

Міністерство освіти і науки України

Відокремлений структурний підрозділ «Тернопільський фаховий коледж
Тернопільського національного технічного університету імені Івана Пулюя»

(повне найменування вищого навчального закладу)

Відділення інформаційних технологій, менеджменту, туризму
та підготовки іноземних громадян

(назва відділення)

Циклова комісія комп'ютерної інженерії

(повна назва циклової комісії)

ПОЯСНЮВАЛЬНА ЗАПИСКА

до кваліфікаційної роботи

фахового молодшого бакалавра

(освітньо-професійного ступеня)

на тему: Розробка вебсайту для барберів “Briolin”

Виконав: студент IV курсу, групи KI-412

Спеціальності 123 Комп'ютерна інженерія
(шифр і назва спеціальності)

Софія МИНЬКО
(ім'я та прізвище)

Керівник Леся ШТОКАЛО
(ім'я та прізвище)

Рецензент _____
(ім'я та прізвище)

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

Відділення **інформаційних технологій, менеджменту, туризму
та підготовки іноземних громадян**

Циклова комісія **комп'ютерної інженерії**

Освітньо-професійний ступінь **фаховий молодший бакалавр**

Освітньо-професійна програма: **Обслуговування комп'ютерних та систем і мереж**

Спеціальність: **123 Комп'ютерна інженерія**

Галузь знань: **12 Інформаційні технології**

ЗАТВЕРДЖУЮ

Голова циклової комісії

комп'ютерної інженерії

_____ Андрій ЮЗЬКІВ

“31” березня 2025 року

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

Минько Софії Миколаївни

(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи: **Розробка веб-сайту для барберів «Briolin»**

керівник роботи **Штокало Леся Ярославівна**

(прізвище, ім'я, по батькові)

Затверджені наказом Відокремленого структурного підрозділу «Тернопільський фаховий коледж Тернопільського національного технічного університету імені Івана Пулюя» від 28.03.2025р № 4/9-166а.

2. Строк подання студентом роботи: **13 червня 2025 року.**

3. Вихідні дані до роботи: **Visual Studio Code, Microsoft Edge, TypeScript, HTML, CSS, Node.js, Express.js, MongoDB, Angular.**

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити):
Загальний розділ. Розробка технічного та робочого проєкту. Спеціальний розділ. Економічний розділ. Охорона праці та безпека життєдіяльності.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

- структурна схема сайту;
- блок-схема скрипта;
- текст програми;
- таблиця техніко-економічних показників.

6. Консультанти розділів роботи

Розділ	Ім'я, прізвище та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Економічний розділ	Богдана МАРТИНЮК викладач		
Охорона праці та безпека життєдіяльності	Володимир ШТОКАЛО викладач		

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Отримання і аналіз технічного завдання	01.04	
2	Збір і узагальнення інформації	05.05	
3	Написання першого розділу	16.05	
4	Розробка технічного та робочого проекту	23.05	
5	Написання спеціального розділу	30.05	
6	Розрахунок економічної частини	02.06	
7	Написання розділу охорони праці	04.06	
8	Виконання графічної частини	09.06	
9	Оформлення проєкту	11.06	
10	Погодження нормоконтролю	12.06	
11	Попередній захист роботи	13.06	
12	Захист кваліфікаційної роботи		

7. Дата видачі завдання: 01 квітня 2025 року

Студент

(підпис)

Софія МИНЬКО

(ім'я та прізвище)

Керівник роботи

(підпис)

Леся ШТОКАЛО

(ім'я та прізвище)

АНОТАЦІЯ

Минько С. М. Розробка вебсайту для барберів «Briolin»: кваліфікаційна робота на здобуття освітньо-професійного ступеня фахового молодшого бакалавра за спеціальністю 123 Комп'ютерна інженерія. Тернопіль: ВСП «ТФК ТНТУ», 2025. 78 с.

Кваліфікаційна робота присвячена розробці спеціалізованого вебсайту для барберів під назвою «Briolin». У межах роботи проаналізовано сучасні технології веброзробки та обґрунтовано вибір стеку MEAN (MongoDB, Express, Angular, Node.js) як основи для створення інформаційно-блогової платформи. Сайт призначений для барберів, власників барбершопів та поціновувачів барбер-культури з метою самопрезентації, обміну досвідом, просування послуг та побудови професійної спільноти.

У процесі реалізації проєкту здійснено аналітичний огляд існуючих рішень, сформульовано технічне завдання, розроблено архітектуру та інтерфейс вебзастосунку, виконано проєктування логіки системи, розроблено програмний код і проведено тестування функціоналу. Особливу увагу приділено адаптивному дизайну, зручності користування, безпеці даних та багатомовності інтерфейсу.

Результати роботи мають практичну цінність для фахівців індустрії краси, можуть бути використані у професійній діяльності, маркетингу, освіті, а також слугувати прикладом прикладної розробки вебсервісів для вузькоспеціалізованої аудиторії.

Ключові слова: вебсайт, барбери, блог, MEAN stack, веброзробка, Angular, MongoDB, Node.js, барбер-культура.

ANNOTATION

Mynko S.M. Development of a website for barbers “Briolin”: qualification work for the degree of professional junior bachelor in specialty 123 Computer Engineering. Ternopil: VSP “TFC TNTU”, 2025. 78 c.

The qualification work is devoted to the development of a specialized website for barbers called “Briolin”. The work analyzes modern web development technologies and substantiates the choice of the MEAN stack (MongoDB, Express, Angular, Node.js) as the basis for creating an information and blog platform. The site is designed for barbers, barbershop owners, and barber culture lovers to present themselves, share experiences, promote services, and build a professional community.

In the course of the project implementation, an analytical review of existing solutions was carried out, the terms of reference were formulated, the architecture and interface of the web application were developed, the system logic was designed, the program code was developed, and the functionality was tested. Particular attention was paid to adaptive design, usability, data security, and multilingualism of the interface.

The results of the work are of practical value for specialists in the beauty industry, can be used in professional activities, marketing, education, and also serve as an example of applied web service development for a highly specialized audience.

Keywords: website, barbers, blog, MEAN stack, web development, Angular, MongoDB, Node.js, barber culture.

ЗМІСТ

ВСТУП	8
1. ЗАГАЛЬНИЙ РОЗДІЛ	9
1.1 Аналітичний огляд існуючих рішень	9
1.2 Технічне завдання	13
1.2.1 Найменування та область застосування	13
1.2.2 Призначення розробки.....	14
1.2.3 Вимоги до програмного забезпечення	15
1.2.4 Вимоги до програмної документації.....	22
1.2.5 Техніко-економічні показники	24
1.2.6 Стадії та етапи розробки.....	26
1.2.7 Порядок контролю та прийому	27
2. РОЗРОБКА ТЕХНІЧНОГО ТА РОБОЧОГО ПРОЕКТУ	29
2.1 Постановка задачі на розробку програмного забезпечення	29
2.1.1 Огляд програмних засобів для розробки веб-сайту.....	30
2.2 Опис та обґрунтування вибору методу організації вхідних та вихідних даних	32
2.3 Розробка алгоритму	33
2.3.1 Зовнішнє проектування програми	33
2.3.2 Проектування логіки програми.....	41
2.4 Визначення інформаційних зв'язків програмних компонентів	43
2.5 Написання скрипта.....	45
2.5 Тестування сайту	47
3. СПЕЦІАЛЬНИЙ РОЗДІЛ	50
3.1 Технологія MEAN	50
3.2 Процес встановлення програмного комплексу на ПК	51

					2025.КВР.123.412.10.00.00 ПЗ								
Змн.	Арк.	№ докум.	Підпис	Дата									
Розроб.		МИНЬКО С. М.			Розробка вебсайту для барберів “Briolin” Пояснювальна записка				Літ.		Арк.	Аркушів	
Перевір.		ШТОКАЛО Л.Я											
Реценз.													
Н. Контр.		ЮЗЬКІВ А.							ВСП «ТФК ТНТУ» група КІ-412 м. Тернопіль				
Затверд.													

3.3 Інструкція з Розміщення сайту на хостингу	54
3.4 Інструкція з експлуатації веб-сайту	55
4 ЕКОНОМІЧНИЙ РОЗДІЛ	57
4.1 Визначення стадій технологічного процесу та загальної тривалості проведення НДР	57
4.2 Визначення витрат на оплату праці та відрахувань на соціальні заходи..	58
4.3 Розрахунок витрат на електроенергію	60
4.5 Розрахунок суми амортизаційних відрахувань	60
4.6 Обчислення накладних витрат	61
4.7 Складання кошторису витрат та визначення собівартості НДР	61
4.8 Розрахунок ціни НДР	62
4.9 Визначення економічної ефективності і терміну окупності капітальних вкладень.....	62
5. ОХОРОНА ПРАЦІ ТА БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ.....	64
5.1 Шкідливі чинники комп'ютерних технологій та захист від них	64
5.2 Вимоги до освітлення виробничих приміщень	66
5.3 Психологія безпеки праці	67
ВИСНОВКИ.....	70
ПЕРЕЛІК ПОСИЛАНЬ	71
ДОДАТОК А - КОД СКРИПТА HTML	74
ДОДАТОК В - КОД СКРИПТА CSS	76

ВСТУП

У сучасному цифровому світі наявність якісного веб-сайту є одним із ключових чинників успішного просування бізнесу. Особливо це актуально для індустрії краси, яка стрімко розвивається та вимагає нових підходів до комунікації з клієнтами. Барбершопи — сучасні перукарні для чоловіків — активно впроваджують цифрові інструменти для створення власного іміджу, залучення нових відвідувачів і підтримки лояльності постійних клієнтів.

Незважаючи на широкий вибір готових рішень, на ринку все ще відчувається потреба у спеціалізованих веб-ресурсах, які б враховували унікальні особливості барбер-культури. Саме тому виникла ідея створити веб-сайт "Briolin", який виконує функції блогу, що об'єднує барберів, клієнтів та зацікавлених у цій сфері осіб. Основне призначення ресурсу — інформування користувачів про новини, техніки стрижки, догляд за волоссям і бородою, а також надання можливості обміну досвідом між майстрами.

Метою кваліфікаційної роботи є розробка веб-сайту для барберів, який стане інформаційним майданчиком, зручним у використанні як для фахівців галузі, так і для широкої аудиторії. У межах роботи реалізовано дизайн, архітектуру та функціонал, що забезпечують сучасний вигляд, адаптивність до різних пристроїв і простоту керування контентом.

Результати розробки можуть бути використані у професійній діяльності барберів, власників барбершопів, а також у навчальних закладах для ознайомлення студентів із прикладною розробкою веб-додатків. Запропоноване рішення відповідає актуальним вимогам до створення інформаційних ресурсів та демонструє ефективний підхід до веб-програмування з урахуванням особливостей конкретної цільової аудиторії.

					2025.КВР.123.412.10.00.00 ПЗ	Арк.
						8
Змн.	Арк.	№ докум.	Підпис	Дата		

1. ЗАГАЛЬНИЙ РОЗДІЛ

1.1 Аналітичний огляд існуючих рішень

У сфері надання послуг велике значення має наявність онлайн-ресурсів, що дозволяють фахівцям ефективно презентувати свою діяльність, комунікувати з клієнтами та створювати навколо себе професійну спільноту. Барбери — не виняток. У світі активно функціонують десятки сайтів, які обслуговують потреби барберів як блогерів, підприємців і митців. Проте більшість із них або вузькоспеціалізовані, або є частиною великих платформ. Це створює нішу для розробки персоналізованого тематичного блогу на кшталт "Briolin".

У процесі створення таких сайтів важливу роль відіграє вибір відповідних технологій, які визначають функціональні можливості, гнучкість, стабільність та перспективи розвитку проєкту. На сьогодні у веб-розробці активно застосовуються різноманітні стекові рішення, серед яких особливу увагу заслуговують Ruby Stack, LAMP Stack та MEAN Stack. Кожен із зазначених підходів має власні особливості, які впливають на швидкість розробки, зручність підтримки, продуктивність системи та можливості її масштабування.

Ruby Stack базується на мові програмування Ruby у поєднанні з веб-фреймворком Ruby on Rails. Основною перевагою цього рішення є висока швидкість створення веб-застосунків завдяки використанню принципів автоматизації конфігурації та зменшення дублювання коду. Завдяки великій кількості готових бібліотек цей стек активно застосовується при розробці стартапів, систем управління контентом, SaaS-сервісів, інтернет-магазинів та соціальних платформ. Разом з тим, його використання потребує досвіду оптимізації продуктивності, особливо у випадках роботи з великими обсягами даних та високим навантаженням[8].

					2025.КВР.123.412.10.00.00 ПЗ	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дата		

LAMP Stack є одним із найстаріших та найстабільніших рішень у галузі веб-розробки, яке широко застосовується для створення корпоративних сайтів, блогів, порталів, форумів, інформаційних систем і електронної комерції. Його стабільна робота, доступність численних навчальних матеріалів та підтримка великої кількості готових систем управління контентом зумовили популярність цього підходу протягом багатьох років. Проте сучасні вимоги до інтерактивності й масштабованості веб-застосунків поступово зміщують акценти у бік новітніх рішень [5].

MEAN Stack є сучасним технологічним підходом, в якому використовується єдина мова програмування JavaScript на всіх рівнях розробки — як для клієнтської, так і для серверної частини. Він включає базу даних MongoDB, серверний фреймворк Express.js, клієнтський фреймворк Angular та серверне середовище Node.js[1].

Завдяки цьому забезпечується гнучкість, масштабованість, висока продуктивність та інтерактивність веб-застосунків. Такий підхід дозволяє ефективно реалізовувати односторінкові застосунки, сервіси реального часу, а також мобільні бекенди. Разом з тим, він потребує глибоких технічних знань у сфері асинхронної обробки даних та роботи з базами NoSQL. [4].

Аналіз сучасних аналогів дозволяє більш чітко окреслити потребу у створенні нового рішення. Розглянемо деякі з них.

1. Booksy (див. рис. 1.1) - платформа онлайн-запису до майстрів краси [22]. Виконує функції такі як: пошук барбера, перегляд профілю, відгуки, онлайн-бронювання.

Її перевагою є простота використання для клієнтів, інтеграція з календарем. А недоліком те, що вона орієнтована переважно на сервіс запису, не містить блогів, контенту від барберів чи соціальної складової.

					2025.КВР.123.412.10.00.00 ПЗ	Арк.
						10
Змн.	Арк.	№ докум.	Підпис	Дата		

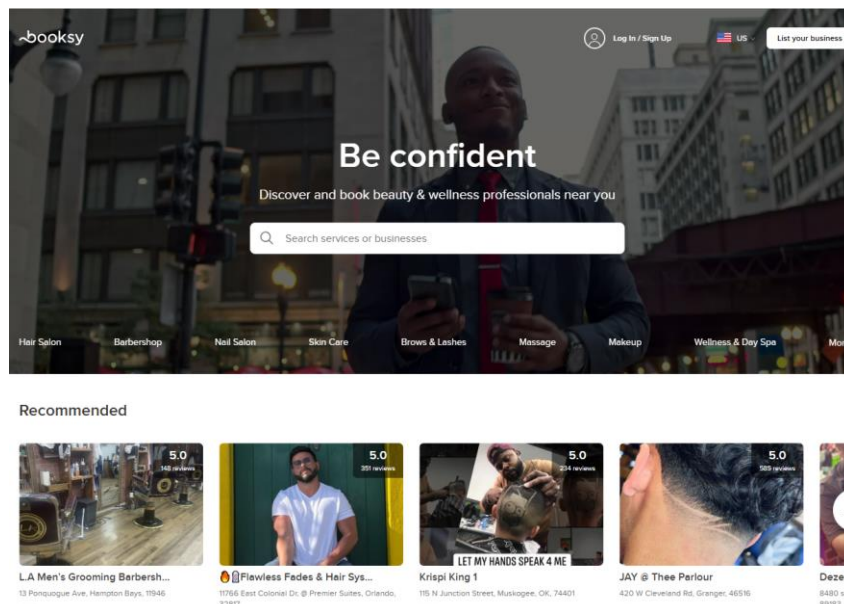


Рисунок 1.1 - Платформа онлайн-запису Booksy

2. BarberShopCo (barbershop.co.nz) - корпоративний сайт барбершопу (див. рис.1.2). Містить інформацію про послуги, команди, локації, блог-новини, магазин. Перевагою цієї платформи є те, що вона комбінує інформативну складову з елементами просування бренду. Але вона не є платформою для спільноти барберів — контент генерує лише компанія [21].

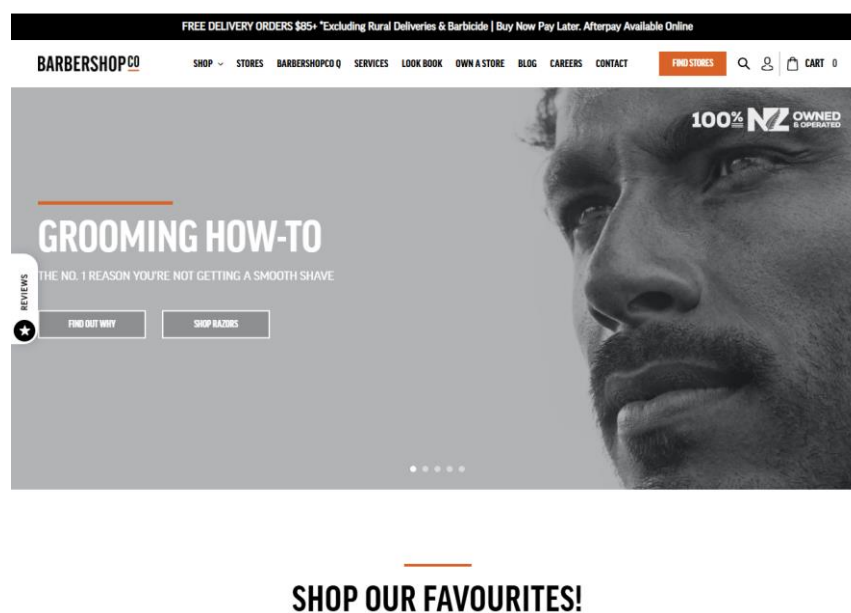


Рисунок 1.2 - Корпоративний сайт барбершопу BarberShopCo

					2025.КВР.123.412.10.00.00 ПЗ	Арк.
						11
Змн.	Арк.	№ докум.	Підпис	Дата		

3. YouTube та Instagram профілі барберів (наприклад, @barber.josh.o.p) - Соціальні мережі як майданчик для просування. Містить у собі можливості публікація відео, технік стрижки, взаємодії з підписниками. Перевагаю такої платформи є наявність великої аудиторія, візуальний контент. Але наявна залежність від алгоритмів соцмереж, відсутність структурованості матеріалів, складність пошуку по темах [30].

