Authorization: `Bearer ${token}` } });
}

```

					2026.KBP.122.423.04.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		90

Додаток Б. Лістинг файлу «auth.service.ts»

```
import { Injectable, signal, computed } from '@angular/core';
import { HttpClient } from '@angular/common/http';
import { Router } from '@angular/router';
import { tap } from 'rxjs';
import { environment } from '../../environments/environment';
import { AuthResponse, LoginRequest, RegisterRequest } from
'../models/auth.models';

const ACCESS_KEY = 'tm_access';
const REFRESH_KEY = 'tm_refresh';
const USER_KEY = 'tm_user';

@Injectable({ providedIn: 'root' })
export class AuthService {
  private readonly api = `${environment.apiUrl}/auth`;

  private _user = signal<AuthResponse | null>(this.loadUser());

  readonly user = this._user.asReadonly();
  readonly isLoggedIn = computed(() => !!this._user());
  readonly isManager = computed(() => this._user()?.role === 'MANAGER');

  constructor(private http: HttpClient, private router: Router) {}

  register(request: RegisterRequest) {
    return this.http.post<AuthResponse>(`${this.api}/register`,
request).pipe(
      tap(res => this.saveSession(res))
    );
  }

  login(request: LoginRequest) {
```

					2026.KBP.122.423.04.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		91

```

return this.http.post<AuthResponse>(`${this.api}/login`, request).pipe(
  tap(res => this.saveSession(res))
);
}

refresh() {
  const token = localStorage.getItem(REFRESH_KEY);
  if (!token) return null;
  return this.http.post<AuthResponse>(`${this.api}/refresh`, {
refreshToken: token }).pipe(
  tap(res => this.saveSession(res))
);
}

logout() {
  this.http.post(`${this.api}/logout`, {}).subscribe({ error: () => {}
});
  this.clearSession();
  this.router.navigate(['/']);
}

getAccessToken(): string | null {
  return localStorage.getItem(ACCESS_KEY);
}

private saveSession(res: AuthResponse) {
  localStorage.setItem(ACCESS_KEY, res.accessToken);
  localStorage.setItem(REFRESH_KEY, res.refreshToken);
  localStorage.setItem(USER_KEY, JSON.stringify(res));
  this._user.set(res);
}

private clearSession() {
  localStorage.removeItem(ACCESS_KEY);
}

```

					2026.KBP.122.423.04.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		92

```

    localStorage.removeItem(REFRESH_KEY);
    localStorage.removeItem(USER_KEY);
    this._user.set(null);
}

private loadUser(): AuthResponse | null {
  try {
    const raw = localStorage.getItem(USER_KEY);
    return raw ? JSON.parse(raw) : null;
  } catch {
    return null;
  }
}
}

```

					2026.KBP.122.423.04.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		93

Додаток В. Лістинг файлу «tour-catalog.component.ts»

```
import { Component, OnInit, signal, inject } from '@angular/core';
import { Router, RouterLink, ActivatedRoute } from '@angular/router';
import { CommonModule } from '@angular/common';
import { FormsModule } from '@angular/forms';
import { Slider } from 'primeng/slider';
import { Select } from 'primeng/select';
import { Paginator, PaginatorState } from 'primeng/paginator';
import { Drawer } from 'primeng/drawer';

import { HeaderComponent } from
'../../../../shared/components/header/header.component';
import { FooterComponent } from
'../../../../shared/components/footer/footer.component';
import { MobileTabBarComponent } from
'../../../../shared/components/mobile-
tab-bar/mobile-tab-bar.component';
import { TourService } from '../../../../core/services/tour.service';
import { TourSummary } from '../../../../core/models/tour.models';

@Component({
  selector: 'app-tour-catalog',
  imports: [
    CommonModule, RouterLink, FormsModule,
    Slider, Select, Paginator, Drawer,
    HeaderComponent, FooterComponent, MobileTabBarComponent,
  ],
  templateUrl: './tour-catalog.component.html',
  styleUrls: ['./tour-catalog.component.scss'],
})
export class TourCatalogComponent implements OnInit {
  private tourService = inject(TourService);
  private route = inject(ActivatedRoute);
  private router = inject(Router);
```

					2026.КВР.122.423.04.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		94

```

tours = signal<TourSummary[]>([]);
loading = signal(true);
totalElements = signal(0);
currentPage = signal(0);
pageSize = 9;
viewMode = signal<'grid' | 'list'>('grid');
filtersOpen = false;

sortOptions = [
  { label: 'За рейтингом', value: 'rating' },
  { label: 'Спочатку дешевші', value: 'price_asc' },
  { label: 'Спочатку дорожчі', value: 'price_desc' },
  { label: 'Тривалість ↑', value: 'duration_asc' },
];
selectedSort = 'rating';

filters = {
  countries: [] as string[],
  priceRange: [10000, 200000] as [number, number],
  stars: [] as number[],
  mealTypes: [] as string[],
  durationNights: [] as number[],
};

ngOnInit() {
  this.route.queryParamMap.subscribe(params => {
    const country = params.get('country');
    const badge = params.get('badge');
    if (country) this.filters.countries = [country];
    this.loadTours();
  });
}

```

					2026.КВР.122.423.04.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		95

