

Міністерство освіти і науки України  
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії  
(повна назва факультету)

Кафедра комп'ютерних наук  
(повна назва кафедри)

# КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

Бакалавр

(назва освітнього ступеня)

на тему: Розробка веб-сервісу для бронювання ліків

Виконав: студент IV курсу, групи СН-41

спеціальності 122 Комп'ютерні науки  
(шифр і назва спеціальності)

(підпис)

Воробець І.В.

(прізвище та ініціали)

Керівник

(підпис)

Фриз М.Є.

(прізвище та ініціали)

Нормоконтроль

(підпис)

Шимчук Г.В.

(прізвище та ініціали)

Завідувач кафедри

(підпис)

Боднарчук І.О.

(прізвище та ініціали)

Рецензент

(підпис)

Пастух О.А.

(прізвище та ініціали)

Тернопіль  
2021



## 6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Безпека життєдіяльності, основи охорони праці	Гурик О.Я., доцент кафедри МТ		

7. Дата видачі завдання \_\_\_\_\_ 25 січня 2021 р. \_\_\_\_\_

## КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	Ознайомлення з завданням до кваліфікаційної роботи	25.01.2021	Виконано
2.	Підбір та аналіз джерел щодо бронювання ліків, веб-сервісів для бронювання ліків	26.01.2021- 05.02.2021	Виконано
3.	Переклад та опрацювання джерел про розробку веб-сервісів	06.02.2021- 12.02.2021	Виконано
4.	Розроблення веб-сервісу для бронювання ліків	13.02.2021- 24.02.2021	Виконано
5.	Оформлення розділу «Аналіз предметної області та проектування веб-сервісу для бронювання ліків»	07.06.2021- 08.06.2021	Виконано
6.	Оформлення розділу «Розробка, розгортання та використання веб-сервісу для бронювання ліків»	09.06.2021- 10.06.2021	Виконано
7.	Виконання завдань до розділу «Безпека життєдіяльності, основи охорони праці»	11.06.2021- 12.06.2021	Виконано
8.	Оформлення кваліфікаційної роботи	13.06.2021	Виконано
9.	Нормоконтроль	14.06.2021	Виконано
10.	Перевірка кваліфікаційної роботи на плагіат	15.06.2021	Виконано
11.	Попередній захист кваліфікаційної роботи	18.06.2021	Виконано
12.	Захист кваліфікаційної роботи	21.06.2021	

Студент

\_\_\_\_\_ (підпис)

Воробець І.В.

\_\_\_\_\_ (прізвище та ініціали)

Керівник роботи

\_\_\_\_\_ (підпис)

Фриз М.Є.

\_\_\_\_\_ (прізвище та ініціали)

## АНОТАЦІЯ

Розробка веб-сервісу для бронювання ліків // Кваліфікаційна робота освітнього рівня «Бакалавр» // Воробець Ігор Володимирович // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра комп'ютерних наук, група СН-41 // Тернопіль, 2021 // С. 56, рис. – 11, табл. – 0, слайдів – 17, додат. – 5, бібліогр. – 45.

**Ключові слова:** бронювання, ліки, аптечні мережі, веб-сервіс, трирівнева архітектура, база даних, веб-хостинг, CDN-хостинг.

Кваліфікаційна робота присвячена проектуванню й розробці веб-сервісу для бронювання ліків для його подальшого використання в невеликій аптечній мережі регіонального рівня.

У першому розділі кваліфікаційної роботи виконано аналіз предметної області та огляд наявних в Україні веб-сервісів для бронювання ліків. Здійснено формування вимог, зовнішніх сутностей і варіантів використання. Проведено проектування й опис архітектури веб-сервісу.

У другому розділі кваліфікаційної роботи здійснено вибір засобів розробки веб-сервісу та проведено його обґрунтування. Виконано розробку моделі даних і компонентів веб-сервісу для бронювання ліків. Проведено розгортання розробленого веб-сервісу на вибраному веб-хостингу.

У розділі «Безпека життєдіяльності, основи охорони праці» визначено основні психологічні чинники небезпеки, що призводять до нещасних випадків, та здійснено розгляд вимог ергономіки до організації робочого місця оператора ПК.

## ANNOTATION

Web-service development for medicines booking // Qualifying paper for the degree of «Bachelor» // Vorobets Ihor Volodymyrovych // Ternopil Ivan Puluj National Technical University, Faculty of Computer Information Systems and Software Engineering, Computer Science Department, group CH-41 // Ternopil, 2021 // P. 56, fig. – 11, slides – 17, annexes – 5, ref. – 45.

**Keywords:** booking, medicines, pharmacy networks, web service, three-tier architecture, database, web hosting, CDN hosting.