4. The Barber Post (thebarberpost.com) - тематичний блог про барбер-культуру [35]. На ньому розміщені статті, інтерв'ю, новини індустрії. Перевагою цього блогу є якісний контент. Але відсутній інтерактив, платформа закрита для зовнішніх авторів.

На основі цих даних може провести порівняльний аналіз усіх платформ. Він буде у вигляді таблиці. (Таблиця 1.1)

Таблиця 1.1 - Порівняльний аналіз усіх платформ

Платформа	Контент	Онлайн-запис	Блог	Спільнота	Адаптація під локальні ринки
Booksy	Ні	Так	Ні	Ні	Так
BarberShopCo	Так	Частково	Так	Ні	Ні
Соцмережі	Так	Ні	Так	Частково	Так
The Barber Post	Так	Ні	Так	Ні	Ні
Briolin (заплановано)	Так	Ні	Так	Так	Так

Аналіз існуючих рішень демонструє, що жодна з поточних платформ не надає повноцінної можливості для самопрезентації барберів у форматі блогу з гнучким контентом, можливістю вести авторську стрічку, публікувати галереї, статті, навчальні матеріали, комунікувати в коментарях та створювати спільноту. Проєкт "Briolin" має на меті закрити цю нішу, поєднавши функціональність блогу, інформативність портфоліо та елементи соціальної взаємодії.

Таким чином, проведений аналіз існуючих рішень показав, що на сучасному ринку відсутня комплексна платформа, яка б задовольняла потреби барберів у повноцінній самопрезентації, створенні авторського контенту, формуванні професійної спільноти та зручній взаємодії з аудиторією. Існуючі сервіси або обмежуються наданими функціональними можливостями, або мають вузьку спеціалізацію, орієнтовану на окремі аспекти діяльності майстрів.

У зв'язку з цим доцільним є розроблення нового веб-сайту "Briolin", який поєднуватиме переваги блогу, професійного портфоліо та соціальної платформи. Враховуючи проведене технічне дослідження різних стеків технологій, для реалізації проєкту було обрано MEAN Stack, який забезпечує необхідну гнучкість, масштабованість, продуктивність та відповідає сучасним тенденціям веб-розробки.

1.2 Технічне завдання

1.2.1 Найменування та область застосування

Повне найменування програмного продукту: Веб-сайт для барберів "Briolin".

Позначення: "Briolin" — блогова платформа для барберів.

Область застосування:

					2025.КВР.123.412.10.00.00 ПЗ	Арк.
						13
Змн.	Арк.	№ докум.	Підпис	Дата		

Програмний виріб призначений для використання у сфері перукарського мистецтва та догляду за зовнішністю, зокрема в галузі барберінгу. Сайт створюється як інформаційно-блогова платформа, орієнтована на барберів, стилістів, власників барбершопів та поціновувачів барбер-культури. Основна мета — надати спеціалістам можливість вести особистий блог, ділитися техніками, оглядами засобів, прикладами робіт, комунікувати з колегами та клієнтами.

Об'єктом використання сайту можуть бути як окремі барбери, що бажають створити власний бренд та поділитися професійними напрацюваннями, так і цілі барбершопи, які прагнуть вести корпоративні блоги, публікувати новини салону, розміщувати приклади робіт своїх працівників та залучати нових клієнтів. Окрім того, веб-сайт може бути впроваджений у навчальні заклади, професійні спільноти, платформи онлайн-освіти, а також у якості додаткового інструмента для розвитку галузевих тематичних порталів.

1.2.2 Призначення розробки

Експлуатаційне призначення веб-сайту «Briolin» полягає в забезпеченні цифрової присутності фахівців барбер-сфери в інтернет-просторі. Головна мета використання ресурсу — надати барберам та іншим користувачам інструмент для самовираження, просування власних послуг, обміну професійними знаннями, залучення нових клієнтів та формування репутації в професійному середовищі. Завдяки цьому сайту фахівці можуть більш ефективно презентувати свої роботи, знайомити аудиторію з новими техніками, продуктами й трендами, а також вибудовувати ділові зв'язки з іншими майстрами та потенційними клієнтами[25].

Функціональне призначення платформи полягає у наданні широкого функціоналу, що дозволяє досягати вищезазначених цілей.

					2025.КВР.123.412.10.00.00 ПЗ	Арк.
						14
Змн.	Арк.	№ докум.	Підпис	Дата		

Зокрема, користувачі зможуть створювати особисті облікові записи, публікувати текстові, фото- та відеоматеріали, вести блоги, ділитися техніками виконання стрижок або догляду, оглядати інструменти й косметичні засоби, взаємодіяти через коментарі та повідомлення, формувати портфоліо робіт. Додаткові функції включають систему підписки на інших користувачів, фільтрацію контенту за тематиками, категоріями, хештегами, а також можливість інтеграції з соціальними мережами для розширення охоплення аудиторії. Завдяки адаптивному дизайну платформа буде доступною як на стаціонарних комп'ютерах, так і на мобільних пристроях, що забезпечить зручність її використання у будь-яких умовах[35].

1.2.3 Вимоги до програмного забезпечення

Для забезпечення ефективної реалізації веб-сайту «Briolin» визначено перелік функціональних і нефункціональних вимог, що дозволять повноцінно реалізувати поставлену задачу та забезпечити зручність користування ресурсом.

У рамках розробки веб-сайту передбачено обробку вхідних даних, які поданні у таблиці 1.2.

Вхідні дані для програмного комплексу надходять у вигляді інформації, яку користувачі вводять через веб-інтерфейс. До таких даних належать: дані для реєстрації (ім'я, логін, email, пароль), дані для входу (логін та пароль), інформація для створення, редагування або видалення постів (заголовок, текст, категорія, фото), а також запити на пошук чи фільтрацію публікацій, зміна профілю користувача та вибір мови інтерфейсу.

Всі ці дані подаються у вигляді веб-форм або інтерактивних елементів інтерфейсу за ініціативою користувача у будь-який момент роботи з системою.

					2025.КВР.123.412.10.00.00 ПЗ	Арк.
						15
Змн.	Арк.	№ докум.	Підпис	Дата		

Таблиця 1.2. - Вхідні дані

Назва вхідного повідомлення	Форма подання	Періодичність надходження	Джерело даних
Дані для реєстрації	Веб-форма	За потреби	Користувач
Дані для входу	Веб-форма	За потреби	Користувач
Дані нового поста	Веб-форма	За потреби	Користувач (автор)
Дані для редагування/видалення поста	Веб-форма	За потреби	Користувач (автор/адмін)
Запит на пошук/фільтрацію	Веб-форма	За потреби	Користувач

Джерелами вхідної інформації є користувачі сайту, які взаємодіють із інтерфейсом через браузер.

Вихідна інформація формується у вигляді веб-сторінок, повідомлень, карток постів, профілів користувачів та адміністративних панелей. Користувачі отримують результати у вигляді списків постів, детальної інформації про окремі публікації, повідомлень про успішне виконання дій або помилки, а також персональних даних профілю.

Адміністратор додатково має доступ до списку користувачів та їхніх публікацій. Вся вихідна інформація генерується за запитом користувача або у відповідь на певні дії в системі.

Всю вихідну інформацію можна переглянути у таблиці 1.3.

					2025.КВР.123.412.10.00.00 ПЗ	Арк.
						16
Змн.	Арк.	№ докум.	Підпис	Дата		

Таблиця 1.3 - Вихідна інформація

Назва вихідного повідомлення	Форма подання	Періодичність видання	Користувач інформації
Список постів	Веб-сторінка	За запитом	Всі користувачі
Детальна інформація про пост	Веб-сторінка	За запитом	Всі користувачі
Список користувачів (адмін)	Веб-сторінка	За запитом	Адміністратор
Повідомлення про помилки	Веб-інтерфейс	За потреби	Всі користувачі
Повідомлення про успішні дії	Веб-інтерфейс	За потреби	Всі користувачі

Функціональні можливості програмного комплексу охоплюють реєстрацію та автентифікацію користувачів, створення, редагування та видалення постів, пошук і фільтрацію публікацій за категоріями, адміністрування користувачів, а також підтримку багатомовності інтерфейсу [20]. Для забезпечення коректної роботи всі форми проходять валідацію як на клієнтській, так і на серверній стороні. Доступ до адміністративних функцій обмежується роллю користувача. Всі дії, пов'язані зі зміною даних, супроводжуються перевіркою прав доступу та коректності введених даних. Система захищена від несанкціонованого доступу за допомогою токенів авторизації та розмежування ролей.

Реєстрація користувача передбачає обробку даних, які вводяться у відповідну форму: ім'я, логін, email та пароль. Після надсилання ці дані перевіряються на унікальність логіна та email, а також на відповідність формату (наприклад, email має містити символ "@", пароль — не менше шести

					2025.КВР.123.412.10.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		17

символів). У разі успішної перевірки створюється новий запис у базі даних, а користувач отримує повідомлення про успішну реєстрацію. У випадку помилки система повертає відповідне повідомлення з поясненням причини.

Авторизація користувача здійснюється шляхом введення логіна та пароля. Система перевіряє відповідність введених даних записам у базі. У разі успіху користувач отримує токен доступу, який використовується для подальшої ідентифікації у системі. Якщо дані некоректні, виводиться повідомлення про помилку.

Створення поста виконується авторизованим користувачем через спеціальну форму, де вказуються заголовок, текст, категорія та додається фото. Всі поля проходять валідацію: заголовок і текст не можуть бути порожніми, категорія має відповідати одному з визначених значень, фото — відповідати обмеженням за розміром і форматом. Після успішної перевірки дані зберігаються у базі, а користувач отримує підтвердження про створення поста.

Редагування та видалення постів доступне лише автору поста або адміністратору. При редагуванні користувач змінює дані у формі, які знову проходять валідацію. Видалення поста супроводжується перевіркою прав доступу та підтвердженням дії.

Пошук і фільтрація постів здійснюються за допомогою введення ключових слів або вибору категорії. Система обробляє запит, формує відповідний запит до бази даних і повертає користувачу список постів, що відповідають критеріям.

Адміністрування користувачів передбачає можливість перегляду списку користувачів, їхніх постів, а також видалення користувачів. Ці функції доступні лише користувачам із роллю адміністратора.

Багатомовність реалізується шляхом вибору мови інтерфейсу. При зміні мови всі текстові елементи інтерфейсу автоматично перекладаються відповідно до обраної мови.

					2025.КВР.123.412.10.00.00 ПЗ	Арк.
						18
Змн.	Арк.	№ докум.	Підпис	Дата		

Всі функції супроводжуються перевіркою прав доступу, валідацією даних, обробкою виняткових ситуацій (наприклад, відсутність запису, помилки з'єднання з базою даних). Усі повідомлення про помилки є інформативними та зрозумілими для користувача.

Точність обробки даних забезпечується коректною роботою з текстовими, числовими та датованими полями, а також зображеннями. Для дат використовується стандартний формат ISO, що гарантує правильне відображення незалежно від мови чи часової зони.

Теоретичною основою для реалізації функцій є архітектура "клієнт-сервер" із використанням REST API для обміну даними між фронтендом (Angular) та бекендом (Node.js, Express), а також документно-орієнтованої бази даних MongoDB для зберігання інформації [20]. Для автентифікації та авторизації застосовується механізм JWT (JSON Web Token), що забезпечує безпечний розподіл прав доступу.

Обмеження системи полягають у необхідності мати підключення до Інтернету, використанні сучасного браузера, а також у дотриманні встановлених форматів і розмірів для вхідних даних. Усі критичні операції супроводжуються додатковими перевітками для запобігання помилкам і несанкціонованому доступу.

Часові характеристики програмного комплексу визначаються вимогами до зручності та швидкодії для кінцевого користувача. Оскільки система є інформаційно-пошуковою та орієнтована на взаємодію з користувачем у режимі реального часу, основні операції повинні виконуватися швидко та без затримок, що помітні користувачу.

Відповідь на основні запити користувача, такі як завантаження головної сторінки, перегляд списку постів, пошук або фільтрація публікацій, має здійснюватися не довше 3 секунд. Операції створення, редагування чи видалення постів, а також реєстрація та вхід користувача повинні виконуватися у межах 5 секунд.

					2025.КВР.123.412.10.00.00 ПЗ	Арк.
						19
Змн.	Арк.	№ докум.	Підпис	Дата		

Такий рівень швидкодії забезпечує комфортну роботу з системою та відповідає сучасним вимогам до веб-застосунків.

Жорсткі часові обмеження для окремих операцій, що виконуються рідко (наприклад, адміністративні дії), не встановлюються, однак система повинна залишатися чутливою до дій користувача та оперативно повідомляти про результати виконання запитів.

Вимоги до надійності програмного комплексу полягають у забезпеченні стабільної та безпечної роботи системи за будь-яких умов експлуатації. Архітектура комплексу має ієрархічну структуру, що складається з клієнтської частини (Angular), серверної частини (Node.js, Express) та бази даних (MongoDB). Для контролю збійних ситуацій реалізовано перевірку наявності підключення до бази даних перед виконанням критичних операцій, а також обробку помилок з боку сервера та клієнта.

Всі вхідні дані проходять валідацію як на клієнтській, так і на серверній стороні. Наприклад, для текстових полів перевіряється довжина, для email — відповідність формату, для паролів — мінімальна довжина та складність. Для завантаження зображень контролюється тип файлу та його розмір. Це дозволяє мінімізувати ймовірність помилок, спричинених некоректними діями користувача.

Система забезпечує збереження даних навіть у разі аварійного завершення роботи завдяки використанню надійної бази даних MongoDB, яка підтримує механізми журналювання та відновлення[28]. Усі виняткові ситуації, такі як відсутність запису, помилки з'єднання чи спроба доступу до неіснуючих ресурсів, обробляються із виведенням зрозумілих повідомлень для користувача.

Захист від несанкціонованого доступу реалізовано за допомогою механізму автентифікації та авторизації на основі JWT-токенів, а також розмежування ролей користувачів[26].

					2025.КВР.123.412.10.00.00 ПЗ	Арк.
						20
Змн.	Арк.	№ докум.	Підпис	Дата		

Для виконання критичних дій, таких як видалення постів чи користувачів, обов'язково є перевірка прав доступу та, за необхідності, повторне введення пароля.

Усі повідомлення про помилки містять чіткий опис причини та місця виникнення проблеми, що дозволяє користувачу або адміністратору оперативно реагувати на несправності та забезпечує високу надійність функціонування програмного комплексу.

Умови експлуатації програмного комплексу визначаються його функціональними можливостями, вимогами до апаратного та програмного забезпечення, а також особливостями взаємодії користувача із системою.

Програмний комплекс реалізує такі функціональні можливості: реєстрація та автентифікація користувачів, створення, редагування, видалення та перегляд постів із категоризацією, пошук і фільтрація публікацій, адміністрування користувачів (для адміністратора), підтримка багатомовності інтерфейсу, а також захист даних і контроль доступу до адміністративних функцій. Кожна з цих функцій реалізована через зручний веб-інтерфейс, що дозволяє користувачам взаємодіяти із системою у режимі реального часу.

Взаємодія користувача з програмним комплексом здійснюється через сучасний веб-браузер. Користувачі можуть працювати з системою як з персонального комп'ютера, так і з мобільних пристроїв (планшет, смартфон). Інтерфейс адаптований для різних розмірів екранів, що забезпечує комфортну роботу незалежно від типу пристрою[30].

Програма працює у режимі постійного підключення до серверної частини через мережу Інтернет. Всі основні операції (створення, редагування, перегляд, пошук постів) виконуються у реальному часі. Вихідна інформація (наприклад, список постів, детальна інформація про публікацію, профіль користувача) відображається на екрані користувача. За потреби, користувач може скористатися стандартними можливостями браузера для друку сторінки.

					2025.КВР.123.412.10.00.00 ПЗ	Арк.
						21
Змн.	Арк.	№ докум.	Підпис	Дата		

Для ефективної роботи програмного комплексу рекомендується використовувати комп'ютер із сучасною операційною системою (Windows 10/11, Linux, macOS), процесором із частотою не нижче 2 ГГц, оперативною пам'яттю від 4 ГБ, а також стабільним підключенням до Інтернету. На стороні клієнта необхідний сучасний браузер (Google Chrome, Mozilla Firefox, Microsoft Edge, Safari) з підтримкою стандартів HTML5 та CSS3. Серверна частина функціонує на Node.js версії 18 або вище, база даних — MongoDB версії 6 або вище.

Програмний комплекс є кросплатформеним і не залежить від типу операційної системи користувача. Для інформаційної та програмної сумісності використовується REST API, що дозволяє інтегрувати систему з іншими програмними продуктами за потреби.

1.2.4 Вимоги до програмної документації

Програмна документація є невід'ємною частиною процесу створення та супроводу програмного продукту. Вона забезпечує можливість правильного розуміння, підтримки та вдосконалення програмного забезпечення впродовж усього життєвого циклу.

Відповідно до державного стандарту ДСТУ 4159-2003 «Програмне забезпечення. Документація», попередній склад програмної документації для розробленого веб-сайту включає пояснювальну записку, технічне завдання, програмний код із коментарями, інструкцію користувача, інструкцію з встановлення та конфігурації системи, а також журнал тестування [13].

Пояснювальна записка містить опис мети створення сайту, постановку задачі, технічні та функціональні вимоги. Технічне завдання - це документ, який розповідає, як повинен працювати вебсайт, як він має виглядати та де його слід використовувати. Код програмного забезпечення організований у файлах з чіткою структурою та коментарями.

					2025.КВР.123.412.10.00.00 ПЗ	Арк.
						22
Змн.	Арк.	№ докум.	Підпис	Дата		

Інструкція користувача містить опис інтерфейсу, послідовність дій користувача та можливі сценарії використання сайту. Інструкція з встановлення та конфігурації системи описує необхідні дії для розгортання веб-сайту на сервері. Журнал тестування фіксує проведення тестів, результати, знайдені помилки та методи їх усунення.

Вимоги до тексту програмного коду спрямовані на забезпечення зрозумілості, підтримуваності. Кожен програмний модуль повинен містити початковий блок коментарів, у якому чітко зазначається його назва, призначення, описуються вхідні та вихідні параметри, вказуються можливі обмеження й умови використання, дата створення або останнього редагування, а також автор модуля. Коментарі мають бути лаконічними, зрозумілими та інформативними, розташовуватись безпосередньо над відповідними рядками коду. Заохочується використання самодокументованих імен змінних, функцій і класів.