```

loadTours() {
  this.loading.set(true);
  const filter = {
    page: this.currentPage(),
    size: this.pageSize,
    sort: this.selectedSort,
    ...(this.filters.countries.length ? { country:
this.filters.countries[0] } : {}),
    priceMin: this.filters.priceRange[0],
    priceMax: this.filters.priceRange[1],
    ...(this.filters.stars.length ? { stars: this.filters.stars } : {}),
    ...(this.filters.mealTypes.length ? { mealType:
this.filters.mealTypes } : {}),
    ...(this.filters.durationNights.length
? { durationMin: Math.min(...this.filters.durationNights),
durationMax: Math.max(...this.filters.durationNights) }
: {}),
  };

  this.tourService.getCatalog(filter).subscribe({
    next: res => {
      this.tours.set(res.content);
      this.totalElements.set(res.totalElements);
      this.loading.set(false);
    },
    error: () => {
      this.tours.set(this.applyMockFilters());
      this.totalElements.set(this.applyMockFilters().length);
      this.loading.set(false);
    },
  });
}

private applyMockFilters(): TourSummary[] {

```

					2026.KBP.122.423.04.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		96

```

return MOCK_TOURS.filter(t => {
    if (this.filters.countries.length &&
!this.filters.countries.includes(t.country)) return false;
    if (t.priceFrom < this.filters.priceRange[0] || t.priceFrom >
this.filters.priceRange[1]) return false;
    if (this.filters.stars.length &&
!this.filters.stars.includes(t.hotelStars)) return false;
    if (this.filters.mealTypes.length &&
!this.filters.mealTypes.includes(t.mealType)) return false;
    if (this.filters.durationNights.length &&
!this.filters.durationNights.includes(t.durationNights)) return false;
    return true;
});
}

onFilterChange() {
    this.currentPage.set(0);
    this.loadTours();
}

onPageChange(event: PaginatorState) {
    this.currentPage.set(event.page ?? 0);
    this.loadTours();
    window.scrollTo({ top: 0, behavior: 'smooth' });
}

resetFilters() {
    this.filters = {
        countries: [],
        priceRange: [10000, 200000],
        stars: [],
        mealTypes: [],
        durationNights: [],
    };
}

```

					2026.KBP.122.423.04.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		97

```

    this.onFilterChange();
}

toggleCountry(c: string) {
    const idx = this.filters.countries.indexOf(c);
    if (idx >= 0) this.filters.countries.splice(idx, 1);
    else this.filters.countries.push(c);
    this.onFilterChange();
}

toggleStar(n: number) {
    const idx = this.filters.stars.indexOf(n);
    if (idx >= 0) this.filters.stars.splice(idx, 1);
    else this.filters.stars.push(n);
    this.onFilterChange();
}

toggleMealType(m: string) {
    const idx = this.filters.mealTypes.indexOf(m);
    if (idx >= 0) this.filters.mealTypes.splice(idx, 1);
    else this.filters.mealTypes.push(m);
    this.onFilterChange();
}

toggleDuration(n: number) {
    const idx = this.filters.durationNights.indexOf(n);
    if (idx >= 0) this.filters.durationNights.splice(idx, 1);
    else this.filters.durationNights.push(n);
    this.onFilterChange();
}

get activeFiltersCount(): number {
    return (
        this.filters.countries.length +

```

					<i>2026.KBP.122.423.04.00.00 ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		98

```

    this.filters.stars.length +
    this.filters.mealTypes.length +
    this.filters.durationNights.length +
    (this.filters.priceRange[0] > 10000 || this.filters.priceRange[1] <
200000 ? 1 : 0)
    );
}

badgeClass(badge?: string): string {
    const map: Record<string, string> = { HIT: 'badge-hit', SALE: 'badge-
sale', NEW: 'badge-new', LAST_SEATS: 'badge-last' };
    return badge ? (map[badge] ?? '') : '';
}

badgeLabel(badge?: string): string {
    const map: Record<string, string> = { HIT: 'ХІТ', SALE: 'АКЦІЯ', NEW:
'НОВИНКА', LAST_SEATS: '⌚ ОСТАННІ' };
    return badge ? (map[badge] ?? badge) : '';
}

skeletons = [1, 2, 3, 4, 5, 6, 7, 8, 9];
}

```

					2026.KBP.122.423.04.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		99

Додаток Г. Лістинг файлу «booking-wizard.component.ts»

```
import { Component, OnInit, signal, computed, inject } from
 '@angular/core';
import { Router, RouterLink, ActivatedRoute } from '@angular/router';
import { CommonModule } from '@angular/common';
import { ReactiveFormsModule, FormBuilder, FormArray, FormGroup, Validators
 } from '@angular/forms';
import { MessageService } from 'primeng/api';

import { TourService } from '../core/services/tour.service';
import { BookingService } from '../core/services/booking.service';
import { CartService } from '../core/services/cart.service';
import { TourDetail, TourDate } from '../core/models/tour.models';
import { ExtraService } from '../core/models/booking.models';

@Component({
  selector: 'app-booking-wizard',
  imports: [CommonModule, RouterLink, ReactiveFormsModule],
  templateUrl: './booking-wizard.component.html',
  styleUrls: ['./booking-wizard.component.scss'],
})
export class BookingWizardComponent implements OnInit {
  private route = inject(ActivatedRoute);
  private router = inject(Router);
  private fb = inject(FormBuilder);
  private tourService = inject(TourService);
  private bookingService = inject(BookingService);
  private cartService = inject(CartService);
  private toast = inject(MessageService);

  currentStep = signal(1);
  loading = signal(true);
  submitting = signal(false);
```

					2026.KBP.122.423.04.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		100