The qualification paper is devoted to design and development of a web service for medicines booking for further use in a small pharmacy network at the regional level.

In the first section of the qualification paper, an analysis of the subject area and overview of available in Ukraine web services for medicines booking were performed. The requirements, actors and use cases formation was conducted. The design and description of the web service architecture was carried out.

In the second section of the qualification paper, the choice of means of web service development was made and its substantiation was carried out. The data model and components of web service for medicines booking have been developed. The developed web service has been deployed on the selected web hosting.

In the section «Life safety, basics of labor protection» the main psychological risk factors that lead to accidents are identified, and the ergonomics requirements for the organization of the workplace of PC operator are considered.

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

БД – база даних.

ПЗ – програмне забезпечення.

ПК – персональний комп'ютер.

СКБД – система керування базами даних.

AJAX (англ. Asynchronous JavaScript And XML) – асинхронний JavaScript і XML, підхід до створення веб-сторінок з використанням асинхронного (у фоновому режимі) обміну даних між веб-браузером і веб-сервером.

CDN (англ. Content Delivery Network) – мережа доставки контенту.

CRUD (англ. Create, Read, Update, Delete) – чотири базові функції для роботи з даними: створення, читання, оновлення, видалення.

CSS (англ. Cascading Style Sheets) – каскадні таблиці стилів.

E-Mail (англ. electronic mail) – електронна пошта.

ER (англ. entity-relationship) – модель «сутність-зв'язок».

FTP (англ. File Transfer Protocol) – протокол передачі файлів.

HTML (англ. HyperText Markup Language) – мова розмітки гіпертексту.

IT (англ. information technologies) – інформаційні технології.

JS (англ. JavaScript) – динамічна мова програмування, що використовується для створення скриптів веб-сторінок.

PHP (англ. Hypertext Preprocessor, Personal Home Page) – гіпертекстовий препроцесор, скриптова мова програмування, створена для генерування веб-сторінок на стороні веб-сервера.

SQL (англ. Structured Query Language) – мова структурованих запитів, декларативна мова програмування для взаємодії з реляційними базами даних.

URL (англ. Uniform Resource Locator) – уніфікований локатор ресурсів, система стандартизованих адрес електронних ресурсів.

## ЗМІСТ

ВСТУП .....	7
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПРОЕКТУВАННЯ ВЕБ-СЕРВІСУ ДЛЯ БРОНЮВАННЯ ЛІКІВ.....	9
1.1 Аналіз предметної області щодо сфери бронювання ліків .....	9
1.1.1 Короткий опис предметної області .....	9
1.1.2 Огляд наявних веб-сервісів для бронювання ліків .....	10
1.1.3 Постановка завдання.....	13
1.2 Формування вимог до веб-сервісу для бронювання ліків .....	15
1.3 Визначення зовнішніх сутностей і варіантів використання щодо веб- сервісу для бронювання ліків.....	17
1.4 Проектування архітектури веб-сервісу для бронювання ліків.....	22
2 РОЗРОБКА, РОЗГОРТАННЯ ТА ВИКОРИСТАННЯ ВЕБ-СЕРВІСУ ДЛЯ БРОНЮВАННЯ ЛІКІВ.....	25
2.1 Обґрунтування вибору засобів розробки веб-сервісу для бронювання ліків .....	25
2.2 Розробка моделі даних веб-сервісу для бронювання ліків .....	28
2.3 Розробка компонентів веб-сервісу для бронювання ліків .....	32
2.4 Розгортання веб-сервісу для бронювання ліків .....	39
2.4.1 Перенесення зображень на CDN-хостинг.....	40
2.4.2 Пересилання файлів веб-сервісу на веб-хостинг .....	41
2.4.3 Перенесення бази даних веб-сервісу на веб-хостинг .....	42
2.5 Демонстрація роботи веб-сервісу для бронювання ліків.....	44
3 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ .....	46
3.1 Психологічні чинники небезпеки .....	46
3.2 Вимоги ергономіки до організації робочого місця оператора ПК.....	49
ВИСНОВКИ.....	51
ПЕРЕЛІК ДЖЕРЕЛ .....	52
ДОДАТКИ	

## ВСТУП

**Актуальність теми роботи.** XXI століття відзначається широким впровадженням інформаційних технологій у різні сфери людської діяльності, у тому числі й у сферу фармацевтики [1].

Таким чином, при здійсненні продажу фармацевтичних засобів використовується веб-сервіси для бронювання ліків. Такі веб-сервіси стають заміною онлайн-аптек [2]: за допомогою них споживачі можуть переглянути, де продаються необхідні ліки, забронювати, забрати їх у вибраному аптечному закладі й оплатити бронювання.