Структура програмного коду повинна передбачати наявність лише однієї точки входу та однієї точки виходу для кожного модуля, що сприяє підвищенню надійності та передбачуваності роботи системи. Не допускається використання прихованих залежностей між модулями — зв'язки мають бути явними.

Весь програмний код повинен дотримуватися єдиного стилю, відповідно до рекомендацій середовища розробки (наприклад, ESLint для JavaScript/TypeScript)[23]: відступи, розмітка, використання лапок, порожніх рядків тощо мають бути стандартизовані. Код поділяється на окремі логічні блоки, а функції не повинні перевищувати 40 рядків і мають виконувати одну логічну дію.

Зважаючи на веборієнтовану архітектуру, до документації висуваються додаткові вимоги. Вона повинна містити інформацію щодо REST API (опис ендпоінтів, методів, форматів запитів і відповідей), роботи з базою даних (структура колекцій, зв'язки, приклади запитів).

					2025.KBP.123.412.10.00.00 ПЗ	Арк.
						23
Змн.	Арк.	№ докум.	Підпис	Дата		

А також безпекових аспектів (захист паролем, обмеження доступу, авторизація), особливостей реалізації фронтенд-функцій (сценарії анімацій, рендерингу даних, перевірка форм). Документація має бути доступною для розробника, тестувальника та адміністратора, що забезпечить її ефективне використання під час обслуговування, супроводу та модернізації вебсайту[32].

1.2.5 Техніко-економічні показники

У процесі розробки вебсайту було використано сучасне апаратне та програмне забезпечення, що дозволило забезпечити високу продуктивність і якість кінцевого продукту. Робота здійснювалася на персональному комп'ютері Hewlett-Packard моделі HP EliteDesk 800 G1 SFF із процесором Intel(R) Core(TM) i5-4570 CPU @ 3.20GHz (4 ядра, 4 логічних потоки), 16 ГБ оперативної пам'яті DDR3, HDD-накопичувачем на 512 ГБ та інтегрованою графікою Intel(R) HD Graphics 4600. Дисплей мав роздільну здатність 1280x1024 при частоті оновлення 75 Гц, що було достатньо для розробки та тестування користувацького інтерфейсу. Комп'ютер працював під керуванням операційної системи Linux Mint, що гарантує сумісність із більшістю сучасних програмних продуктів для розробки. BIOS Hewlett-Packard L01 v02.21 (UEFI), дата випуску — 17.12.2013. Загальний обсяг фізичної пам'яті становив 15,9 ГБ, доступно фізичної пам'яті — 7,58 ГБ, а віртуальної — 16,4 ГБ.

Для локального тестування та розгортання застосовувалося власне середовище розробки на основі Node.js, що дозволяє запускати серверну частину (Express) та підключати базу даних MongoDB безпосередньо на персональному комп'ютері.

Усі етапи розробки — від створення структури сторінок до тестування функціоналу — виконувалися на цьому ПК, що дозволило уникнути додаткових витрат на оренду віддалених серверів. Для фронтенд-розробки використовувався Angular, який також розгортався локально через

					2025.КВР.123.412.10.00.00 ПЗ	Арк.
						24
Змн.	Арк.	№ докум.	Підпис	Дата		

відповідний CLI. Такий підхід забезпечив повну автономність і контроль над усіма компонентами проєкту під час розробки.

Програмне забезпечення, використане під час розробки, відповідало сучасним стандартам вебіндустрії. Для створення структури та стилізації сторінок застосовувалися HTML5 і CSS3, що забезпечують адаптивність і кросбраузерність інтерфейсу [30]. Динамічна логіка на стороні клієнта реалізовувалася за допомогою JavaScript (ES6+) та Angular версії 9.1.7 [20], що дозволило створити швидкий і зручний застосунок із багатим функціоналом.

Серверна частина була побудована на Node.js версії 14.17.0 із використанням фреймворку Express 4.17.1, що забезпечило гнучкість і масштабованість бекенду. Для зберігання та обробки даних використовувалася документоорієнтована база даних MongoDB 4.4 у поєднанні з бібліотекою Mongoose 5.13.21, яка спрощує роботу з моделями даних і забезпечує їхню цілісність.

Розробка здійснювалася у середовищі Visual Studio Code версії 1.65, яке забезпечує зручний інтерфейс, підтримку розширень і інтеграцію з системою контролю версій Git 2.36. Для тестування інтерфейсу користувача використовувався браузер Google Chrome, що дозволяло перевіряти коректність відображення сторінок і роботу скриптів у реальному часі. Для тестування API-запитів і налагодження взаємодії між фронтендом і бекендом застосовувався Postman версії 9.31. [37]

Використання сучасних безкоштовних фреймворків і програмних засобів дозволило значно знизити витрати на розробку, забезпечити високу якість, надійність і масштабованість вебсайту. Такий підхід сприяв ефективній організації процесу розробки, швидкому впровадженню нових функцій і легкому супроводу проєкту в майбутньому. В результаті створений сайт відповідає сучасним вимогам до швидкодії, безпеки та зручності користування.

					2025.КВР.123.412.10.00.00 ПЗ	Арк.
						25
Змн.	Арк.	№ докум.	Підпис	Дата		

1.2.6 Стадії та етапи розробки

Розробка вебсайту для барберів "Briolin" передбачає проходження стандартних стадій життєвого циклу програмного забезпечення. На кожній стадії виконується відповідний набір робіт, необхідних для досягнення мети проєкту[10].

На першій стадії було проведено аналіз вимог замовника, визначено основні функціональні та нефункціональні вимоги до системи, а також сформульовано технічне завдання [17]. Далі виконувалося проектування архітектури вебсайту, розроблялася структура бази даних, визначалися основні компоненти фронтенду та бекенду, а також їх взаємодія [7].

На етапі реалізації здійснювалася безпосередня розробка програмного коду: спочатку створювалися основні модулі бекенду (серверна логіка, API, робота з базою даних), після чого розроблявся фронтенд (користувацький інтерфейс, компоненти, інтеграція з API). Паралельно проводилося тестування окремих модулів, виправлення помилок та оптимізація роботи системи.

Після завершення основної розробки виконувалося комплексне тестування програмного комплексу, перевірка відповідності функціональних можливостей вимогам технічного завдання, а також підготовка документації для користувачів та адміністраторів. На фінальному етапі здійснювалося розгортання системи на сервері, налаштування середовища експлуатації та навчання персоналу замовника основам роботи з вебсайтом.

Такий підхід дозволив забезпечити поетапний контроль якості, своєчасне виявлення та усунення недоліків, а також ефективне впровадження програмного комплексу у виробничу експлуатацію.

Цей підрозділ дозволяє чітко побачити послідовність і логіку розробки сайту, що є важливим як для контролю виконання, так і для подальшого супроводу.

					2025.КВР.123.412.10.00.00 ПЗ	Арк.
						26
Змн.	Арк.	№ докум.	Підпис	Дата		

1.2.7 Порядок контролю та прийому

Для забезпечення якості розробленого програмного виробу передбачено комплекс заходів з контролю та прийому роботи системи.

Перевірка працездатності програмного комплексу здійснюється у кілька етапів, що охоплюють як ручне, так і автоматизоване тестування[9]. Основна мета контролю — впевнитися у відповідності функціональних можливостей заявленим вимогам, стійкості до помилок користувача та захищеності від несанкціонованого доступу.

Для контролю використовуються контрольні приклади, які охоплюють усі основні сценарії роботи з системою: реєстрацію та авторизацію користувача, створення, редагування, видалення та перегляд постів, пошук і фільтрацію публікацій, а також адміністрування користувачів. Кожен сценарій перевіряється з різними вхідними даними, включаючи граничні та некоректні значення, щоб оцінити стійкість системи до помилок.

Набір тестів включає функціональні тести (перевірка роботи окремих модулів і компонентів), інтеграційні тести (перевірка взаємодії між фронтендом, бекендом і базою даних), а також регресійні тести для виявлення можливих помилок після внесення змін у код. Окремо перевіряється робота системи з різними ролями користувачів (звичайний користувач, адміністратор), багатомовність інтерфейсу, а також захист від несанкціонованого доступу до адміністративних функцій[33].

Автоматизоване тестування здійснюється за допомогою модульних тестів (unit tests) для основних сервісів і компонентів, а також end-to-end тестів (e2e), які імітують дії реального користувача у браузері. Для цього використовуються інструменти Karma, Jasmine та Protractor, що входять до складу Angular-проєкту [20].

Прийомка програмного комплексу проводиться після успішного проходження всіх тестів, підтвердження працездатності на контрольних

					2025.КВР.123.412.10.00.00 ПЗ	Арк.
						27
Змн.	Арк.	№ докум.	Підпис	Дата		

прикладів і перевірки відповідності функціональних можливостей технічному завданню.

Результати тестування фіксуються у журналі тестування, де зазначаються виявлені помилки та способи їх усунення[27]. Остаточне приймання системи здійснюється за участю замовника, який підтверджує відповідність програмного продукту заявленим вимогам та готовність до впровадження.

					2025.КВР.123.412.10.00.00 ПЗ	Арк.
						28
Змн.	Арк.	№ докум.	Підпис	Дата		

2. РОЗРОБКА ТЕХНІЧНОГО ТА РОБОЧОГО ПРОЕКТУ

2.1 Постановка задачі на розробку програмного забезпечення

Веб-сервер — це програмний або апаратний комплекс, основне завдання якого полягає у прийманні, обробці та відповіді на запити клієнтів через мережу. Найчастіше клієнтами виступають веб-браузери, а обмін даними відбувається за протоколами HTTP або HTTPS. Сервери є ключовою ланкою у функціонуванні Інтернету: вони забезпечують доступ до вебсайтів, обробку динамічних запитів, зберігання, обробку та передачу інформації, а також взаємодію з базами даних та іншими сервісами. [9]

Існує багато різновидів веб-серверів, кожен з яких має свої особливості та сфери застосування. Наприклад, Apache HTTP Server — один із найстаріших і найпопулярніших серверів, який відзначається стабільністю та широкими можливостями налаштування. Nginx відомий своєю високою продуктивністю та ефективністю при обробці великої кількості одночасних з'єднань, тому часто використовується як зворотний проксі або балансувальник навантаження. Microsoft IIS (Internet Information Services) — це серверне рішення для платформи Windows, яке інтегрується з іншими продуктами Microsoft. Окрему нішу займає Node.js — це не просто веб-сервер, а повноцінне середовище виконання JavaScript на серверній стороні, що дозволяє створювати високопродуктивні, масштабовані та динамічні вебзастосунки [3].

Node.js вирізняється асинхронною, подієво-орієнтованою архітектурою, завдяки чому ідеально підходить для розробки сучасних вебдодатків, які потребують швидкої обробки великої кількості одночасних запитів. На базі Node.js створено багато популярних фреймворків, зокрема Express, який спрощує розробку серверної логіки, маршрутизації та взаємодії з базами даних. [24]

					2025.КВР.123.412.10.00.00 ПЗ	Арк.
						29
Змн.	Арк.	№ докум.	Підпис	Дата		

Для веб-розробника, який працює над створенням динамічних сайтів, тестуванням нових функцій або інтеграцією сторонніх скриптів, наявність локального сервера на робочому комп'ютері є обов'язковою умовою. Локальний сервер дозволяє безпечно тестувати сайт, налагоджувати його функціонал, перевіряти роботу з базою даних, а також забезпечує можливість імітувати реальні умови роботи майбутнього вебресурсу. Серед популярних рішень для організації локального серверного середовища можна виділити OpenServer, XAMPP, Denwer, які зазвичай використовуються для PHP-проектів. Однак для сучасних JavaScript-проектів, зокрема на основі стеку MEAN (MongoDB, Express, Angular, Node.js), оптимальним вибором є власний сервер на Node.js.

У ході розробки цього проекту було використано локальний сервер, реалізований на Node.js із застосуванням фреймворку Express. Такий підхід дозволив створити повноцінний динамічний вебзастосунок, ефективно тестувати всі його функції, швидко вносити зміни у код, налагоджувати взаємодію між фронтендом і бекендом, а також забезпечити гнучкість і масштабованість архітектури. Завдяки цьому розробник отримав повний контроль над усіма компонентами проекту та міг оперативно реагувати на будь-які зміни чи потреби у процесі створення сайту.

2.1.1 Огляд програмних засобів для розробки веб-сайту

Для створення сучасного, динамічного та масштабованого вебсайту було використано стек технологій MEAN, який включає MongoDB, Express, Angular і Node.js. Кожен із цих компонентів виконує свою важливу роль у розробці, а також доповнюється низкою допоміжних програмних засобів.

Node.js — це середовище виконання JavaScript на серверній стороні, яке дозволяє створювати високопродуктивні вебсервери та обробляти численні одночасні запити завдяки асинхронній архітектурі. У цьому проекті

					2025.КВР.123.412.10.00.00 ПЗ	Арк.
						30
Змн.	Арк.	№ докум.	Підпис	Дата		

використовувалася версія Node.js 14.17.0, яка забезпечує стабільну роботу та підтримку сучасних стандартів JavaScript [31].

Express — це мінімалістичний і гнучкий фреймворк для Node.js, який значно спрощує розробку серверної частини застосунку. Express дозволяє легко організовувати маршрутизацію, обробку запитів, роботу з middleware, а також інтеграцію з базами даних. У проєкті застосовувалася версія Express 4.17.1 [24].

MongoDB — документоорієнтована система управління базами даних, яка зберігає інформацію у вигляді гнучких JSON-подібних документів. Це дозволяє швидко змінювати структуру даних без необхідності складних міграцій. Для взаємодії з MongoDB використовувалася бібліотека Mongoose версії 5.13.21, яка забезпечує зручну роботу з моделями даних, валідацію та зв'язки між колекціями [29].

Angular — сучасний фреймворк для розробки односторінкових вебзастосунків (SPA) на стороні клієнта. Angular дозволяє створювати масштабовані, структуровані та інтерактивні інтерфейси користувача, використовуючи TypeScript, шаблони, двосторонню прив'язку даних і модульну архітектуру. У цьому проєкті використовувалася версія Angular 9.1.7 [36].

Visual Studio Code — інтегроване середовище розробки (IDE), яке забезпечує зручний інтерфейс для написання коду, підтримку розширень, інтеграцію з системою контролю версій Git, а також інструменти для налагодження та тестування застосунку. У роботі використовувалася версія Visual Studio Code 1.65 [37].

Postman — інструмент для тестування API-запитів, який дозволяє перевіряти роботу серверної частини, надсилати різні типи HTTP-запитів, аналізувати відповіді та налагоджувати взаємодію між фронтендом і бекендом. У проєкті застосовувалася версія Postman 9.31 [27].

					2025.КВР.123.412.10.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		31

Google Chrome — сучасний веббраузер, який використовувався для тестування інтерфейсу користувача, перевірки адаптивності, кросбраузерності та продуктивності сайту [30].

Завдяки використанню цих програмних засобів вдалося забезпечити ефективну організацію процесу розробки, швидке впровадження нових функцій, зручність тестування та підтримку сучасних стандартів вебіндустрії.

2.2 Опис та обґрунтування вибору методу організації вхідних та вихідних даних

У даному програмному комплексі структура та організація вхідних і вихідних даних визначаються вимогами до функціональності системи, а також необхідністю забезпечення швидкого доступу, пошуку та обробки інформації. Всі основні дані зберігаються у документно-орієнтованій базі даних MongoDB, що дозволяє гнучко організувати структуру записів та легко масштабувати систему [29].

Вхідні дані включають інформацію, яку користувачі вводять через веб-інтерфейс: дані для реєстрації (ім'я, логін, email, пароль), дані для входу, інформацію для створення, редагування або видалення постів (заголовок, текст, категорія, фото), а також запити на пошук чи фільтрацію, зміну профілю та вибір мови інтерфейсу. Всі ці дані мають чітко визначену структуру, що відповідає вимогам до валідації та безпеки. [30]

Вихідні дані формуються у вигляді відповідей на запити користувача: списки постів, детальна інформація про окремі публікації, повідомлення про успішне виконання дій або помилки, а також дані профілю користувача. Для адміністратора додатково формуються списки користувачів та їхніх публікацій [35].

					2025.КВР.123.412.10.00.00 ПЗ	Арк.
						32
Змн.	Арк.	№ докум.	Підпис	Дата		

Ієрархічна структура даних виглядає наступним чином:

- Користувач (User): містить унікальний ідентифікатор, ім'я, логін, email, роль, дату реєстрації, список ідентифікаторів створених постів.
- Пост (Post): містить унікальний ідентифікатор, заголовок, текст, категорію, дату створення, ідентифікатор автора, посилання на фото.
- Категорія (Category): містить унікальний ідентифікатор та назву категорії.
- Повідомлення (Message): містить текст повідомлення, тип (інформаційне, помилка, підтвердження), дату та ідентифікатор користувача.

Для організації даних використовується принцип нормалізації, що дозволяє уникнути дублювання інформації та забезпечує цілісність даних. Основним ключем упорядкування для користувачів є унікальний логін або email, для постів — дата створення (для хронологічного відображення) та категорія (для фільтрації). Пошук постів здійснюється за ключовими словами у заголовку та тексті, а також за категорією [].

Такий підхід до організації вхідних і вихідних даних забезпечує ефективну роботу системи, швидкий пошук і фільтрацію інформації, а також простоту масштабування та супроводу програмного комплексу.

2.3 Розробка алгоритму

2.3.1 Зовнішнє проектування програми

Сайт побудований на основі Angular, із серверною частиною на Node.js (Express) та базою даних MongoDB. Основні сторінки та функціональні блоки організовані таким чином, щоб забезпечити зручну навігацію для користувачів і адміністраторів.

Структурну схему сайту для барберів “Briolin” наведено на рисунку 2.1.