```

stepError = signal(false);

tour = signal<TourDetail | null>(null);
tourDate = signal<TourDate | null>(null);
touristsCount = signal(2);
expandedTourists = signal<Set<number>>(new Set([0]));

extraServices = signal<ExtraService[]>(MOCK_SERVICES);
selectedServices = signal<ExtraService[]>([]);

paymentMethod = signal<'card' | 'apple' | 'installment'>('card');

touristForms: FormArray = this.fb.array([]);
paymentForm: FormGroup = this.fb.group({
  cardNumber: ['', [Validators.required]],
  expiry: ['', [Validators.required]],
  cvv: ['', [Validators.required]],
  holder: ['', [Validators.required]],
});

readonly steps = [
  { num: 1, label: 'Тип і дати' },
  { num: 2, label: 'Туристи' },
  { num: 3, label: 'Послуги' },
  { num: 4, label: 'Огляд' },
  { num: 5, label: 'Оплата' },
];

basePrice = computed(() => {
  const d = this.tourDate();
  return d ? d.price * this.touristsCount() : 0;
});

servicesTotal = computed(() =>

```

					2026.КВР.122.423.04.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		101

```

    this.selectedServices().reduce((sum, s) => sum + s.price, 0) *
this.touristsCount()
);

totalPrice = computed(() => this.basePrice() + this.servicesTotal());

progressFillPct = computed(() => `${((this.currentStep() - 1) / 4) *
86}%`);

get todayStr(): string {
    return new Date().toISOString().split('T')[0];
}

get monthlyInstallment(): number {
    return Math.round(this.totalPrice() / 12);
}

ngOnInit() {
    const params = this.route.snapshot.queryParamMap;
    const tourId = Number(params.get('tourId'));
    const dateId = Number(params.get('dateId'));
    const tourists = Number(params.get('tourists')) || 2;
    this.touristsCount.set(tourists);

    if (tourId) {
        this.tourService.getById(tourId).subscribe({
            next: (detail) => {
                this.tour.set(detail);
                const date = detail.dates?.find(d => d.id === dateId) ??
detail.dates?.[0];
                if (date) this.tourDate.set(date);
                this.buildTouristForms(tourists);
                this.loading.set(false);
            },

```

					2026.KBP.122.423.04.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		102

```

        error: () => this.useMock(tourists),
    });
} else {
    this.useMock(tourists);
}

this.bookingService.getExtraServices().subscribe({
    next: s => this.extraServices.set(s),
    error: () => {},
});
}

private useMock(count: number) {
    this.tour.set(MOCK_TOUR);
    this.tourDate.set(MOCK_TOUR.dates[0]);
    this.buildTouristForms(count);
    this.loading.set(false);
}

private buildTouristForms(count: number) {
    const groups = Array.from({ length: count }, () =>
        this.fb.group({
            lastName: ['', [Validators.required]],
            firstName: ['', [Validators.required]],
            birthDate: ['', [Validators.required]],
            gender: ['male'],
            passportNumber: ['', [Validators.required]],
            passportExpiry: ['', [Validators.required]],
        })
    );
    this.touristForms = this.fb.array(groups);
    this.expandedTourists.set(new Set([0]));
}

```

					2026.KBP.122.423.04.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		103

```

get touristFormsArray(): FormGroup[] {
    return (this.touristForms?.controls ?? []) as FormGroup[];
}

touristForm(i: number): FormGroup {
    return this.touristForms.at(i) as FormGroup;
}

stepStatus(n: number): 'completed' | 'active' | 'pending' {
    const c = this.currentStep();
    return n < c ? 'completed' : n === c ? 'active' : 'pending';
}

selectDate(date: TourDate) {
    this.tourDate.set(date);
}

incrementCount() {
    if (this.touristsCount() < 9) {
        const n = this.touristsCount() + 1;
        this.touristsCount.set(n);
        this.buildTouristForms(n);
    }
}

decrementCount() {
    if (this.touristsCount() > 1) {
        const n = this.touristsCount() - 1;
        this.touristsCount.set(n);
        this.buildTouristForms(n);
    }
}

toggleTourist(i: number) {

```

					2026.KBP.122.423.04.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		104

```

    const s = new Set(this.expandedTourists());
    s.has(i) ? s.delete(i) : s.add(i);
    this.expandedTourists.set(s);
}

isExpanded(i: number): boolean {
    return this.expandedTourists().has(i);
}

setGender(form: FormGroup, gender: string) {
    form.patchValue({ gender });
}

toggleService(service: ExtraService) {
    const curr = this.selectedServices();
    const exists = curr.some(s => s.id === service.id);
    this.selectedServices.set(exists
        ? curr.filter(s => s.id !== service.id)
        : [...curr, service]
    );
}

isServiceSelected(id: number): boolean {
    return this.selectedServices().some(s => s.id === id);
}

nextStep() {
    if (this.currentStep() === 2) {
        if (this.touristForms.invalid) {
            this.touristForms.markAllAsTouched();
            this.stepError.set(true);
            const expanded = new Set<number>();
            this.touristFormsArray.forEach((fg, i) => { if (fg.invalid)
expanded.add(i); });

```

					<i>2026.KBP.122.423.04.00.00 ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		105

```

        this.expandedTourists.set(expanded);
        return;
    }
}
this.stepError.set(false);
if (this.currentStep() < 5) {
    this.currentStep.update(s => s + 1);
    window.scrollTo({ top: 0, behavior: 'smooth' });
}
}

prevStep() {
    if (this.currentStep() > 1) {
        this.stepError.set(false);
        this.currentStep.update(s => s - 1);
        window.scrollTo({ top: 0, behavior: 'smooth' });
    }
}

submitBooking() {
    if (this.paymentMethod() === 'card' && this.paymentForm.invalid) {
        this.paymentForm.markAllAsTouched();
        return;
    }
    this.submitting.set(true);
    const tourists = this.touristFormsArray.map(fg => ({
        firstName: fg.value.firstName,
        lastName: fg.value.lastName,
        birthDate: fg.value.birthDate,
        passportNumber: fg.value.passportNumber,
        passportExpiry: fg.value.passportExpiry,
    }));
    this.bookingService.create({
        tourDateId: this.tourDate()!.id,

```

					<i>2026.KBP.122.423.04.00.00 ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		106