Вагомим фактором зростання популярності веб-сервісів для бронювання ліків стала пандемія COVID-19. У зв'язку цією пандемією встановлюються карантинні обмеження на максимальну кількість людей, які можуть перебувати у приміщеннях. Тоді споживачам доводиться стояти в черзі перед аптечними закладами та чекати на покупку ліків.

Натомість, за допомогою веб-сервісів вони можуть забронювати ліки онлайн і потім піти до аптечного закладу, де їм просто видадуть ліки.

Отже, у зв'язку зі збільшенням використання інформаційних технологій у фармацевтиці та у зв'язку з карантинними обмеженнями через пандемію COVID-19, розробка веб-сервісу для бронювання є актуальною темою для дослідження.

**Мета і завдання роботи.** Метою розробки є надання додаткових послуг бронювання ліків під час їх продажу, що дозволяє споживачам отримати вищий рівень обслуговування, пов'язаного з покупкою ліків, і дозволяє аптечним закладам залучати нових клієнтів для реалізації ліків.

Для розробки пропонується веб-сервіс, який призначений для використання в аптечній мережі регіонального рівня і зможе зайняти свою нішу на фармацевтичному ринку. Початковою назвою розроблюваного веб-сервісу для бронювання є «PharmBooking».



Для досягнення визначеної мети необхідно виконати наступні завдання:

- провести аналіз джерел, що стосуються сфери бронювання ліків;
- здійснити огляд наявних найбільших веб-сервісів для бронювання ліків, що працюють на території України;
- провести формулювання основних вимог до розроблюваного веб-сервісу для бронювання ліків та здійснити виділення зовнішніх сутностей і варіанти використання;
- здійснити проектування й опис архітектури розроблюваного веб-сервісу для бронювання ліків;
- здійснити проектування та розробку бази даних і інших компонентів розроблюваного веб-сервісу для бронювання ліків, провести опис їх взаємодії та принципів роботи;
- здійснити розгортання веб-сервісу для бронювання ліків шляхом перенесення його компонентів на веб-хостинг.

**Практичне значення отриманих результатів.** Веб-сервіс для бронювання ліків, розроблений у ході виконання кваліфікаційної роботи, може використовуватися невеликими аптечними мережами, такими, як «Сальве» або «Ромашка», що розташовують свої аптечні заклади в регіональних містах.

# 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПРОЕКТУВАННЯ ВЕБ-СЕРВІСУ ДЛЯ БРОНЮВАННЯ ЛІКІВ

## 1.1 Аналіз предметної області щодо сфери бронювання ліків

### 1.1.1 Короткий опис предметної області

Інформаційні технології надають можливість фармацевтичному ринку використовувати нові способи для ведення бізнесу та створення комунікації з кінцевими споживачами. Одним з видів використання інформаційних технологій у фармацевтичній галузі є запровадження онлайн-аптек, що здійснюють реалізацію ліків так само, як і інші інтернет-магазини. Проте, відповідно до українського законодавства (та й законодавства багатьох країн Європейського Союзу), торгівля лікарськими засобами повинна проводитися лише за допомогою аптечних закладів, а здійснення електронної торгівлі або торгівлі через пошту чи інші неаптечні заклади заборонене [2].

Для того, щоб працювати в рамках закону України та використовувати інформаційні технології й мережу Інтернет для оптимізації своєї роботи й отримання більшої кількості нових клієнтів, аптеки можуть використовувати веб-сервіси для бронювання ліків.

У загальному робота таких сервісів виглядає так: споживач хоче купити ліки або навіть декілька різних видів ліків в одній аптеці. Для цього він використовує веб-сервіс для бронювання, переглядає наявність потрібних ліків у аптеках та здійснює їх бронювання в найближчій аптеці, у якій вони доступні. Після цього споживач може в назначений час відвідати вибрану аптеку, здійснити оплату за заброньовані ліки та забрати їх. Аптека ж, натомість, у такий спосіб отримує нового клієнта.

Таким чином, веб-сервіс для бронювання ліків дає вигоду як споживачеві, який здійснює бронювання, так і аптеці, яка виконує це бронювання. Крім цього, такі веб-сервіси можуть надавати наступні переваги:

- нижча вартість лікарських засобів, ніж у звичайних аптеках за рахунок зменшення витрат;
- знижки та персональні пропозиції за допомогою програми лояльності для постійних клієнтів;
- можливість вибирати будь-який зручний для клієнта час під час бронювання;
- надання додаткової інформації про лікарські засоби з метою залучення потенційних клієнтів;
- забезпечення доступності ліків для споживачів, які проживають у великих населених пунктах;
- продаж реклами виробникам лікарських засобів [2, 3, 4].

Саме поєднання позитивних сторін для споживачів й аптек та деякі законодавчі обмеження дали можливість веб-сервісам для бронювання ліків зайняти своє вагоме місце в інтернет-комерції лікарських засобів в Україні і стати засобом цифрової комунікації на фармацевтичному ринку.