					2025.КВР.123.412.10.00.00 ПЗ	Арк.
						33
Змн.	Арк.	№ докум.	Підпис	Дата		

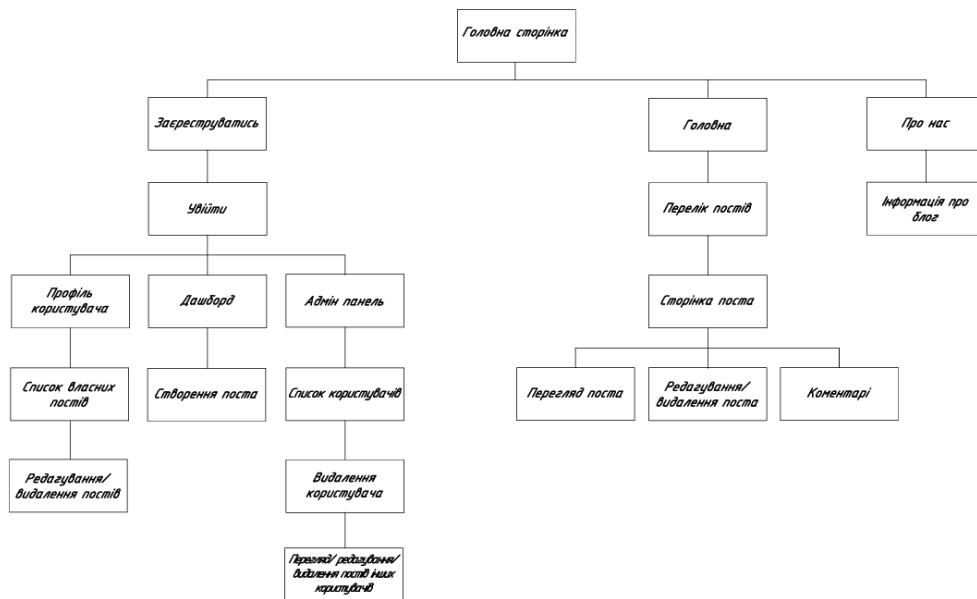


Рисунок 2.1 - Структурна схема сайту

Кожна функція інтерфейсу має чітко визначене призначення. Наприклад, головна сторінка містить список постів із можливістю пошуку та фільтрації за категоріями. Користувач може перейти до перегляду окремого поста. На рисунку 2.2 відображається вигляд головної сторінки для незареєстрованих і неавторизованих користувачів. Рисунок 2.3 це продовження головної сторінки.

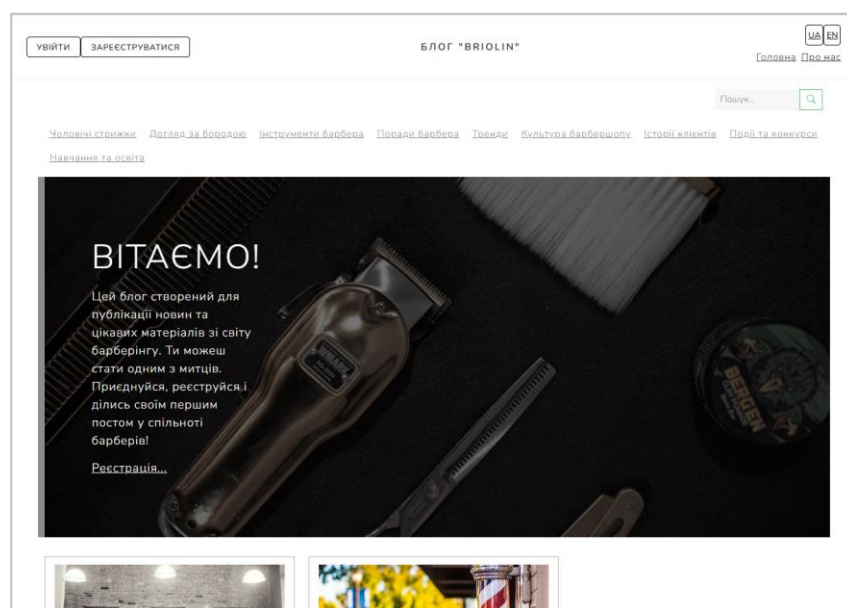


Рисунок 2.2 - Головна сторінка вебсайту

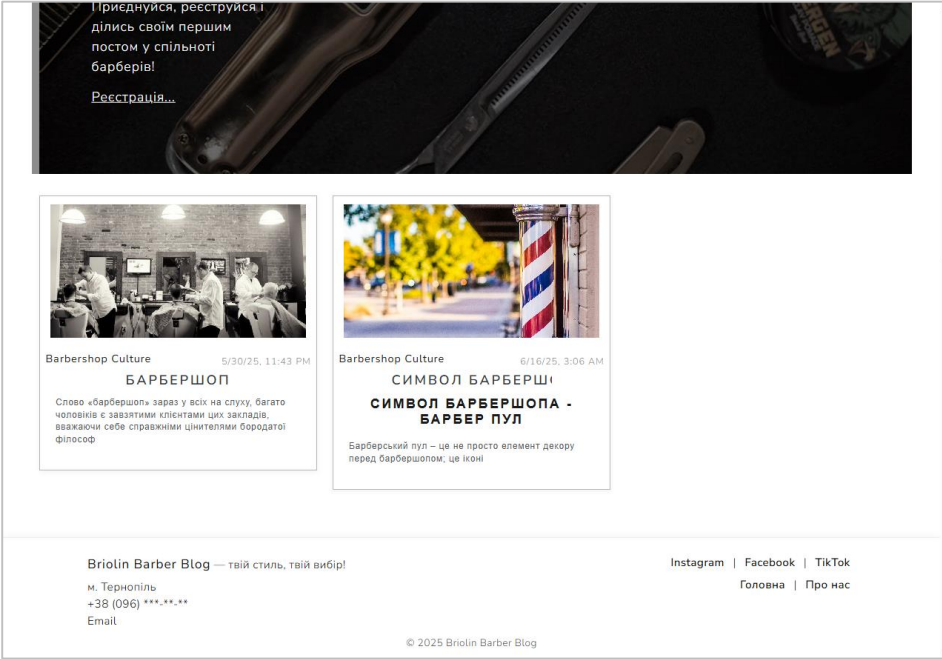


Рисунок 2.3 - Продовження головної сторінки вебсайту

Для можливості перегляду повноцінного сайту з усіма його можливостями потрібно перейти на сторінку реєстрації зображеної на рисунку 2.4. Є два посилання переходу на цю сторінку: через кнопку у лівому верхньому куті головної сторінки або посиланні вказаному на фото головної сторінки.

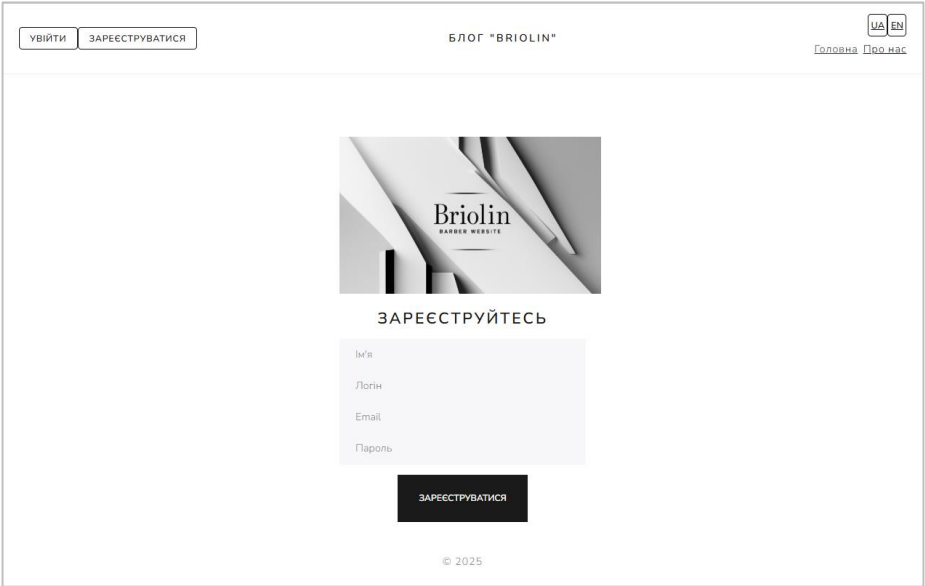


Рисунок 2.4 - Сторінка реєстрації

Наступним кроком є сторінка входу, де користувач вносить логін і пароль, які вводив при реєстрації. На рисунку 2.5 наведено зовнішній вигляд сторінки входу на сайт.

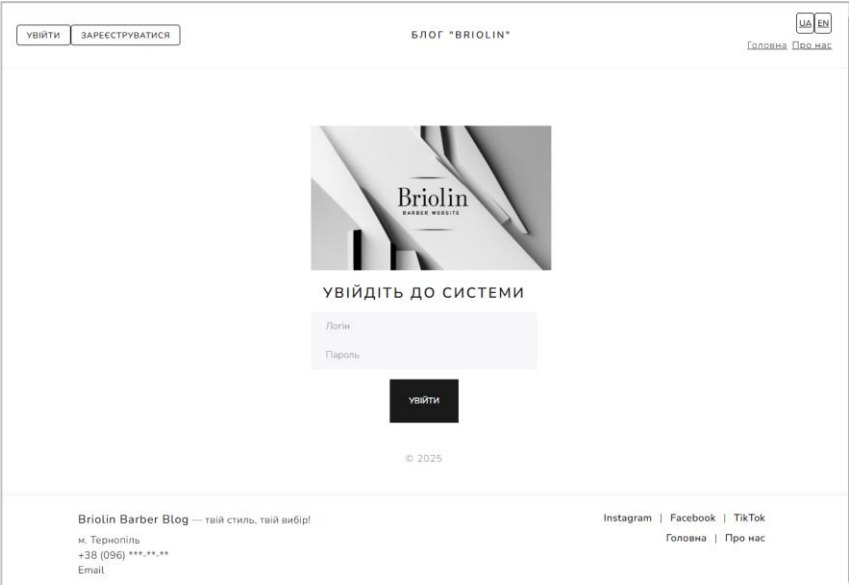
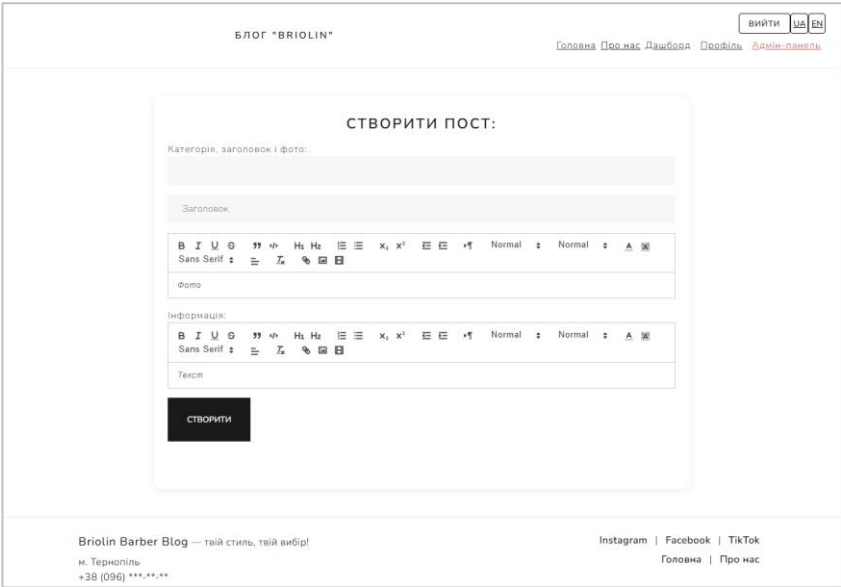


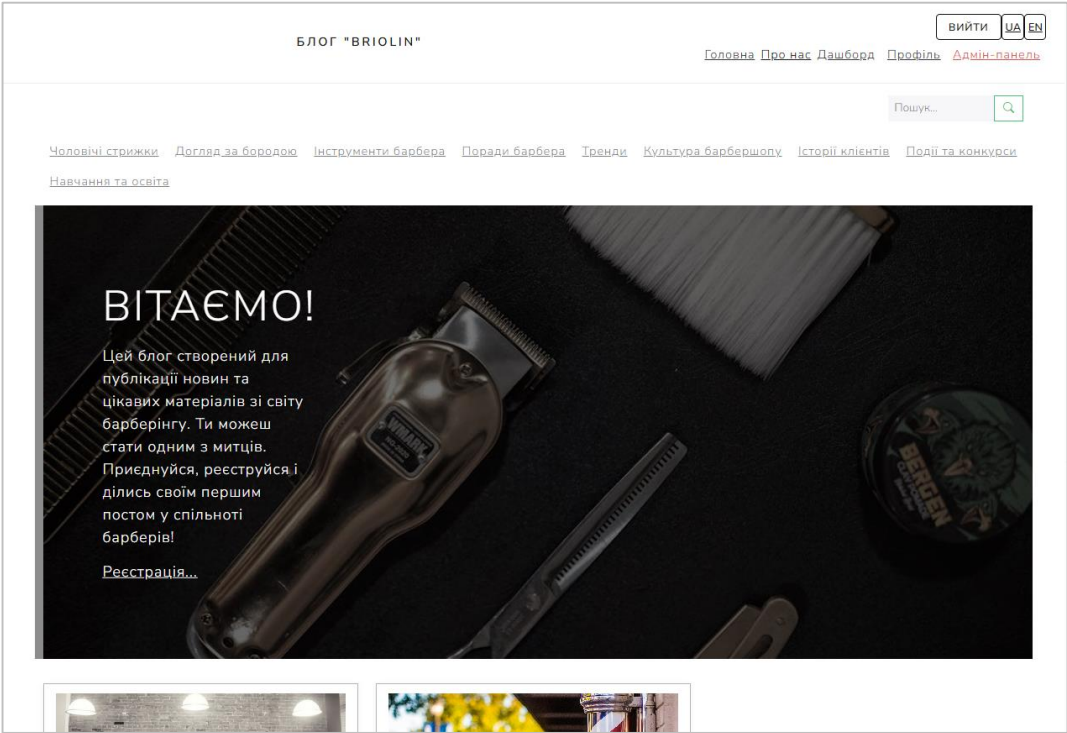
Рисунок 2.5 - Сторінка входу на сайт

Після того як користувач увійшов на вебсайт його одразу переносить на сторінку дашборд для того, щоб він створив свій перший пост. На рисунку 2.6 наведено сторінку створення постів.



Рисунку 2.6 - Сторінка створення постів

Ось і можна повернутись до того, як виглядає головна сторінка для авторизованого користувача. Це наведено нам на рисунку 2.7. Зміни тільки відбуваються у заголовку сторінки. Тепер користувачеві доступні кнопки переходу на сторінку створення посту та профіль користувача. Адмін панель буде доступна тільки адміністраторові сайту.



Після створення першого посту користувач може повернутись на головну сторінку і відкрити для перегляду будь-який з наявних постів. Сторінка поста відображає повну інформацію про вибраний пост, коментарі до нього. Автор поста або адміністратор можуть редагувати чи видаляти пост. На рисунку 2.8 надано зовнішній вигляд цієї сторінки.

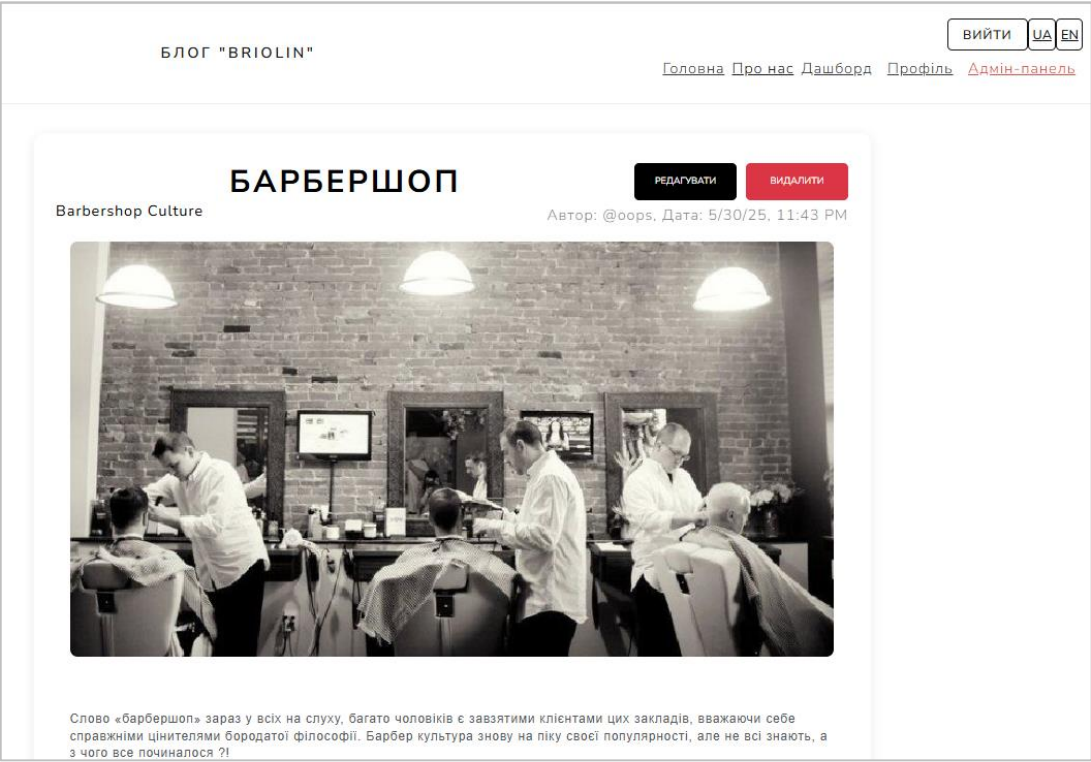


Рисунок 2.8 - Зовнішній вигляд сторінки посту

На рисунку 2.9 продовження сторінки поста.