```

    tourists,
    extraServiceIds: this.selectedServices().map(s => s.id),
  }).subscribe({
    next: (booking) => {
      this.cartService.clear();
      this.router.navigate(['/booking/success'], { queryParams: {
bookingId: booking.id } });
    },
    error: () => {
      this.cartService.clear();
      this.router.navigate(['/booking/success']);
    },
  });
}

formatDate(s?: string): string {
  if (!s) return '';
  return new Date(s).toLocaleDateString('uk-UA', { day: '2-digit', month:
'short' });
}

formatDateLong(s?: string): string {
  if (!s) return '';
  return new Date(s).toLocaleDateString('uk-UA', { day: '2-digit', month:
'long', year: 'numeric' });
}

formatPrice(n: number): string {
  return n.toLocaleString('uk-UA') + ' €';
}

mealLabel(meal: string): string {
  return ({ BB: 'Сніданок', HB: 'Напівпансіон', FB: 'Повний пансіон', AI:
'Все включено' } as Record<string, string>)[meal] ?? meal;
}

```

					2026.КВР.122.423.04.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		107

```

}

serviceIcon(type: string): string {
    return ({ INSURANCE: '🛡️', TRANSFER: '🚌', EXCURSION: '🏛️' } as
Record<string, string>)[type] ?? '♦️';
}

serviceIconBg(type: string): string {
    return ({ INSURANCE: '#E9F2FB', TRANSFER: '#FFF3EA', EXCURSION:
'#F0FDF4' } as Record<string, string>)[type] ?? 'var(--color-bg)';
}

touristLabel(count: number): string {
    if (count === 1) return 'дорослий';
    if (count >= 2 && count <= 4) return 'дорослих';
    return 'дорослих';
}
}
}

```

					2026.КВР.122.423.04.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		108

Додаток Д. Лістинг файлу «SecurityConfig.java»

```
package com.tourmaster.config;

import com.tourmaster.security.JwtAuthFilter;
import lombok.RequiredArgsConstructor;
import org.springframework.beans.factory.annotation.Value;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
import org.springframework.http.HttpMethod;
import org.springframework.security.authentication.AuthenticationManager;
import org.springframework.security.authentication.AuthenticationProvider;
import
org.springframework.security.authentication.dao.DaoAuthenticationProvider;
import
org.springframework.security.config.annotation.authentication.configuration
.AuthenticationConfiguration;
import
org.springframework.security.config.annotation.method.configuration.EnableM
ethodSecurity;
import
org.springframework.security.config.annotation.web.builders.HttpSecurity;
import
org.springframework.security.config.annotation.web.configuration.EnableWebS
ecurity;
import
org.springframework.security.config.annotation.web.configurers.AbstractHttp
Configurer;
import org.springframework.security.config.http.SessionCreationPolicy;
import org.springframework.security.core.userdetails.UserDetailsService;
import org.springframework.security.crypto.bcrypt.BCryptPasswordEncoder;
import org.springframework.security.crypto.password.PasswordEncoder;
import org.springframework.security.web.SecurityFilterChain;
import
```

					2026.KBP.122.423.04.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		109

```

org.springframework.security.web.authentication.UsernamePasswordAuthenticat
ionFilter;
import org.springframework.web.cors.CorsConfiguration;
import org.springframework.web.cors.CorsConfigurationSource;
import org.springframework.web.cors.UrlBasedCorsConfigurationSource;

import java.util.List;

@Configuration
@EnableWebSecurity
@EnableMethodSecurity
@RequiredArgsConstructor
public class SecurityConfig {

    private final JwtAuthFilter jwtAuthFilter;
    private final UserDetailsService userDetailsService;

    @Value("${app.cors.allowed-origins}")
    private String allowedOrigins;

    @Bean
    public SecurityFilterChain securityFilterChain(HttpSecurity http)
throws Exception {
        http
            .csrf(AbstractHttpConfigurer::disable)
            .cors(cors ->
cors.configurationSource(corsConfigurationSource()))
            .sessionManagement(session ->
session.sessionCreationPolicy(SessionCreationPolicy.STATELESS))
            .authorizeHttpRequests(auth -> auth
                .requestMatchers(HttpMethod.OPTIONS, "/*").permitAll()
                .requestMatchers(
                    "/api/auth/**",
                    "/actuator/health",

```

					2026.KBP.122.423.04.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		110

```

        "/swagger-ui.html", "/swagger-ui/**", "/v3/api-docs/**"
    ).permitAll()
    .requestMatchers(HttpMethod.GET,
        "/api/tours", "/api/tours/**"
    ).permitAll()
    .requestMatchers("/api/admin/**").hasRole("MANAGER")
    .anyRequest().authenticated()
)
    .authenticationProvider(authenticationProvider())
    .addFilterBefore(jwtAuthFilter,
UsernamePasswordAuthenticationFilter.class);

    return http.build();
}

@Bean
public CorsConfigurationSource corsConfigurationSource() {
    CorsConfiguration config = new CorsConfiguration();
    config.setAllowedOrigins(List.of(allowedOrigins));
    config.setAllowedMethods(List.of("GET", "POST", "PUT", "PATCH",
"DELETE", "OPTIONS"));
    config.setAllowedHeaders(List.of("Authorization", "Content-Type"));
    config.setAllowCredentials(true);

    UrlBasedCorsConfigurationSource source = new
UrlBasedCorsConfigurationSource();
    source.registerCorsConfiguration("/api/**", config);
    return source;
}

@Bean
public AuthenticationProvider authenticationProvider() {
    DaoAuthenticationProvider provider = new
DaoAuthenticationProvider();

```

					2026.KBP.122.423.04.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		111

```

        provider.setUserDetailsService(userDetailsService);
        provider.setPasswordEncoder(passwordEncoder());
        return provider;
    }

    @Bean
    public AuthenticationManager
authenticationManager(AuthenticationConfiguration config) throws Exception
    {
        return config.getAuthenticationManager();
    }

    @Bean
    public PasswordEncoder passwordEncoder() {
        return new BCryptPasswordEncoder();
    }
}

```

					2026.KBP.122.423.04.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		112

Додаток Е. Лістинг файлу «JwtService.java»

```
package com.tourmaster.security;

import io.jsonwebtoken.Claims;
import io.jsonwebtoken.Jwts;
import io.jsonwebtoken.security.Keys;
import lombok.extern.slf4j.Slf4j;
import org.springframework.beans.factory.annotation.Value;
import org.springframework.security.core.userdetails.UserDetails;
import org.springframework.stereotype.Service;

import javax.crypto.SecretKey;
import java.nio.charset.StandardCharsets;
import java.util.Date;
import java.util.Map;
import java.util.function.Function;

@Slf4j
@Service
public class JwtService {

    @Value("${app.jwt.secret}")
    private String secret;

    @Value("${app.jwt.access-expiration}")
    private long accessExpiration;

    @Value("${app.jwt.refresh-expiration}")
    private long refreshExpiration;

    public String generateAccessToken(UserDetails userDetails) {
        return buildToken(Map.of(), userDetails, accessExpiration);
    }
}
```

					2026.KBP.122.423.04.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		113

```

public String generateRefreshToken(UserDetails userDetails) {
    return buildToken(Map.of(), userDetails, refreshExpiration);
}

public String extractUsername(String token) {
    return extractClaim(token, Claims::getSubject);
}

public boolean isValidToken(String token, UserDetails userDetails) {
    final String username = extractUsername(token);
    return username.equals(userDetails.getUsername()) &&
!isTokenExpired(token);
}

private String buildToken(Map<String, Object> extraClaims, UserDetails
userDetails, long expiration) {
    return Jwts.builder()
        .claims(extraClaims)
        .subject(userDetails.getUsername())
        .issuedAt(new Date())
        .expiration(new Date(System.currentTimeMillis() +
expiration))
        .signWith(getSigningKey())
        .compact();
}

private boolean isTokenExpired(String token) {
    return extractExpiration(token).before(new Date());
}

private Date extractExpiration(String token) {
    return extractClaim(token, Claims::getExpiration);
}

```

					2026.KBP.122.423.04.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		114

```

private <T> T extractClaim(String token, Function<Claims, T>
claimsResolver) {
    final Claims claims = extractAllClaims(token);
    return claimsResolver.apply(claims);
}

private Claims extractAllClaims(String token) {
    return Jwts.parser()
        .verifyWith(getSigningKey())
        .build()
        .parseSignedClaims(token)
        .getPayload();
}

private SecretKey getSigningKey() {
    byte[] keyBytes = secret.getBytes(StandardCharsets.UTF_8);
    return Keys.hmacShaKeyFor(keyBytes);
}
}

```

					2026.KBP.122.423.04.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		115

Додаток Ж. Лістинг файлу «TourSpecification.java»

```
package com.tourmaster.repository;

import com.tourmaster.entity.Tour;
import com.tourmaster.entity.TourDate;
import com.tourmaster.entity.TourStatus;
import jakarta.persistence.criteria.*;
import org.springframework.data.jpa.domain.Specification;

import java.math.BigDecimal;
import java.time.LocalDate;
import java.util.ArrayList;
import java.util.List;

public class TourSpecification {

    public static Specification<Tour> filter(
        String q,
        String country,
        BigDecimal priceMin,
        BigDecimal priceMax,
        List<Short> stars,
        List<String> mealTypes,
        Integer durationMin,
        Integer durationMax,
        LocalDate departureDateFrom,
        LocalDate departureDateTo,
        String badge
    ) {
        return (root, query, cb) -> {
            List<Predicate> predicates = new ArrayList<>();

            // Only active tours
```

					2026.KBP.122.423.04.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		116

```

        predicates.add(cb.equal(root.get("status"),
TourStatus.ACTIVE));

        if (q != null && !q.isBlank()) {
            String pattern = "%" + q.toLowerCase() + "%";
            predicates.add(cb.or(
                cb.like(cb.lower(root.get("title")), pattern),
                cb.like(cb.lower(root.get("country")), pattern),
                cb.like(cb.lower(root.get("hotelName")), pattern)
            ));
        }

        if (country != null && !country.isBlank()) {
            predicates.add(cb.equal(cb.lower(root.get("country")),
country.toLowerCase()));
        }

        if (priceMin != null) {

predicates.add(cb.greaterThanOrEqualTo(root.get("priceFrom"), priceMin));
        }

        if (priceMax != null) {
            predicates.add(cb.lessThanOrEqualTo(root.get("priceFrom"),
priceMax));
        }

        if (stars != null && !stars.isEmpty()) {
            predicates.add(root.get("hotelStars").in(stars));
        }

        if (mealTypes != null && !mealTypes.isEmpty()) {
            predicates.add(root.get("mealType").in(mealTypes));
        }

```

					2026.KBP.122.423.04.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		117