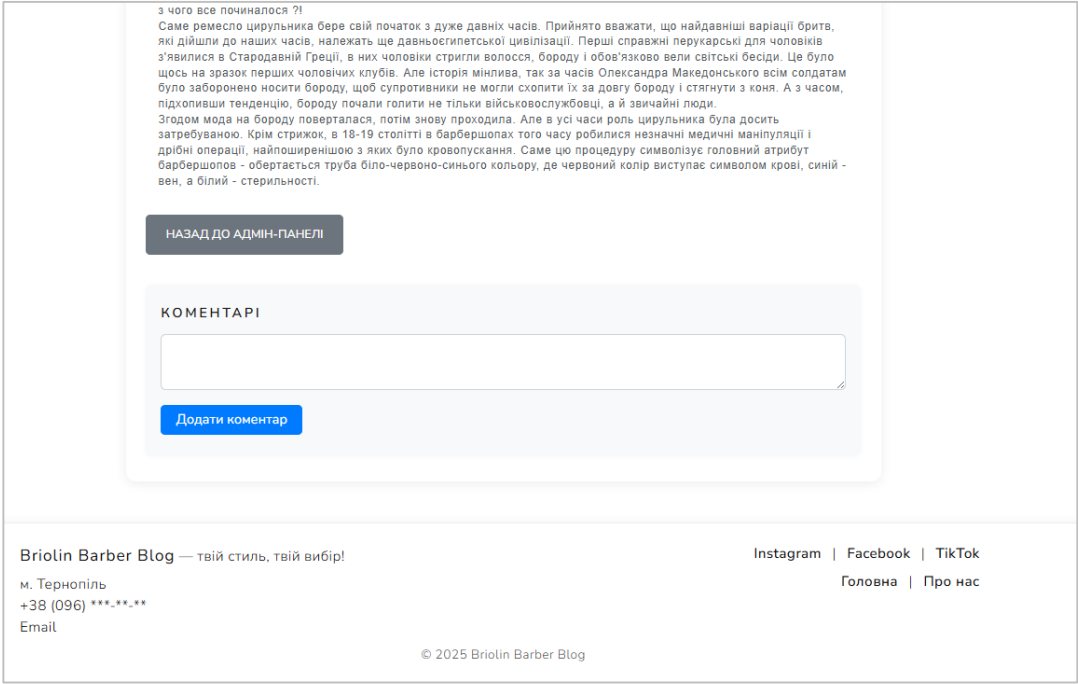


Рисунок 2.9 - Продовження сторінки поста

Особистий кабінет дозволяє користувачу переглядати, редагувати та видаляти власні пости (див. рис. 2.10).

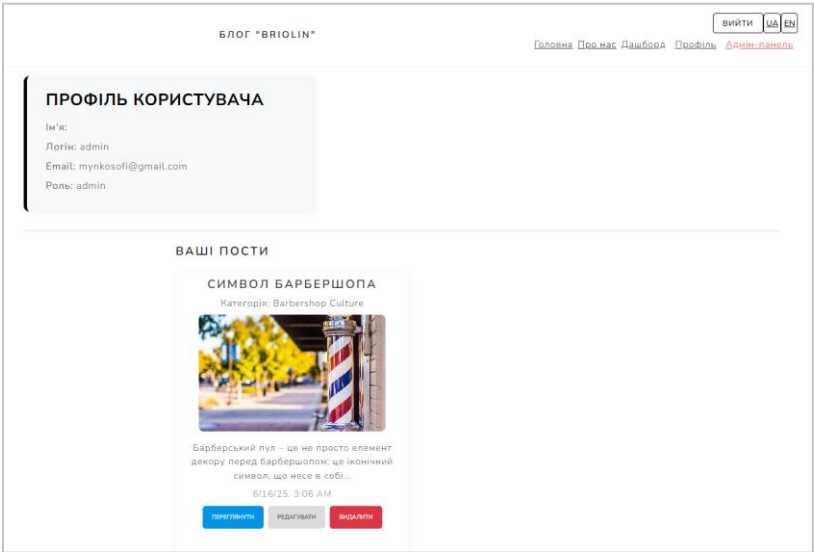


Рисунок 2.10 - Особистий кабінет користувача

Адмін-панель надає адміністратору розширені можливості для керування користувачами та контентом. На рисунку 2.11 наведено зовнішній вигляд сторінки адміністратора.

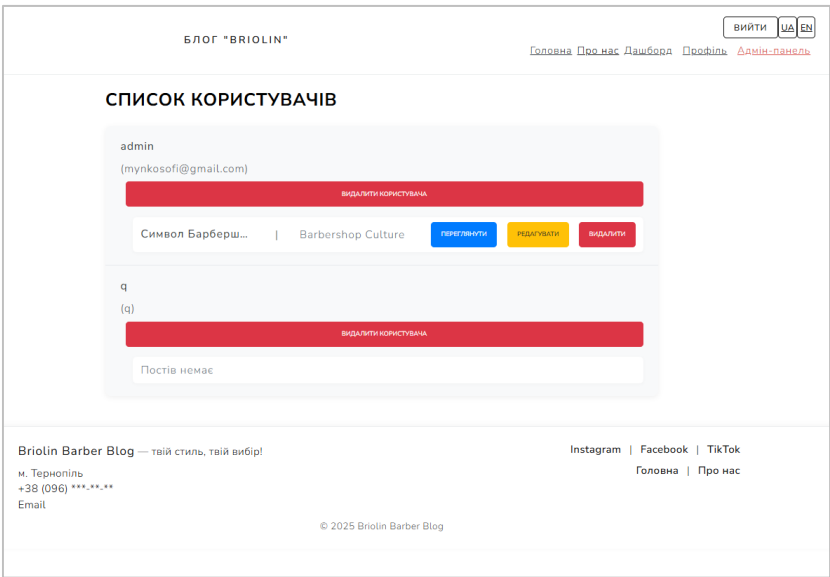


Рисунок 2.11 - Зовнішній вигляд сторінки адміністратора

Останньою сторінкою сайту є інформативна сторінка про автора блогу. На рисунку 2.12 наведено сторінку «Про нас».

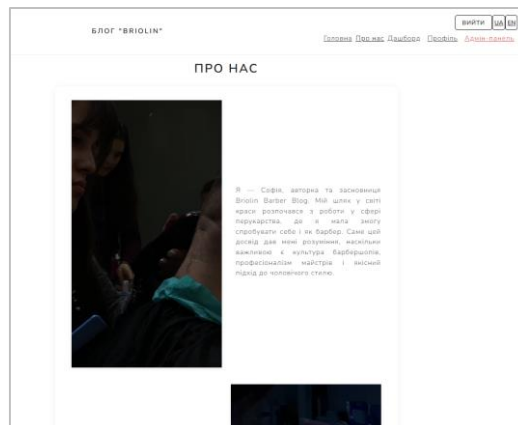


Рисунок 2.12 - Сторінка «Про нас»

Таблиця 2.1 – Ієрархічна структура зовнішніх функцій

Розділ	Функціонал
Головна сторінка	Відображення списку постів, пошук постів, фільтрація за категоріями, навігація до сторінки поста або реєстрації
Сторінка поста	Перегляд повної інформації про пост (заголовок, текст, фото, категорія, дата), коментарі, редагування/видалення поста (для автора/адміна)
Реєстрація	Заповнення форми реєстрації, створення нового облікового запису
Вхід	Введення логіну та паролю, авторизація користувача
Дашборд	Створення нового поста, перегляд власних постів
Профіль користувача	Перегляд, редагування та видалення власних постів
Адмін-панель	Перегляд списку користувачів, зміна ролей, видалення користувачів, перегляд/редагування/видалення постів інших користувачів
Сторінка «Про нас»	Перегляд інформації про блог, відео, текст про автора

2.3.2 Проектування логіки програми

Логіка розв'язання задачі у програмному комплексі "Briolin" організована за принципом ієрархічної деталізації функцій (метод "зверху-вниз"). На верхньому рівні виділяються основні групи функцій: керування користувачами, керування постами, пошук і фільтрація, адміністрування та локалізація. Кожна з цих груп деталізується на підфункції, які відповідають окремим сценаріям взаємодії користувача із системою. [9]

Ієрархічна схема зовнішніх функцій (верхній рівень)

- Керування користувачами (реєстрація, автентифікація, редагування профілю)
- Керування постами (створення, редагування, видалення, перегляд)
- Пошук і фільтрація (за ключовим словом, за категорією)
- Адміністрування (керування користувачами, модерація постів, перегляд статистики)
- Локалізація (зміна мови інтерфейсу)

Опис функціональних специфікацій за рівнями

Верхній рівень:

Кожна група функцій приймає на вхід директиви користувача (наприклад, натискання кнопки "Створити пост", введення даних у форму реєстрації, вибір категорії для фільтрації). Система реагує на ці дії відповідно до сценарію: перевіряє коректність введених даних, взаємодіє з базою даних, формує відповідь (наприклад, повідомлення про успіх чи помилку, оновлений список постів) [30].

Деталізація (нижчий рівень):

Для кожної функції описується:

- Ввід: які саме дані вводить користувач (наприклад, логін і пароль для входу, текст і категорія для створення поста).

					2025.КВР.123.412.10.00.00 ПЗ	Арк.
						41
Змн.	Арк.	№ докум.	Підпис	Дата		

- Вивід: яку інформацію отримує користувач у відповідь (наприклад, повідомлення про успішну реєстрацію, список постів, повідомлення про помилку).

- Перетворення: які зміни відбуваються у стані системи (наприклад, створення нового запису у базі даних, оновлення профілю користувача, фільтрація списку постів).

Обробка непередбачених даних:

Усі форми та поля проходять валідацію. У разі введення некоректних або неповних даних система виводить інформативне повідомлення про помилку, не допускаючи збереження некоректної інформації [33]. При спробі доступу до заборонених функцій (наприклад, адміністрування без відповідних прав) користувач отримує повідомлення про відмову в доступі.

Структурна модель програмного комплексу зображена на рисунку 2.13.

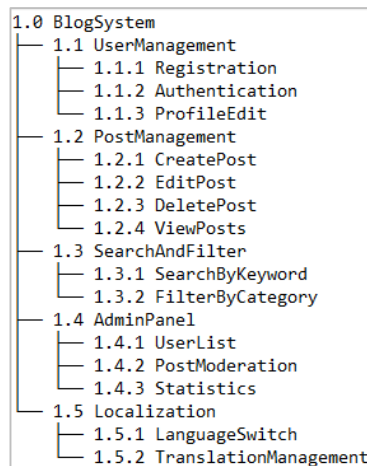


Рисунок 2.13 - Структурна модель програмного комплексу

Програмний комплекс складається з п'яти основних підсистем, кожна з яких відповідає за окремий напрям функціональності.

Підсистема ****UserManagement**** відповідає за реєстрацію нових користувачів, автентифікацію (вхід у систему) та редагування профілю.

Підсистема ****PostManagement**** забезпечує створення, редагування, видалення та перегляд публікацій (постів) користувачами.

					2025.КВР.123.412.10.00.00 ПЗ	Арк.
						42
Змн.	Арк.	№ докум.	Підпис	Дата		

Підсистема ****SearchAndFilter**** реалізує пошук постів за ключовими словами та фільтрацію за категоріями, що дозволяє швидко знаходити потрібну інформацію.

Підсистема ****AdminPanel**** надає адміністраторам можливість переглядати список користувачів, модерувати пости та переглядати статистику роботи системи.

Підсистема ****Localization**** відповідає за зміну мови інтерфейсу та керування перекладами, що забезпечує багатомовність вебсайту.

Така структура дозволяє ефективно організувати розробку, тестування та подальший супровід програмного комплексу, а також забезпечує гнучкість і масштабованість системи.

2.4 Визначення інформаційних зв'язків програмних компонентів

Інформаційні зв'язки між функціями програмного комплексу "BlogSystem" визначаються потоками управління та обміну даними між основними підсистемами: керування користувачами, керування постами, пошук і фільтрація, адміністрування та локалізація. Зв'язки реалізуються через виклики REST API, обробку подій у фронтенді та взаємодію з базою даних [31].

Структурна модель інформаційних зв'язків (текстовий вигляд) зображена на рисунку 2.14.

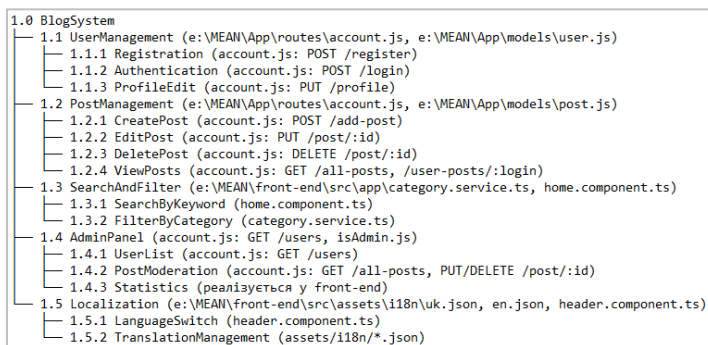


Рисунок 2.14 - Структурна модель інформаційних зв'язків

					2025.КВР.123.412.10.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		43

Таблиця 2.2 - Інформаційні зв'язки

№ функції	Найменування функції	Модуль/файл, що реалізує
1.	Реєстрація користувача	account.js, user.js
2.	Авторизація користувача	account.js, user.js
3.	Редагування профілю	account.js, user.js
4.	Створення поста	account.js, post.js
5.	Редагування поста	account.js, post.js
6.	Видалення поста	account.js, post.js
7.	Перегляд постів	account.js, post.js
8.	Пошук за ключовим словом	home.component.ts
9.	Фільтрація за категорією	category.service.ts
10.	Перегляд списку користувачів	account.js, isAdmin.js
11.	Модерація постів	account.js, isAdmin.js
12.	Перегляд статистики	front-end (dashboard)
13.	Перемикання мови	header.component.ts
14.	Керування перекладами	assets/i18n/*.json

Для візуалізації зв'язків між компонентами може бути використана UML-діаграма класів або компонентів, де відображаються основні сутності (User, Post, Category), сервіси (AuthService, CategoryService), а також їх взаємодія через API.

2.5 Написання скрипта

У межах проєкту реалізовано окремий скрипт (плагін) — компонент профілю користувача, який відповідає за управління особистим кабінетом. Даний компонент написаний мовою TypeScript з використанням Angular і є невід'ємною частиною клієнтської частини сайту. [36]

Його основне призначення — забезпечити користувачу можливість керувати власними публікаціями у зручному інтерфейсі. Після завантаження сторінки профілю скрипт отримує інформацію про поточного користувача з локального сховища браузера, що дозволяє ідентифікувати, хто саме увійшов у систему. Далі, використовуючи Angular-сервіс авторизації, компонент надсилає запит до серверної частини застосунку для отримання списку всіх постів, створених цим користувачем. Всі отримані пости відображаються у профілі, і для кожного з них доступні дії: перегляд, редагування та видалення.

Перегляд поста здійснюється шляхом перенаправлення користувача на окрему сторінку з детальною інформацією про вибраний пост. Редагування також виконується через перенаправлення на відповідну сторінку редагування, де користувач може змінити вміст поста. Видалення поста супроводжується підтвердженням дії — користувачу пропонується підтвердити своє рішення, після чого надсилається запит на сервер для видалення поста з бази даних. Після успішного видалення відповідний пост одразу зникає зі списку без необхідності оновлювати сторінку. [31]

Весь цей функціонал реалізовано за допомогою сучасних підходів Angular: використання сервісів для взаємодії з API, реактивного оновлення даних у шаблоні, а також маршрутизації для переходу між сторінками. Такий

					2025.КВР.123.412.10.00.00 ПЗ	Арк.
						45
Змн.	Арк.	№ докум.	Підпис	Дата		

підхід дозволяє зробити роботу з особистим кабінетом максимально швидкою, зручною та безпечною для користувача, а також забезпечує ефективну інтеграцію клієнтської та серверної частин застосунку. [32]

На рисунку 2.15 наведено код скрипта «Профіль». А файли з стилями та зовнішнім виглядом винесено в додатки.

```

TS profile.component.ts X
front-end > src > app > profile > TS profile.component.ts > ProfileComponent > editPost
1  import { Component, OnInit } from '@angular/core';
2  import { AuthService } from '../auth.service';
3  import { Router } from '@angular/router';
4
5  @Component({
6    selector: 'app-profile',
7    templateUrl: './profile.component.html',
8    styleUrls: ['./profile.component.scss']
9  })
10 export class ProfileComponent implements OnInit {
11   user: any;
12   userPosts: any[] = [];
13
14   constructor(private authService: AuthService, private router: Router) {}
15
16   ngOnInit(): void {
17     this.user = JSON.parse(localStorage.getItem('user') || 'null');
18     if (this.user?.login) {
19       this.authService.getUserPosts(this.user.login).subscribe(posts => {
20         this.userPosts = posts;
21       });
22     }
23   }
24   viewPost(postId: string) {
25     this.router.navigate(['/post', postId]);
26   }
27
28   editPost(postId: string) {
29     this.router.navigate(['/edit-post', postId]);
30   }
31
32   deletePost(postId: string) {
33     if (confirm('Are you sure you want to delete this post?')) {
34       this.authService.deletePost(postId).subscribe(() => {
35         this.userPosts = this.userPosts.filter(post => post._id !== postId);
36       });
37     }
38   }
39   getShortText(html: string, maxLength = 100): string {
40     const div = document.createElement('div');
41     div.innerHTML = html || '';
42     const text = div.textContent || div.innerText || '';
43     return text.length > maxLength ? text.slice(0, maxLength) + '...' : text;
44   }
45 }

```

Рисунку 2.15 - Код скрипта «Профіль».

					2025.КВР.123.412.10.00.00 ПЗ	Арк.
						46
Змн.	Арк.	№ докум.	Підпис	Дата		

2.5 Тестування сайту

Тестування сайту — це процес перевірки працездатності, коректності відображення та функціональності вебресурсу на різних пристроях, операційних системах і браузерях. Мета тестування полягає у виявленні та усуненні помилок, забезпеченні зручності користування, адаптивності дизайну та стабільної роботи всіх функцій незалежно від середовища використання.

Тестування сайту “Briolin Barber Blog” було зроблено за допомогою Google Chrome, Microsoft Edge, Opera. На рисунках нижче наведено приклади відображення сайту у різних браузерах.

На рисунку 2.16 наведено вигляд сайту на моніторі Google Chrome.