```

        if (durationMin != null) {

predicates.add(cb.greaterThanOrEqualTo(root.get("durationNights"),
durationMin));
        }

        if (durationMax != null) {

predicates.add(cb.lessThanOrEqualTo(root.get("durationNights"),
durationMax));
        }

        if (badge != null && !badge.isBlank()) {
            predicates.add(cb.equal(root.get("badge"), badge));
        }

// Filter by departure date via tour_dates subquery
if (departureDateFrom != null || departureDateTo != null) {
    Subquery<Long> sub = query.subquery(Long.class);
    Root<TourDate> td = sub.from(TourDate.class);
    List<Predicate> datePreds = new ArrayList<>();
    datePreds.add(cb.equal(td.get("tour"), root));
    datePreds.add(cb.greaterThan(td.get("availableSeats"), 0));
    if (departureDateFrom != null) {

datePreds.add(cb.greaterThanOrEqualTo(td.get("departureDate"),
departureDateFrom));
        }
        if (departureDateTo != null) {

datePreds.add(cb.lessThanOrEqualTo(td.get("departureDate"),
departureDateTo));
        }
    }
}

```

					2026.KBP.122.423.04.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		118

```

        sub.select(td.get("id")).where(datePreds.toArray(new
Predicate[0]));
        predicates.add(cb.exists(sub));
    }

    // Avoid duplicate rows when joining
    if (query.getResultType() != Long.class) {
        query.distinct(true);
    }

    return cb.and(predicates.toArray(new Predicate[0]));
};
}
}

```

					2026.KBP.122.423.04.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		119

Додаток II. Контрольний список тестування вебсайту TourMaster

№	Тест-кейс	Очікуваний результат	Фактичний результат	Статус
1	Завантаження головної сторінки	Усі секції відображаються	Відповідає очікуванню	Пройдено
2	Перехід до каталогу через форму пошуку	URL отримує query-параметри, результати відфільтровані	Відповідає очікуванню	Пройдено
3	Клік на картку напрямку у сітці	Перехід до /tours?country=...	Відповідає очікуванню	Пройдено
4	Реєстрація з валідними даними	Автоматичний вхід, перенаправлення на головну сторінку	Відповідає очікуванню	Пройдено
5	Реєстрація з існуючим email	Toast: «обліковий запис вже існує»	Відповідає очікуванню	Пройдено
6	Вхід з невірним паролем	Toast з описом помилки, форма активна	Відповідає очікуванню	Пройдено
7	Оновлення JWT токена при 401	Запит повторюється автоматично, користувач не помічає	Відповідає очікуванню	Пройдено
8	Доступ до /booking без авторизації	Перенаправлення на /auth/login?returnUrl=/booking	Відповідає очікуванню	Пройдено
9	Фільтрація за країною	Список турів оновлено, URL містить country=...	Відповідає очікуванню	Пройдено
10	Комбінування кількох фільтрів одночасно	Результати відповідають усім активним критеріям	Відповідає очікуванню	Пройдено

№	Тест-кейс	Очікуваний результат	Фактичний результат	Статус
11	Нульовий результат пошуку	Відображається порожній стан з кнопкою скидання	Відповідає очікуванню	Пройдено
12	Перемикання grid / list	Режим змінюється без перезавантаження даних	Відповідає очікуванню	Пройдено
13	Пагінація	Перехід між сторінками, URL оновлюється (?page=N)	Відповідає очікуванню	Пройдено
14	Відновлення фільтрів при «назад» у браузері	Стан фільтрів відновлено з URL	Відповідає очікуванню	Пройдено
15	Повноекранний режим Galleria	Галерея відкривається на весь екран, закривається Esc	Відповідає очікуванню	Пройдено
16	Перемикання вкладок на сторінці туру	Вміст відповідної вкладки відображається без затримки	Відповідає очікуванню	Пройдено
17	Sticky-сайдбар при прокрутці	Сайдбар залишається у видимій зоні при прокрутці донизу	Відповідає очікуванню	Пройдено
18	Вибір дати вильоту	Ціна у сайдбарі оновлюється відповідно до price_override	Відповідає очікуванню	Пройдено
19	Кнопка «Забронювати» без вибору дати	Відображається підказка, перехід заблоковано	Відповідає очікуванню	Пройдено
20	Повернення до wizard після login через returnUrl	Редирект на /booking, стан кошика збережено	Відповідає очікуванню	Пройдено

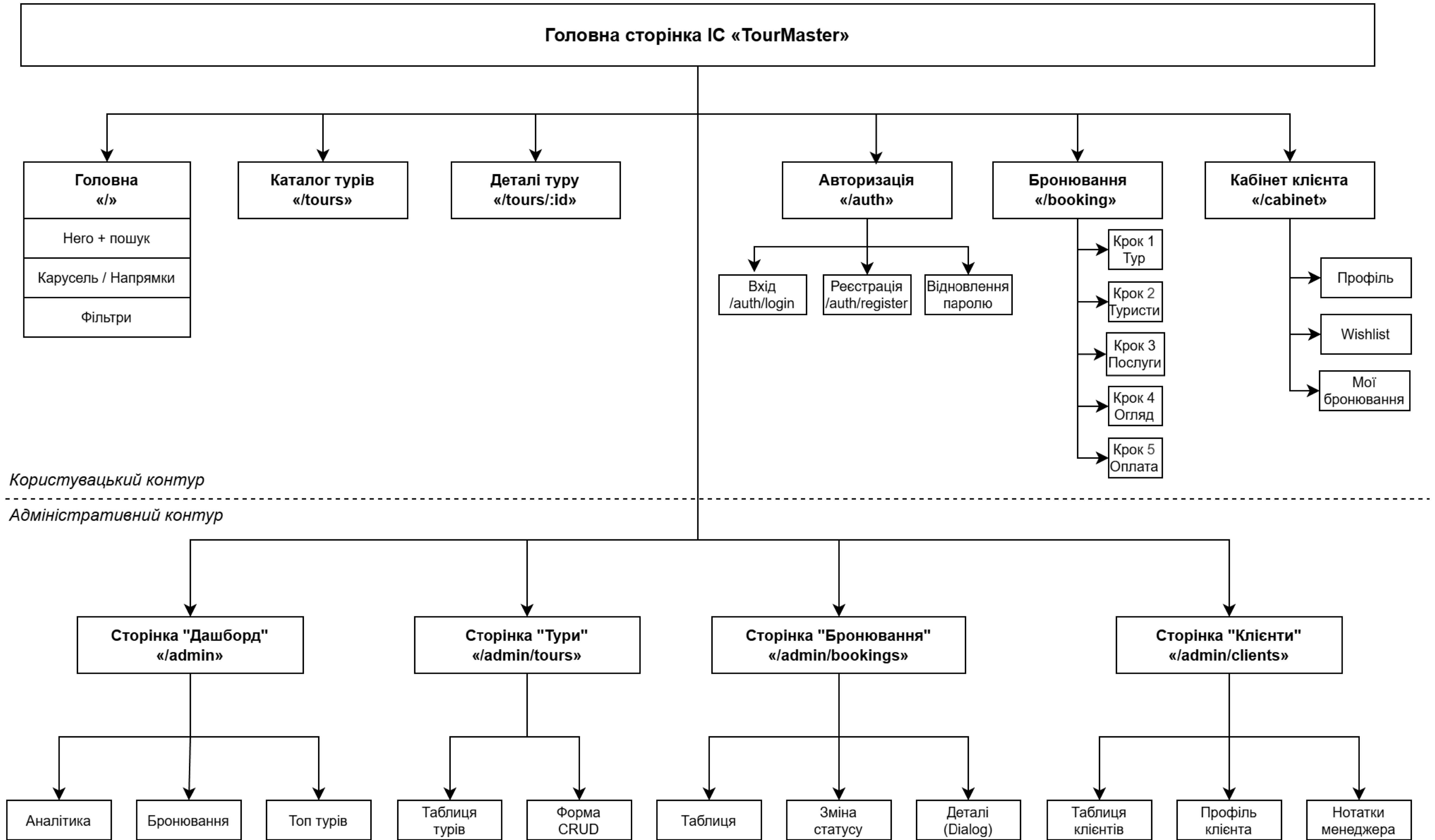
					2026.KBP.122.423.04.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		121

№	Тест-кейс	Очікуваний результат	Фактичний результат	Статус
21	Валідація кроку 2 (порожні поля туриста)	Кнопка «Далі» заблокована, поля підсвічені червоним	Відповідає очікуванню	Пройдено
22	Валідація терміну дії паспорта (минула дата)	Поле підсвічується, перехід заблоковано	Відповідає очікуванню	Пройдено
23	Вибір додаткових послуг (крок 3)	Сума у резюме оновлюється при кожному виборі	Відповідає очікуванню	Пройдено
24	Успішне завершення бронювання	Перехід на /booking/success з номером бронювання	Відповідає очікуванню	Пройдено
25	Відображення нового бронювання	Бронювання зі статусом NEW видно у списку	Відповідає очікуванню	Пройдено
26	Фільтрація бронювань за статусом	Відображаються лише бронювання обраного статусу	Відповідає очікуванню	Пройдено
27	Додавання туру до Wishlist	Іконка серця заповнюється, тур з'являється у /cabinet/wishlist	Відповідає очікуванню	Пройдено
28	Збереження змін профілю	Toast «Профіль оновлено», нові дані відображаються в шапці	Відповідає очікуванню	Пройдено
29	Доступ до /admin під роллю CLIENT	Перенаправлення на /, сторінка адміністратора не відкривається	Відповідає очікуванню	Пройдено
30	Створення нового туру (Dialog)	Тур з'являється у таблиці, каталог оновлюється	Відповідає очікуванню	Пройдено

					2026.KBP.122.423.04.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		122

№	Тест-кейс	Очікуваний результат	Фактичний результат	Статус
31	Редагування туру	Dialog відкривається з поточними даними, зміни зберігаються	Відповідає очікуванню	Пройдено
32	Архівування туру	ConfirmDialog → тур зникає з активного каталогу	Відповідає очікуванню	Пройдено
33	Зміна статусу бронювання (inline)	PATCH-запит виконується, бейдж статусу оновлюється	Відповідає очікуванню	Пройдено
34	Перегляд профілю клієнта з нотатками	Всі нотатки менеджерів відображаються у хронологічному порядку	Відповідає очікуванню	Пройдено
35	Мобільний вигляд (390 px) — шапка	Горизонтальне меню приховано, відображається MobileTabBar	Відповідає очікуванню	Пройдено
36	Мобільний вигляд — фільтри каталогу	Фільтри відкриваються як PrimeNG Sidebar слайдер	Відповідає очікуванню	Пройдено