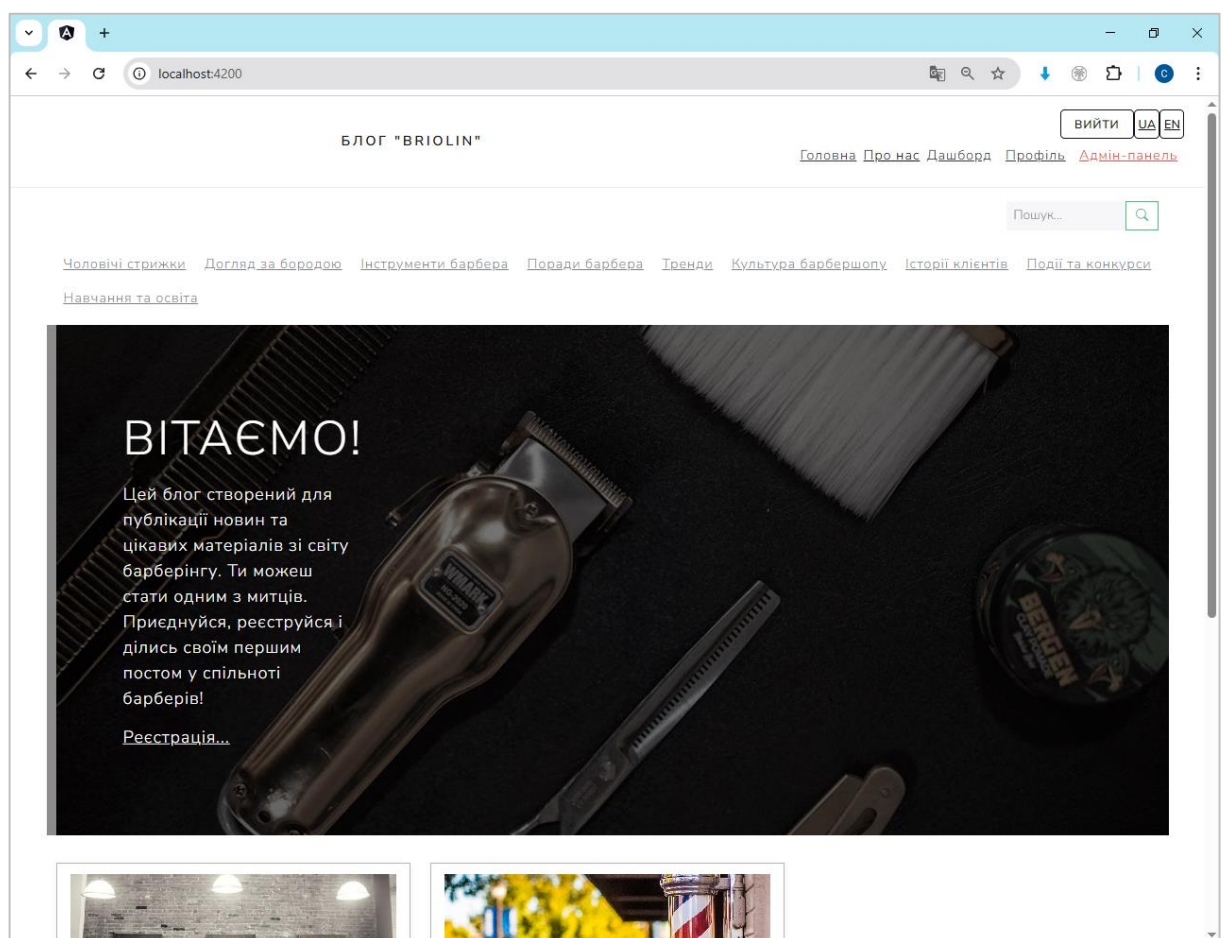


Рисунок 2.16 – Вигляд сайту на моніторі Google Chrome

					2025.КВР.123.412.10.00.00 ПЗ	Арк.
						47
Змн.	Арк.	№ докум.	Підпис	Дата		

На рисунку 2.17 відображається вигляд сайту на моніторі Microsoft Edge.

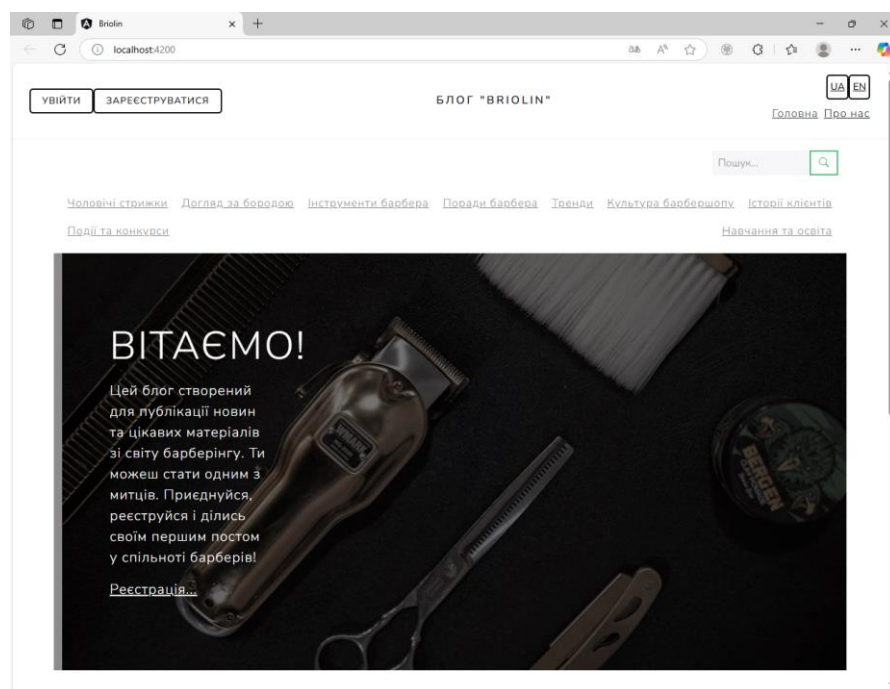


Рисунок 2.17 – Вигляд сайту на моніторі Microsoft Edge

На рисунку 2.18 – Вигляд сайту на моніторі Opera

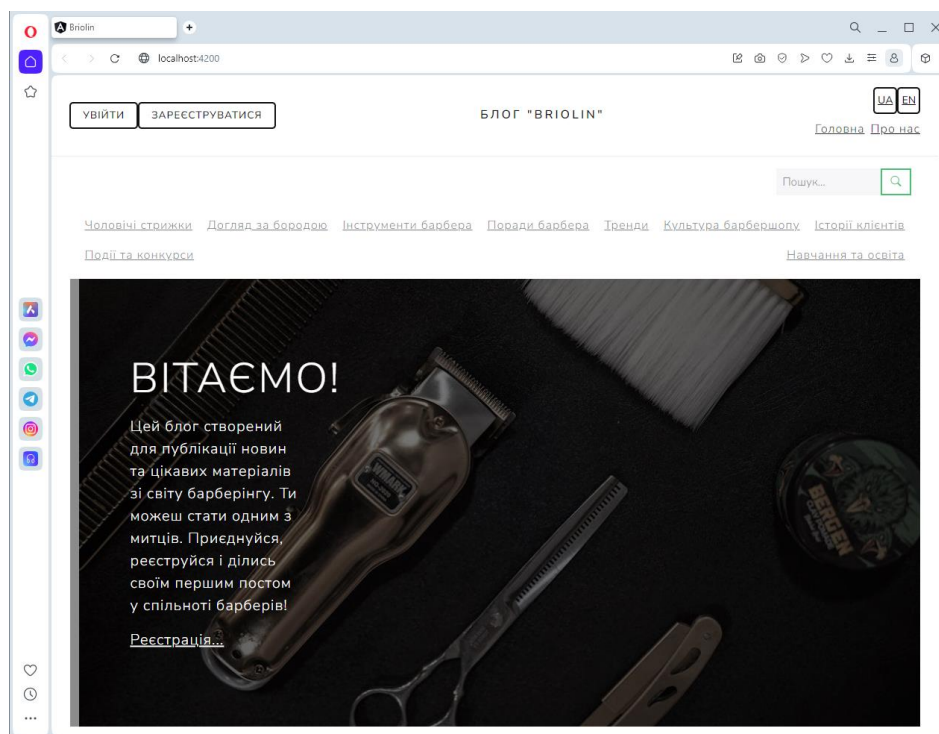


Рисунок 2.18 – Вигляд сайту на моніторі Opera

Змн.	Арк.	№ докум.	Підпис	Дата

2025.КВР.123.412.10.00.00 ПЗ

Арк.

48

На рисунку 2.19 зображено вигляд сайту на моніторі мобільного телефону Iphone 12 pro.

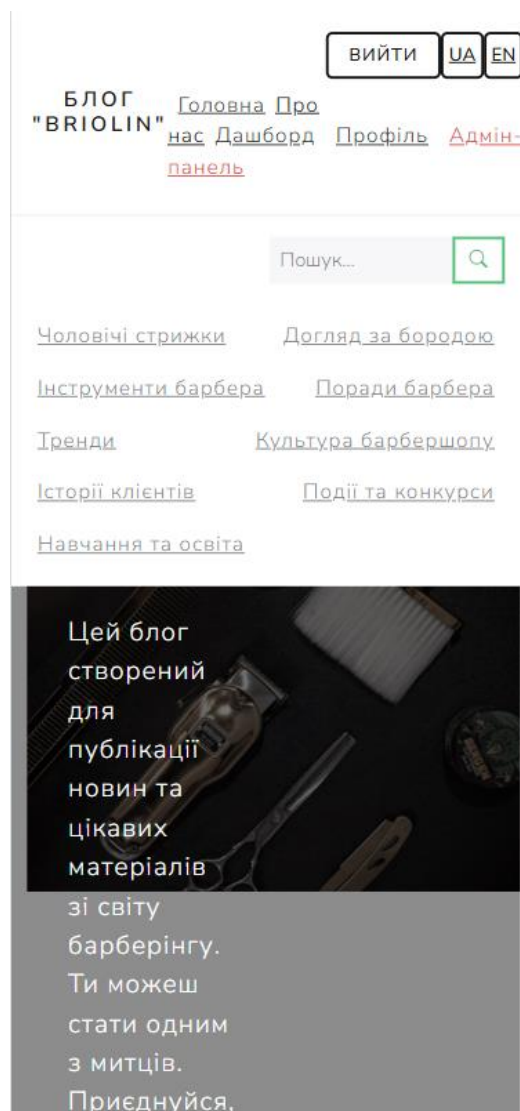


Рисунок 2.19 – Вигляд сайту на мобільному телефоні Iphone 12 pro в Google Chrome

Проведене тестування підтвердило, що сайт коректно відображається та функціонує на всіх основних пристроях і браузерах, забезпечуючи зручність користування для різних категорій користувачів. Усі виявлені незначні недоліки були усунуті в процесі розробки.

					2025.КВР.123.412.10.00.00 ПЗ	Арк.
						49
Змн.	Арк.	№ докум.	Підпис	Дата		

3. СПЕЦІАЛЬНИЙ РОЗДІЛ

3.1 Технологія MEAN

Сайт розроблявся із використанням сучасного технологічного стеку MEAN, який є одним із найпопулярніших рішень для створення повноцінних веб-застосунків.

Назва MEAN є аббревіатурою, що складається з перших літер назв чотирьох основних технологій: MongoDB, Express, Angular і Node.js. Кожен компонент цього стеку виконує свою унікальну функцію, забезпечуючи цілісність, гнучкість і масштабованість програмного комплексу [1].

MongoDB використовується як основна база даних. Це документно-орієнтована система управління базами даних, яка дозволяє зберігати інформацію у вигляді колекцій документів формату JSON. Такий підхід забезпечує гнучкість структури даних, швидкий доступ до інформації та простоту масштабування. У MongoDB зберігаються всі основні дані сайту: облікові записи користувачів, пости, категорії, коментарі та інші сутності, необхідні для роботи блогу [28].

Express.js виконує роль серверного фреймворку для Node.js і відповідає за організацію маршрутизації, обробку HTTP-запитів, реалізацію REST API, а також за підключення до бази даних. Саме Express забезпечує логіку взаємодії між клієнтською частиною сайту та сервером, обробляє запити на створення, редагування, видалення та отримання даних, а також відповідає за автентифікацію та авторизацію користувачів [24].

Angular використовується для розробки клієнтської частини (frontend) сайту. Це сучасний фреймворк, який дозволяє створювати динамічні, багатофункціональні та адаптивні інтерфейси користувача. Angular забезпечує двосторонню прив'язку даних, що дозволяє миттєво відображати зміни на сторінці без перезавантаження.

					2025.КВР.123.412.10.00.00 ПЗ	Арк.
						50
Змн.	Арк.	№ докум.	Підпис	Дата		

Крім того, Angular підтримує модульну структуру проекту, що спрощує розробку, тестування та подальший супровід коду. Для написання коду в Angular використовується мова TypeScript, яка є надбудовою над JavaScript і дозволяє створювати більш безпечний, структурований і зрозумілий код[20].

Node.js є серверною платформою, на якій працює вся логіка бекенду. Завдяки асинхронній моделі обробки подій Node.js забезпечує високу продуктивність і здатність обробляти велику кількість одночасних запитів. Node.js дозволяє запускати JavaScript-код на сервері, що забезпечує єдність технологій на всіх рівнях розробки та спрощує інтеграцію між різними компонентами системи [31].

Весь програмний код серверної частини, а також взаємодія з базою даних реалізовані мовою JavaScript, що дозволяє використовувати одну мову програмування як для клієнтської, так і для серверної частини проекту [2]. Це значно спрощує розробку, тестування та підтримку програмного комплексу. Для фронтенду використовується Angular із застосуванням TypeScript, що підвищує надійність і підтримуваність коду [36].

Завдяки використанню стеку MEAN сайт отримав сучасну архітектуру, що дозволяє легко масштабувати проект, додавати нові функції, забезпечувати високу швидкодію та зручність для кінцевого користувача. Такий підхід також спрощує розгортання системи на різних платформах і забезпечує кросплатформеність програмного комплексу [1].

3.2 Процес встановлення програмного комплексу на ПК

Встановлення програмного комплексу на персональний комп'ютер розпочалося з підготовки необхідного програмного середовища. На першому етапі я завантажила з офіційного сайту дистрибутив Node.js (див. рис. 3.1).

					2025.КВР.123.412.10.00.00 ПЗ	Арк.
						51
Змн.	Арк.	№ докум.	Підпис	Дата		

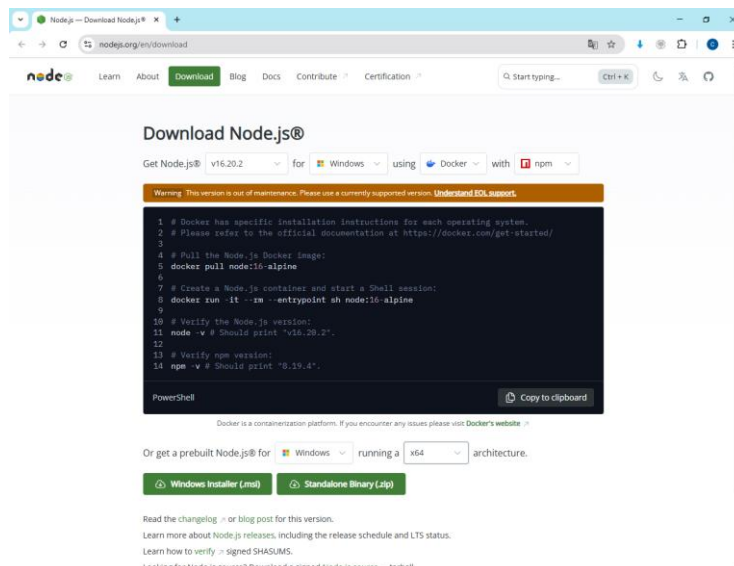


Рисунок 3.1 - Офіційний сайт Node.js

Після чого виконала його інсталяцію, дотримуючись стандартних кроків майстра встановлення (див. рис. 3.2).

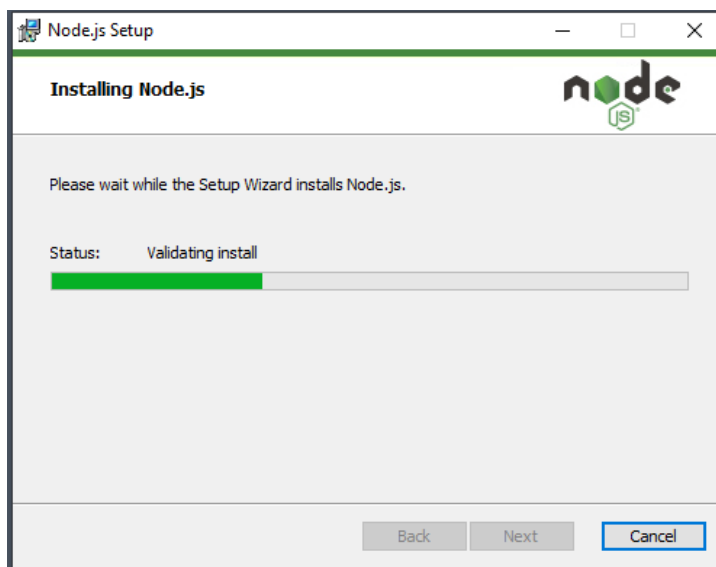


Рисунок 3.2 - Інсталяція Node.js

Після завершення встановлення Node.js перевірила коректність інсталяції за допомогою команд `node -v` та `npm -v` у командному рядку, що дозволило впевнитися у доступності Node.js та менеджера пакетів npm.

Змн.	Арк.	№ докум.	Підпис	Дата

2025.КВР.123.412.10.00.00 ПЗ

Арк.

52

Наступним кроком було встановлення MongoDB Community Server. Я завантажила інсталяційний файл з офіційного сайту MongoDB, запустила його та виконала встановлення за типовими налаштуваннями (див. рис. 3.2). Після завершення інсталяції перевірила запуск MongoDB через відповідний сервіс Windows та переконатися, що база даних працює коректно.

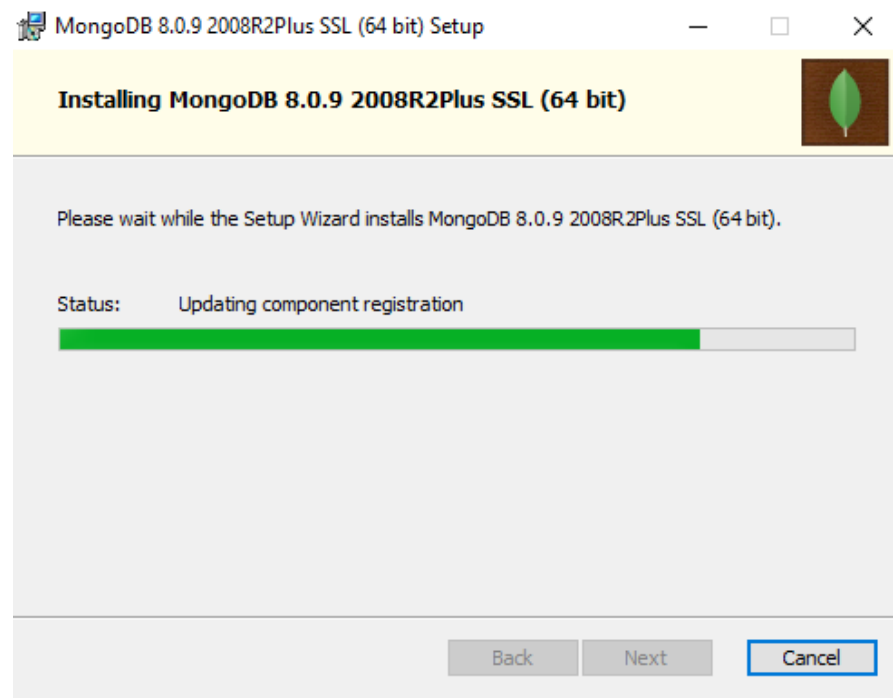


Рисунок 3.2 - Інсталяція MongoDB

Для розробки та запуску клієнтської частини було встановлено Angular CLI. Для цього у командному рядку виконала команду `npm install -g @angular/cli`, що дозволило глобально встановити інструмент для створення та керування Angular-проектами. Після встановлення перевірила його доступність командою `ng version`.