37	Мобільний вигляд — сторінка туру	Сайдбар переміщується вгору, галерея одноколонна	Відповідає очікуванню	Пройдено
38	Планшетний вигляд (820 px) — каталог	Перехід від 3 до 2 колонок, фільтри залишаються відкритими	Відповідає очікуванню	Пройдено

Схема структурна клієнтської частини



Користувацький контур

Адміністративний контур

					2026.KBP.122.423.04.00.00 CC			
Зм.	Арх.	№ докум.	Підпис.	Дата	Розробка інформаційної системи підбору та бронювання турів для агенції «TourMaster» Схема структурна клієнтської частини	Лист	Маса	Масштаб
Розроб.	Гусєв ДІ							
Перевір.	Слободян Р.О.					Архи	Архи	1
Текст						ВСП ТФК ТНТУ КН-423		
Рецензент						м. Тернопіль		
Начальник	Примічк. В.А.				Формат А1			
Затв.					Копія			

UML-діаграма варіантів використання



					2026.KBP.122.4.23.04.00.00 ДВ				
Знак	Арк.	№ докум.	Підпис	Дата	Розробка інформаційної системи підбору та бронювання турів для агенції «TourMaster» UML-діаграма варіантів використання	Лист	Маса	Масштаб	
Розроб	Гусєв ДІ					Аркци	Аркциб	1	
Перевір	Слободян Р.О.					ВСП ТФК ТНТУ КН-423			
Текст						м. Тернопіль			
Рецензент									
Начальник	Пріймак В.А.								
Затв.									
						Копія			
						Формат А1			

flyway_schema_history	
installed_rank	integer NN
version	varchar(50)
description	varchar(200) NN
type	varchar(20) NN
script	varchar(1000) NN
checksum	integer
installed_by	varchar(100) NN
installed_on	timestamp NN
execution_time	integer NN
success	boolean NN

manager_notes	
id	bigserial
user_id	bigint NN
manager_id	bigint NN
note	text NN
created_at	timestamp NN

refresh_tokens	
id	bigserial
token	varchar(512) NN
user_id	bigint NN
expires_at	timestamp NN
created_at	timestamp NN

notifications	
id	bigserial
user_id	bigint NN
booking_id	bigint
type	varchar(50) NN
subject	varchar(255)
body	text
sent_at	timestamp
status	varchar(20) NN

wishlist	
user_id	bigint NN
tour_id	bigint NN
added_at	timestamp NN

users	
id	bigserial
email	varchar(255) NN
password	varchar(255) NN
first_name	varchar(100) NN
last_name	varchar(100) NN
phone	varchar(20)
role	varchar(20) NN
loyalty_points	integer NN
created_at	timestamp NN
updated_at	timestamp NN

reviews	
id	bigserial
user_id	bigint NN
tour_id	bigint NN
booking_id	bigint NN
rating	smallint NN
comment	text
created_at	timestamp NN

bookings	
id	bigserial
user_id	bigint NN
tour_date_id	bigint NN
status	varchar(20) NN
total_price	numeric(12,2) NN
discount	numeric(12,2) NN
manager_note	text
created_at	timestamp NN
updated_at	timestamp NN

booking_extra_services	
booking_id	bigint NN
extra_service_id	bigint NN

tours	
id	bigserial
title	varchar(255) NN
description	text
country	varchar(100) NN
city	varchar(100)
hotel_name	varchar(255)
hotel_stars	smallint
meal_type	varchar(50)
duration_nights	smallint NN
price_from	numeric(12,2) NN
old_price	numeric(12,2)
badge	varchar(50)
rating	numeric(2,1)
reviews_count	integer
image_url	varchar(512)
gallery_urls	text[]
status	varchar(20) NN
created_at	timestamp NN
updated_at	timestamp NN

tour_dates	
id	bigserial
tour_id	bigint NN
departure_date	date NN
return_date	date NN
departure_city	varchar(100)
total_seats	integer NN
available_seats	integer NN
price_override	numeric(12,2)
created_at	timestamp NN

booking_items	
id	bigserial
booking_id	bigint NN
first_name	varchar(100) NN
last_name	varchar(100) NN
birth_date	date NN
passport_number	varchar(50) NN
passport_expiry	date NN

extra_services	
id	bigserial
name	varchar(255) NN
description	text
price	numeric(10,2) NN
type	varchar(50) NN

					2026.KBP.122.4.23.04.00.00 БД		
Знак	Арх	№ докум	Підпис	Дата	Розробка інформаційної системи підбору та бронювання турів для агенції «TourMaster»		
Розроб	Гусев ДІ				ER-діаграма бази даних		
Перевір	Слободян Р.О.				Архіви	Архів	1
Текст					ВСП ТФК ТНТУ		
Рецензент					м. Тернопіль		
Нконтр	Примак В.А.				Формат А1		
Затв					Копія		

Таблиця техніко-економічних показників

№ п/п	Показник	Одиниці вимірювання	Значення
1	Мови програмування	–	Java, TypeScript
2	Технології клієнтської частини	–	Angular 22, PrimeNG 21, Tailwind CSS 4
3	Технології серверної частини	–	Spring Boot 3.3.4, Spring Security 6
4	База даних	–	PostgreSQL 15
5	Середовище розробки	–	IntelliJ IDEA
6	Захист	–	JWT, Spring Security
7	Собівартість	грн.	66 273,57
8	Плановий прибуток	грн.	37 113,20
9	Ціна	грн.	103 386,77
10	Чиста теперішня вартість	грн.	26 021,46
11	Термін окупності	рік	2,1

					2026.KBP.122.423.04.00.00 ТБ		
Зм.	Арк.	№ докум.	Підпис.	Дата.	Розробка інформаційної системи підбору та бронювання турів для агенції «TourMaster» Таблиця техніко-економічних показників		
Розроб.	Гусев ДІ						
Перевір.	Слободян Р.О.						
Т.контр.							
Рецензент							
Н.контр.	Примак В.А.				Арк.циф.	Арх.циф.	1
Затв.					ВПТ ТФК ТНТУ КН-423 м. Тернопіль		
					Копіював Формат А1		