Далі я створила окремі папки для серверної (backend) та клієнтської (frontend) частин проекту. У кожній з цих папок відкрила командний рядок і виконала команду `npm install`, яка автоматично завантажила всі необхідні залежності, зазначені у файлах `package.json`. Для налаштування підключення

до бази даних відкрила відповідний конфігураційний файл config/db.js і вказала параметри підключення до локального екземпляра MongoDB [29].

Після підготовки середовища запустила серверну частину проекту командою `npm run start` у папці `backend`. Переконалася, що сервер стартував без помилок і готовий до обробки запитів. Далі перейшла до папки `frontend` і виконала команду `nodemon`, яка запустила клієнтську частину проекту у режимі розробки. Після цього у браузері відкрила адресу `http://localhost:4200` і перевірила, що сайт коректно відображається та взаємодіє із сервером.

У разі необхідності розгортання проекту на іншому комп'ютері або сервері, я повторювала ці кроки: встановлювала Node.js, MongoDB, Angular CLI, копіювала файли проекту, встановлювала залежності та налаштовувала параметри підключення. Кожен етап встановлення супроводжувався перевіркою коректності роботи відповідного компонента.

3.3 Інструкція з Розміщення сайту на хостингу

Розміщення сайту на хостингу передбачає перенесення всіх файлів проекту на віддалений сервер, налаштування середовища виконання та підключення до бази даних. Спочатку необхідно підготувати фінальну версію коду як фронтенду (Angular), так і бекенду (Node.js + Express). Для цього фронтенд збирається у продакшн-режимі командою `ng build --prod`, після чого вміст папки `dist/` копіюється на сервер. Серверна частина також переноситься на хостинг, де встановлюються всі необхідні залежності за допомогою команди `npm install` [31].

Далі потрібно налаштувати підключення до бази даних MongoDB, вказавши відповідний URI у файлі конфігурації або змінних середовища. Базу даних можна розмістити на MongoDB Atlas або на власному сервері [29]. Після цього серверна частина запускається командою `node index.js` або за допомогою менеджера процесів, наприклад, PM2 [31]. Для коректної роботи сайту з

					2025.КВР.123.412.10.00.00 ПЗ	Арк.
						54
Змн.	Арк.	№ докум.	Підпис	Дата		

доменним іменем часто налаштовується проксі-сервер, такий як Nginx[31], який перенаправляє запити на Node.js застосунок.

Після завершення налаштувань і запуску всіх сервісів сайт стає доступним для користувачів за вказаним доменом. На завершальному етапі перевіряється коректність роботи основних функцій сайту, таких як реєстрація, вхід, створення постів, робота адміністративної панелі та відображення контенту на різних пристроях.

3.4 Інструкція з експлуатації веб-сайту

Інструкція з експлуатації веб-сайту містить основні правила та рекомендації щодо використання ресурсу для різних категорій користувачів. Після розміщення сайту на хостингу користувачі можуть отримати до нього доступ через браузер за допомогою доменного імені. Для повноцінної роботи з функціоналом необхідно пройти реєстрацію, вказавши логін, електронну пошту та пароль, після чого можна виконати вхід у систему. [26]

Зареєстрований користувач отримує можливість створювати нові пости, переглядати власні та чужі публікації, залишати коментарі, а також редагувати або видаляти свої записи через особистий кабінет. Для створення поста потрібно перейти у відповідний розділ, заповнити форму із заголовком, текстом, категорією та, за бажанням, додати зображення. Після збереження пост з'явиться на головній сторінці та буде доступний для перегляду іншими користувачами.

Адміністратор сайту має розширені права: він може переглядати список усіх користувачів, змінювати їх ролі, видаляти облікові записи, а також редагувати або видаляти будь-які пости. Для цього передбачена окрема адміністративна панель, доступ до якої мають лише користувачі з відповідними правами. [24]

					2025.КВР.123.412.10.00.00 ПЗ	Арк.
						55
Змн.	Арк.	№ докум.	Підпис	Дата		

Для коректної роботи сайту рекомендується використовувати сучасні браузері (Google Chrome, Mozilla Firefox, Microsoft Edge, Safari) та стабільне підключення до Інтернету. У разі виникнення проблем із доступом або функціональністю слід звернутися до адміністратора ресурсу або служби технічної підтримки.

					2025.КВР.123.412.10.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		56

4 ЕКОНОМІЧНИЙ РОЗДІЛ

Метою економічної частини є обґрунтування доцільності розробки та впровадження інформаційної веб-платформи "Briolin" для спільноти барберів, доглядом за бородою, трендами та культурою барбершопів. У розділі виконано розрахунки витрат на розробку, визначено собівартість, ціну, а також економічну ефективність проєкту.

4.1 Визначення стадій технологічного процесу та загальної тривалості проведення НДР

Для визначення загальної тривалості проведення НДР доцільно дані витрат часу по окремих операціях технологічного процесу зведено у таблиці 4.1.

Таблиця 4.1 — Середній час виконання НДР та стадії технологічного процесу

№ п/п	Назва операції (стадії)	Виконавець	Середній час виконання операції, год.
1	видача завдання та постановка задачі	керівник проєкту	6 година
2	аналіз задачі, складання технічного завдання	розробник	8 годин
3	планування	розробник	12 годин
4	проєктування	розробник	10 годин
5	розробка (програмування)	розробник	140 годин
6	розгортання	розробник	12 годин
7	тестування	розробник	8 годин
8	затвердження задачі проєкту	керівник проєкту	2 години
Разом		—	198 годин

Згідно, з даними у таблиці, на розробку проєкту вебсайту для барберів «Briolin» було виділено 198 годин [17].

4.2 Визначення витрат на оплату праці та відрахувань на соціальні заходи

Відповідно до Закону України «Про оплату праці» заробітна плата — це «винагорода, обчислена, як правило, у грошовому виразі, яку власник або уповноважений ним орган виплачує працівникові за виконану ним роботу». Розмір заробітної плати залежить від складності та умов виконуваної роботи, професійно-ділових якостей працівника, результатів його праці та господарської діяльності підприємства. Заробітна плата складається з основної та додаткової оплати праці [19].

Основна заробітна плата нараховується на виконану роботу за тарифними ставками, відрядними розцінками чи посадовими окладами і не залежить від результатів господарської діяльності підприємства. Додаткова заробітна плата — це складова заробітної плати працівників, до якої включають витрати на оплату праці, не пов'язані з виплатами за фактично відпрацьований час. Нараховують додаткову заробітну плату залежно від досягнутих і запланованих показників, умов виробництва, кваліфікації виконавців. Джерелом додаткової оплати праці є фонд матеріального стимулювання, який створюється за рахунок прибутку.

Місячний оклад кожного працівника слід враховувати згідно існуючих на даний час тарифних окладів [17].

Основна заробітна плата розраховується за формулою:

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

де T_c — тарифна ставка, грн;

K_r — кількість відпрацьованих годин.

Звідси, для керівника проєкту $Z_{осн.} = 300 \cdot 8 = 2400$ гривень,

для розробника — $Z_{осн.} = 50 \cdot 190 = 9500$ гривень.

					2025.КВР.123.412.10.00.00 ПЗ	Арк.
						58
Змн.	Арк.	№ докум.	Підпис	Дата		

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 0,1 — 0,15.

Для керівника проєкту $Z_{\text{дод.}} = 2400 \cdot 0,13 = 312$ гривні,

для розробника — $Z_{\text{дод.}} = 9500 \cdot 0,1 = 950$ гривень.

Звідси загальні витрати на оплату праці ($B_{\text{о.п.}}$) визначаються за формулою:

$$B_{\text{о.п.}} = Z_{\text{осн.}} + Z_{\text{дод.}} \quad (4.3)$$

Для керівника — $2400 + 312 = 2712$ гривні,

для розробника — $9500 + 950 = 10450$ гривень.

Крім того, слід визначити суму нарахування на заробітну плату:

єдиний соціальний внесок — 22%;

Отже, сума нарахувань на заробітну плату буде становити:

$$B_{\text{с.з.}} = \text{ФОП} \cdot 0,22 \quad (4.4)$$

де, ФОП — фонд оплати праці, грн.

$$B_{\text{с.з.}} = 13162 \cdot 0,22 = 2895,6 \text{ гривень.}$$

Проведені розрахунки витрат на оплату праці зведено у таблиці 4.2.

Таблиця 4.2 — Зведені розрахунки витрат на оплату праці

№ п/п	Категорія працівни- ків	Основна заробітна плата, грн.			Додатк зароб. плата, грн.	На- рах. на ФОП, грн.	Всього витрати на оплату праці, грн.
		Та- риф. ста- вка, грн.	К-сть від- працьов. год.	Фактично нарах. з/пл., грн.			
1	керівник	300	8	2400	312	-	-
2	розробник	50	190	9500	950	-	-
Разом		350	198	11900	1262	2895,6	16057,6

4.3 Розрахунок витрат на електроенергію

Затрати на електроенергію 1-ці обладнання визначаються за формулою:

$$Z_e = W \cdot T \cdot S \quad (4.5)$$

де, W — необхідна потужність, кВт;

T — кількість годин роботи обладнання;

S — вартість кіловат-години електроенергії.

Для наявного персонального комп'ютера у розробника HP EliteDesk 800 G1 SFF потужність = 0,24 кВт, $T = 170$ год.

Вартість кіловат-години електроенергії слід приймати згідно існуючих на даний час тарифів (7 грн).

Звідси,

$$Z_e = 0,24 \cdot 170 \cdot 7 = 285,6 \text{ грн}$$

4.5 Розрахунок суми амортизаційних відрахувань

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. Для наявного ПК у розробника HP EliteDesk 800 G1 SFF на момент купівлі у 2021 році коштував 7000 грн. Оскільки дана сума перевищує 2500 грн — нараховується амортизація. Мінімально допустимі строки корисного використання 2 роки, звідси норма амортизації становить 20%.

Для визначення амортизаційних відрахувань застосовуємо формулу:

$$A = \frac{B_v \cdot H_a}{100\%} \quad (4.6)$$

					2025.КВР.123.412.10.00.00 ПЗ	Арк.
						60
Змн.	Арк.	№ докум.	Підпис	Дата		

де А — амортизаційні відрахування за звітний період, грн.;

Б_в — балансова вартість групи основних фондів на початок звітного періоду, грн.;

Н_а — норма амортизації, %.

$$A = \frac{17000 \cdot 0,04}{150} * 170 = 317,33$$

4.6 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.

$$H_v = B_{o,n} \cdot 0,2 \dots 0,6 \quad (4.7)$$

де, Н_в — накладні витрати.

$$\text{Звідси, } H_v = 13162 \cdot 0,2 = 4760 \text{ гривень.}$$

4.7 Складання кошторису витрат та визначення собівартості НДР

Результати проведених вище розрахунків зведено у таблиці 4.4.

Таблиця 4.4 — Кошторис витрат на НДР

Зміст витрат	Сума, грн.	В % до загальної суми
Витрати на оплату праці (основну і додаткову заробітну плату)	13162,00	62,2%
Відрахування на соціальні заходи	2895,6	12,4%
Витрати на електроенергію	285,6	1,4%
Амортизаційні відрахування	317,33	1,5%
Накладні витрати	4760	22,5
Собівартість	21142,93	100%

Собівартість(C_v) НДР розрахуємо за формулою:

$$C_v = B_{o.n.} + B_{c.z.} + Z_{m.v.} + Z_e + A + H_v \quad (4.8)$$

Звідси, $C_v = 21142,93$ грн.

4.8 Розрахунок ціни НДР

Ціну НДР можна визначити за формулою:

$$Ц = (C_v \cdot (1 + P_{рен}) + B_{н.і.}) \cdot (1 + ПДВ) \quad (4.9)$$

де, $P_{рен.}$ — рівень рентабельності;

$B_{н.і.}$ — вартість носія інформації, грн.;

ПДВ — ставка податку на додану вартість, (20 %).

Звідси, ціна НДР становитиме:

$$Ц = 21142,93 \cdot (1 + 0,3) \cdot (1 + 0,2) = 32982,97 \text{ гривні.}$$

4.9 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 ($T_{ок}$).

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

де K_v — затрати на проєкт;

$\Gamma_{п}$ — грошовий потік за t -ий рік;

t — відповідний рік проєкту;

i — величина дисконтної ставки (10...15%).

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

					2025.КВР.123.412.10.00.00 ПЗ	Арк.
						62
Змн.	Арк.	№ докум.	Підпис	Дата		

$$\text{ЧТВ} = -21142,93 + \frac{12000}{(1 + 0,1)} + \frac{12000}{(1 + 0,1)^2} + \frac{12000}{(1 + 0,1)^3} = 8699,37$$

Термін окупності визначається за формулою:

$$T_{\text{ок}} = T_{\text{пв}} + \frac{H_{\text{в}}}{\Gamma_{\text{пр}}} \quad (4.11)$$

де $T_{\text{пв}}$ — період до повного відшкодування витрат, років;

$H_{\text{в}}$ — невідшкодовані витрати на початок року, грн;

$\Gamma_{\text{пр}}$ — грошовий потік на початок року, грн.

$$T_{\text{ок}} = 2 + \frac{316,43}{12000} = 2 \text{ роки}$$

Всі дані розрахунків внесено у зведену таблицю 4.5 техніко-економічних показників.

Таблиця 4.5 — Техніко-економічні показники НДР

№ п/п	Показник	Значення
1.	Собівартість, грн.	21142,93
2.	Плановий прибуток, грн.	12000
3.	Ціна, грн.	32982,97
4.	Чиста теперішня вартість, грн.	
5.	Термін окупності, рік	2

Собівартість вебсайту становить 21142,93 гривень, а загальна його вартість — 32982,97 гривень. Термін окупності проєкту — 2 рік.

Таким чином, розробка та впровадження веб-платформи “Briolin” є економічно доцільними, оскільки проєкт має позитивну чисту теперішню вартість і короткий термін окупності. Це свідчить про ефективність інвестицій та перспективність подальшого розвитку платформи для спільноти барберів.

5. ОХОРОНА ПРАЦІ ТА БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ

У процесі виконання кваліфікаційної роботи, що передбачає створення, розробку та тестування веб-сайту для барберів «Briolin», особливе значення має дотримання вимог охорони праці, техніки безпеки та екологічних стандартів. Комп'ютерна інженерія як сфера професійної діяльності має свою специфіку — вона пов'язана з тривалим перебуванням працівника за екраном монітора, роботою в сидячому положенні, високим рівнем інтелектуального навантаження та взаємодією з електронним обладнанням [16].

Відповідно, виникає необхідність створення таких умов праці, які мінімізують негативний вплив шкідливих і небезпечних чинників на здоров'я розробника. Це включає правильну організацію робочого місця, дотримання ергономічних вимог, забезпечення раціонального режиму праці та відпочинку, оптимізацію мікроклімату і освітлення в приміщенні, а також впровадження заходів із профілактики професійного стресу та емоційного вигорання.

У цьому розділі детально розглядаються такі аспекти безпеки та охорони праці:

1. Шкідливі чинники комп'ютерних технологій та засоби захисту від них;
2. Вимоги до освітлення виробничих приміщень;
3. Психологія безпеки праці — заходи з профілактики стресу, монотонності та збереження ментального здоров'я працівника.

5.1 Шкідливі чинники комп'ютерних технологій та захист від них

Робота за комп'ютером супроводжується дією кількох груп потенційно шкідливих факторів. Їх тривала дія без дотримання профілактичних заходів може спричинити професійні захворювання, зниження продуктивності та психологічне виснаження.

					2025.КВР.123.412.10.00.00 ПЗ	Арк.
						64
Змн.	Арк.	№ докум.	Підпис	Дата		

Основні групи шкідливих чинників:

1. Фізичні чинники:

Електромагнітне випромінювання (ЕМВ): сучасні монітори типу LCD або LED значно знизили рівень ЕМВ порівняно з CRT-дисплеями, однак при тривалому впливі можливе накопичення електростатичного заряду, негативний вплив на нервову систему. Джерела ЕМВ – монітори, Wi-Fi-модулі, маршрутизатори, блоки живлення[18].

Шум: рівень шуму від системного блоку, вентиляторів, зовнішніх пристроїв (принтери, сканери) не повинен перевищувати 50 дБ. Перевищення – призводить до втоми, головного болю, погіршення концентрації.

Мікроклімат: температура в приміщенні повинна бути в межах 20–24 °С, відносна вологість – 40–60%, швидкість руху повітря – не більше 0,1–0,2 м/с [16].

2. Біологічні чинники:

Відсутні або мінімальні у типових умовах офісу, але при недотриманні гігієнічних норм можливе скупчення пилу на вентиляційних отворах техніки, розвиток грибків у вологому середовищі.

3. Хімічні чинники:

Матеріали корпусів, тонер у принтерах, старі флуоресцентні лампи можуть виділяти леткі органічні сполуки або пил, що містить небезпечні речовини (озон, формальдегід).

4. Психофізіологічні чинники:

Статична поза. Сидіння без належної підтримки та регулярних перерв сприяє розвитку остеохондрозу, варикозу, тунельного синдрому.

Втома зору. Надмірне навантаження на очі при роботі з екранами без належного відпочинку призводить до синдрому "сухого ока", зниження гостроти зору, болю та спазмів.

Когнітивне перевантаження. Постійне переключення між задачами, дедлайни, багатозадачність підвищують рівень стресу та провокують синдром емоційного вигорання.

					2025.КВР.123.412.10.00.00 ПЗ	Арк.
						65
Змн.	Арк.	№ докум.	Підпис	Дата		

Комплекс заходів захисту:

- Застосування ергономічних меблів: крісло з регульованою спинкою та підлокітниками, стіл із відповідною висотою.
- Встановлення монітора на рівні очей, на відстані 50–70 см від обличчя.
- Використання підставки для ніг для запобігання застою крові.
- Регламентовані перерви за методикою 45–10 (кожні 45 хвилин роботи – 10 хвилин активного відпочинку).
- Гімнастика для очей (погляд у далечінь, кругові рухи очима), та профілактична фізкультура.
- Встановлення антивідблискових фільтрів на монітори (у разі відсутності антиблікового покриття).
- Застосування програм для нагадування про перерви (EyeLeo, Stretchly тощо).

5.2 Вимоги до освітлення виробничих приміщень

Освітлення – один із найважливіших факторів, що впливає на працездатність, зорове навантаження та загальний комфорт працівника. Вимоги до освітлення регламентуються стандартами:

— ДСТУ EN 12464-1:2017 «Світло та освітлення. Освітлення робочих місць у приміщеннях» [14].

Вимоги до природного освітлення:

Вікна мають розташовуватись ліворуч від робочого місця. Коефіцієнт природної освітленості (КПО) має бути не менше 1,5% для офісних приміщень. Не допускається пряме сонячне світло на екран монітора – потрібно використовувати жалюзі або штори.

Вимоги до штучного освітлення:

					2025.КВР.123.412.10.00.00 ПЗ	Арк.
						66
Змн.	Арк.	№ докум.	Підпис	Дата		

Мінімальна освітленість робочої поверхні – 300–500 лк. Світильники – переважно LED, з індексом кольоропередачі $Ra \geq 80$, колірна температура – 4000–5000 К. Освітлення повинне бути рівномірним, без пульсацій (коефіцієнт пульсацій не більше 10%). Місцеве освітлення (настільна лампа) має мати матовий плафон, який не створює відблисків на екрані.

Оцінка освітлення:

Замір освітленості проводиться люксометром у центрі робочої поверхні. За потреби застосовують автоматичні системи регулювання освітлення (датчики руху, датчики яскравості).

5.3 Психологія безпеки праці

Психологія безпеки праці є важливим напрямом системи охорони праці, який спрямований на створення сприятливого психоемоційного середовища на робочому місці, зменшення ризиків психічного перенавантаження, профілактику професійного стресу та емоційного вигорання працівників[18]. Особливе значення ця проблема набуває у сфері ІТ, де працівники часто мають справу з інтенсивною розумовою діяльністю, високою відповідальністю, стислими термінами виконання завдань і тривалим перебуванням за комп'ютером [16].

Основні чинники психологічного ризику:

Монотонність праці. Виконання одноманітних завдань протягом тривалого часу, без зміни виду діяльності та творчих викликів, призводить до зниження мотивації, втрати інтересу до роботи та розвитку стану апатії. У комп'ютерних спеціальностях це характерно для рутинного тестування, копіювання коду або обробки великих обсягів даних.

Інформаційне перевантаження. Постійний потік інформації з різних джерел (завдання, листи, чати, повідомлення в месенджерах), робота у багато-задачному режимі (multitasking), необхідність оперативного прийняття рішень

					2025.КВР.123.412.10.00.00 ПЗ	Арк.
						67
Змн.	Арк.	№ докум.	Підпис	Дата		

— усе це виснажує когнітивні ресурси працівника, спричиняє зниження концентрації, появу тривожності та нервового напруження.

Хронічний стрес. Постійні дедлайни, порушення таймінгу у проектах, баги в системі, відповідальність за безперервну роботу інфраструктури, а також конфліктні ситуації між членами команди — усе це створює умови для виникнення тривалого психоемоційного напруження, яке може перерости у синдром хронічного стресу.

Відсутність соціальної підтримки. Недостатній рівень комунікації з колегами, ігнорування з боку керівництва, відсутність зворотного зв'язку та оцінки зусиль працівника сприяють формуванню почуття ізольованості, зниженню мотивації до роботи та погіршенню психологічного стану.

Засади психологічної безпеки праці:

Рациональна організація робочого процесу. Важливою умовою є впровадження гнучкого графіку роботи, оптимальний розподіл завдань, уникнення надмірного навантаження на окремих співробітників, планування щоденного графіка з урахуванням регулярних перерв на відпочинок, релаксацію або зміну діяльності.

Зменшення психоемоційного напруження. Досягається завдяки чергуванню розумової та фізичної активності, запровадженню практик mindfulness (усвідомленості), дихальних технік, коротких фізкультурних пауз, прослуховування спокійної музики або перебування на свіжому повітрі.

Формування позитивного психологічного клімату в колективі. Сприятлива атмосфера у трудовому колективі передбачає взаємоповагу, відкриту комунікацію, підтримку з боку колег та керівництва, конструктивне вирішення конфліктів і визнання досягнень кожного працівника.

Підтримка балансу між роботою та особистим життям (work-life balance). Заохочення працівників до збереження вільного часу для сім'ї, відпочинку, особистих хобі, дозвілля.

					2025.КВР.123.412.10.00.00 ПЗ	Арк.
						68
Змн.	Арк.	№ докум.	Підпис	Дата		

А також дотримання «цифрової гігієни» — зменшення часу перебування онлайн у неробочі години, уникнення нічних повідомлень або позапланової роботи.

Профілактика емоційного вигорання. Регулярне проведення психологічних тренінгів, індивідуальних коучинг-сесій, навчання тайм-менеджменту, створення служб підтримки або залучення корпоративного психолога — важливі елементи формування стійкості до емоційного вигорання.

Розвиток психологічної грамотності працівників. Ознайомлення персоналу з основами саморегуляції, самодіагностики емоційного стану, вивчення технік управління стресом, підвищення культури безпечної поведінки на робочому місці.

					2025.КВР.123.412.10.00.00 ПЗ	Арк.
						69
Змн.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВКИ

У ході виконання кваліфікаційної роботи було розроблено сучасний веб-сайт «Briolin», який є блоговою платформою для барберів, клієнтів та всіх, хто цікавиться барбер-культурою. Для реалізації проекту було використано технологічний стек MEAN (MongoDB, Express, Angular, Node.js), що забезпечило високу продуктивність, масштабованість та зручність у подальшій підтримці.

Було спроектовано структуру даних, що включає користувачів, пости, категорії та повідомлення, з урахуванням нормалізації для уникнення дублювань та забезпечення цілісності. Логіка роботи системи організована за принципом ієрархічної деталізації, що дозволило чітко розмежувати функції керування користувачами, постами, пошуку, адміністрування і локалізації.

Розроблено інтерфейс особистого кабінету користувача з використанням Angular та TypeScript, що забезпечує комфортне управління публікаціями, включаючи створення, редагування, видалення та перегляд постів.

Встановлення та розгортання проекту на локальному комп'ютері і на віддаленому сервері було здійснено з дотриманням сучасних стандартів, що гарантує стабільність і безпеку роботи сайту.

Користувацька експлуатація ресурсу передбачає зручний доступ через браузер із застосуванням сучасних технологій аутентифікації, а адміністративна панель дозволяє ефективно управляти користувачами та контентом.

Таким чином, поставлені цілі були успішно досягнуті: створено функціональний вебсайт із зручним інтерфейсом, надійною архітектурою та можливістю подальшого розвитку і масштабування. Розроблений продукт може бути рекомендований для впровадження у сфері онлайн-блогінгу, зокрема для барберської спільноти.

					2025.КВР.123.412.10.00.00 ПЗ	Арк.
						70
Змн.	Арк.	№ докум.	Підпис	Дата		

ПЕРЕЛІК ПОСИЛАНЬ

1. Brown E., Wilson J. Full Stack Development with MEAN: Modern Web Development with MongoDB, Express, Angular, and Node.js / E. Brown, J. Wilson – Packt Publishing, 2019. – 400 с.
2. Crockford D. JavaScript: The Good Parts / D. Crockford – O'Reilly Media, 2008. – 176 с.
3. Dayley B., Dayley B., Dayley C. Node.js, MongoDB and Angular Web Development / B. Dayley, B. Dayley, C. Dayley – Addison-Wesley, 2014.
4. Flanagan D. JavaScript: The Definitive Guide. 7th Edition / D. Flanagan – O'Reilly Media, 2020. – 706 с.
5. Freeman A., Sanderson P. Pro ASP.NET Core MVC 2 / A. Freeman, P. Sanderson – Apress, 2021.
6. Gamma E., Helm R., Johnson R., Vlissides J. Design Patterns: Elements of Reusable Object-Oriented Software / E. Gamma et al. – Addison-Wesley, 1994.
7. Fowler M. UML Distilled: A Brief Guide to the Standard Object Modeling Language. 3rd Edition / M. Fowler – Addison-Wesley, 2003.
8. Hartl M. Ruby on Rails Tutorial: Learn Web Development with Rails. 6th Edition / M. Hartl – Addison-Wesley, 2021. – 752 с.
9. Pressman R. Software Engineering: A Practitioner's Approach. 8th Edition / R. Pressman – McGraw-Hill Education, 2014.
10. Sommerville I. Software Engineering. 10th Edition / I. Sommerville – Pearson, 2015.
11. Stobart S. Dynamic Web Application Development Using PHP and MySQL / S. Stobart – Cengage Learning, 2014. – 420 с.
12. Welling L., Thomson L. PHP and MySQL Web Development. 5th Edition / L. Welling, L. Thomson – Addison-Wesley, 2016. – 1008 с.
13. ДСТУ 4159-2003. Програмне забезпечення. Документація. – Київ: Держспоживстандарт України, 2003.

					2025.KBP.123.412.10.00.00 ПЗ	Арк.
						71
Змн.	Арк.	№ докум.	Підпис	Дата		

14. ДСТУ EN 12464-1:2017. Світло та освітлення. Освітлення робочих місць у приміщеннях. – Київ: ДП «УкрНДНЦ», 2017.

15. ISO/IEC 25010 (SQaRE). Системи та програмне забезпечення – Моделі якості.

16. Кисельов О.І. Охорона праці в комп'ютерних системах і мережах: навчальний посібник. – Львів: Вид-во ЛНУ, 2020. – 212 с.

17. Методичні вказівки до дипломного проектування для студентів спеціальності 5.091504 «Обслуговування комп'ютерних та інтелектуальних систем і мереж». – Тернопіль: ТНТУ, 2020. – 48 с.

18. Чернявська В., Дуброва Н. Основи безпечної праці: навчальний посібник для здобувачів професійної (професійно-технічної) освіти. Київ: ТОВ «ПРОПАРІР», 2023. 240с.

19. Закон України «Про охорону праці» від 14.10.1992 № 2694-XII (зі змінами і доповненнями). – Київ: Верховна Рада України.

20. Angular – офіційна документація. URL: <https://angular.io> (дата звернення: 18.05.2025).

21. BarberShopCo – офіційний сайт компанії. URL: <https://barbershop.co.nz> (дата звернення: 16.05.2025).

22. Booksy – офіційна документація. URL: <https://booksy.com> (дата звернення: 16.05.2025).

23. ESLint. Linting Rules and Style Guide. URL: <https://eslint.org> (дата звернення: 16.05.2025).

24. Express – офіційна документація. URL: <https://expressjs.com> (дата звернення: 16.05.2025).

25. International Barber Association. Building Your Online Barber Portfolio: Tools and Best Practices. URL: <https://www.internationalbarberassociation.com/resources> (дата звернення: 16.05.2025).

26. JWT.io. Introduction to JSON Web Tokens. URL: <https://jwt.io/introduction/> (дата звернення: 20.05.2025).

					2025.KBP.123.412.10.00.00 ПЗ	Арк.
						72
Змн.	Арк.	№ докум.	Підпис	Дата		

27. Microsoft Learn. Writing Technical Documentation. URL: <https://learn.microsoft.com/en-us/style-guide/> (дата звернення: 24.05.2025).

28. MongoDB Documentation. Manual v6.0. URL: <https://www.mongodb.com/docs/manual/> (дата звернення: 25.05.2025).

29. MongoDB: Schema Design & Data Modeling. URL: <https://www.mongodb.com/docs/manual/core/data-modeling-introduction/> (дата звернення: 18.05.2025).

30. Mozilla Developer Network (MDN). URL: <https://developer.mozilla.org> (дата звернення: 20.05.2025).

31. Node.js – офіційна документація. URL: <https://nodejs.org/en/docs/> (дата звернення: 21.05.2025).

32. OpenAPI Initiative. OpenAPI Specification v3.0. URL: <https://swagger.io/specification/> (дата звернення: 19.05.2025).

33. OWASP Foundation. OWASP Cheat Sheet Series. URL: <https://cheatsheetseries.owasp.org> (дата звернення: 22.05.2025).

34. OWASP Foundation. Secure Coding Practices. URL: <https://owasp.org> (дата звернення: 23.05.2025).

35. The Barber Post – блог барберів. URL: <https://thebarberpost.com> (дата звернення: 19.05.2025).

36. TypeScript Handbook. URL: <https://www.typescriptlang.org/docs/> (дата звернення: 19.05.2025).

37. Visual Studio Code. – офіційний сайт редактора. URL: <https://code.visualstudio.com> (дата звернення: 21.05.2025).

38. W3C. HTML5 Specification. URL: <https://www.w3.org/TR/html5/> (дата звернення: 25.05.2025).

					2025.КВР.123.412.10.00.00 ПЗ	Арк.
						73
Змн.	Арк.	№ докум.	Підпис	Дата		

ДОДАТОК А - КОД СКРИПТА HTML

Файл profile.component.html

```
<div class="profile-info" *ngIf="user">
  <h2>{{ 'USER_PROFILE' | translate }}</h2>
  <p><strong>{{ 'NAME' | translate }}:</strong> {{ user.name }}</p>
  <p><strong>{{ 'LOGIN' | translate }}:</strong> {{ user.login }}</p>
  <p><strong>{{ 'EMAIL' | translate }}:</strong> {{ user.email }}</p>
  <p><strong>{{ 'ROLE' | translate }}:</strong> {{ user.role | translate
}}</p>
</div>

<hr>

<div class="container mt-4">
  <h4>{{ 'YOUR_POSTS' | translate }}</h4>
  <div class="posts-container">
    <ng-container *ngIf="userPosts.length > 0; else noPosts">
      <div *ngFor="let post of userPosts" class="card mb-3">
        <div class="card-body">
          <h5 class="card-title">{{ post.title }}</h5>
          <p class="card-text">{{ 'CATEGORY' | translate }}: {{ post.category
| translate }}</p>
          <div [innerHTML]="post.photo"></div>
          <p class="card-text">{{ getShortText(post.text, 100) }}</p>
          <small class="text-muted">{{ post.date | date:'short' }}</small>
          <div class="mt-2">
            <button class="btn btn-primary btn-sm"
(click)="viewPost(post._id)">{{ 'VIEW' | translate }}</button>
            <button class="btn btn-warning btn-sm"
(click)="editPost(post._id)">{{ 'EDIT' | translate }}</button>
```

					2025.KBP.123.412.10.00.00 ПЗ	Арк.
						74
Змн.	Арк.	№ докум.	Підпис	Дата		

```
        <button          class="btn          btn-danger          btn-sm"
(click)="deletePost(post._id)">{{ 'DELETE' | translate }}</button>
    </div>
</div>
</div>
</ng-container>
<ng-template #noPosts>
    <p>{{ 'NO_POSTS_YET' | translate }}</p>
</ng-template>
</div>
</div>
```

					2025.KBP.123.412.10.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		75

ДОДАТОК В - КОД СКРИПТА CSS

Файл profile.component.scss

```
.card {  
  width: 400px;  
  height: 500px;  
  margin: 0;  
  display: flex;  
  flex-direction: column;  
  justify-content: flex-start;  
  border-radius: 10px;  
  box-shadow: 0 2px 8px rgba(0,0,0,0.06);  
  border: none;  
  transition: transform 0.15s, box-shadow 0.15s;  
  background: #fff;  
}  
  
.card:hover {  
  transform: translateY(-4px) scale(1.02);  
  box-shadow: 0 6px 24px rgba(0,0,0,0.12);  
}  
  
.card-body {  
  padding: 0.5rem;  
  text-align: center;  
  height: 100%;  
  display: flex;  
  flex-direction: column;  
  justify-content: flex-start;  
}
```

					2025.KBP.123.412.10.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		76

```
::ng-deep .card-body img {
  width: 320px !important;
  height: 200px !important;
  object-fit: cover !important;
  display: block !important;
  margin: 0 auto 10px auto !important;
  border-radius: 8px !important;
}
```

```
.posts-container {
  display: grid;
  grid-template-columns: repeat(auto-fill, minmax(350px, 1fr));
  gap: 24px;
  align-items: stretch;
  margin-top: 24px;
}
```

```
.profile-info {
  background: #f8f9fa;
  border-radius: 10px;
  padding: 24px 32px;
  margin-bottom: 32px;
  box-shadow: 0 2px 8px rgba(0,0,0,0.04);
  max-width: 500px;
  border-left: 6px solid #000000;
}
```

```
.profile-info h2 {
  font-weight: 700;
  margin-bottom: 18px;
}
```

					2025.KBP.123.412.10.00.00 ПЗ	Арк.
						77
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    color: #000000;
    letter-spacing: 1px;
}

.profile-info p {
    margin: 0 0 8px 0;
    font-size: 1.05rem;
}

.container.mt-4 {
    max-width: 800px;
}

.card-title {
    font-size: 1.25rem;
    font-weight: 600;
    color: #343a40;
}

.card-text {
    margin-bottom: 8px;
}

.btn-sm {
    min-width: 80px;
    margin-right: 8px;
    border-radius: 5px;
    font-weight: 500;
    transition: background 0.15s;
}

```

					2025.KBP.123.412.10.00.00 ПЗ	Арк.
						78
Змн.	Арк.	№ докум.	Підпис	Дата		

```
.btn-primary.btn-sm {
    background: #dcdcdc;
    border: none;
}

.btn-primary.btn-sm:hover {
    background: #0056b3;
}
```

```
.btn-warning.btn-sm {
    background: #dcdcdc;
    border: none;
    color: #212529;
}

.btn-warning.btn-sm:hover {
    background: #e0a800;
}
```

```
.btn-danger.btn-sm {
    background: #dc3545;
    border: none;
}

.btn-danger.btn-sm:hover {
    background: #a71d2a;
}
```

```
.btn-sm:last-child {
    margin-right: 0;
}
```

					2025.KBP.123.412.10.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		79

```
.text-muted {  
    font-size: 0.95rem;  
}  
  
@media (max-width: 600px) {  
    .profile-info, .container.mt-4 {  
        padding: 10px;  
        max-width: 100%;  
    }  
    .card-body {  
        padding: 12px;  
    }  
    .card {  
        width: 100%;  
        height: auto;  
    }  
}
```

					2025.KBP.123.412.10.00.00 ПЗ	Арк.
						80
Змн.	Арк.	№ докум.	Підпис	Дата		