### **1.1.2 Огляд наявних веб-сервісів для бронювання ліків**

На українському ринку лікарських засобів можна виділити два основні типи веб-сервісів для бронювання: онлайн-портали, до яких підключаються різні аптеки, та власні веб-сервіси великих аптечних мереж. Серед наявних онлайн-порталів розглянемо веб-сервіси [Tabletki.ua](http://Tabletki.ua) та [GeoApteka](http://GeoApteka), а серед великих аптечних мереж – мережі «Аптека Доброго Дня» й «1 Соціальна Аптека».

[Tabletki.ua](http://Tabletki.ua) – найбільший в Україні онлайн-портал, призначений для бронювання й пошуку лікарських засобів та іншої фармацевтичної продукції [5]. За допомогою [Tabletki.ua](http://Tabletki.ua) користувачі отримують послуги пошуку потрібних ліків у найближчих аптеках. Для користувачів порталу також доступні можливості перегляду інформації про ліки, ознайомлення з офіційними інструкціями

від виробника. Крім цього, користувачам звичайно доступний і самий сервіс для бронювання цих ліків.

Згідно з [4] на кінець 2018 року користувачі Tabletki.ua мають можливість бронювати біля 70% усіх доступних там ліків. Це пов'язано з тим, що самі аптеки відповідають за кількість ліків, доступних для бронювання. Також вони відповідають за завантаження даних про ліки на сервіс. Портал приділяє велику увагу достовірності й актуальності інформації про фармацевтичні продукцію та вимагає від аптек регулярно надавати останні дані про свій асортимент [4, 5].

Самий проект Tabletki.ua був заснований у 2008 році у вигляді сайту, який надавав користувачам інформацію про ціни на ліки. Від початку свого заснування й до сьогодні цей проект активно й динамічно розвивається [3, 4]. Таким чином, він став одним із перших на фармацевтичному ринку, хто запустив онлайн-сервіси для бронювання ліків.

За даними ресурсу SimilarWeb [6] протягом січня 2021 року на веб-сервісі Tabletki.ua було здійснено біля 5,65 млн відвідувань. Самі представники Tabletki.ua заявляють, що протягом 2019 року щомісячна кількість бронювань ліків зростає з 500 тис. у січні до 1 млн в серпні [3, 5]. Також Tabletki.ua на своєму офіційному сайті [5] вказує, що до веб-сервісів бронювання й пошуку ліків підключено понад 5500 аптек по всій Україні.

Іншим подібним веб-сервісом є GeoArteka [7]. Цей веб-сервіс надає користувачам послуги перегляду актуальної інформації про наявність, вартість і місця продажу лікарських засобів, а також надає послуги їх бронювання. За допомогою GeoArteka користувачі отримують можливість знайти по карті найближчі до себе аптеки, в яких вони можуть купити необхідні ліки, переглянути графік роботи цих аптек та оформити бронювання ліків у них [7, 8]. За інформацією в [3] станом на 2019 рік у GeoArteka з'явилися засоби для підтримки урядової програми «Доступні ліки». Ці засоби допомагають користувачам набагато легше знаходити ліки, які підлягають відшкодуванню за

цією програмою, а також аптеки, що можуть у рамках цієї програми безкоштовно надавати певні ліки своїм клієнтам.

Самий веб-сервіс синхронізований з інформаційною системою «Аптека»: до GeoApteka можуть підключатися тільки ті аптеки, які є клієнтами цієї інформаційної системи [8]. Для залучення більшої кількості аптек керівники GeoApteka обіцяють майбутнім клієнтам збільшення середнього чеку покупця приблизно в чотири рази і збільшення обсягу продажів 12-20% [7, 8].

Також, крім окремих онлайн-порталів і сервісів для бронювання ліків, ми розглянемо й великі мережі аптек, для яких створені власні сервіси. На українському фармацевтичному ринку однією з найбільших аптечних мереж є «Аптека Доброго Дня» [9]. Мережа «Аптека Доброго Дня» – це мережа національного рівня, до якої належать понад 220 закладів, розташованих по всій території України.

Для підтримки постійних клієнтів у мережі розроблена Програма Лояльності, що дозволяє користувачам накопичувати бонуси й розраховуватися ними під час наступних покупок. Крім цього, «Аптека Доброго Дня» також є одним з партнерів програми Fishka, за допомогою якої клієнти можуть розраховуватися бонусами, отриманими при покупках товарів та послуг партнерів. Мережа «Аптека Доброго Дня» часто бере участь у програмах виробників: «Про-прайс», «Медікард», Orange Card. Ці програми допомагають людям з хронічними захворюваннями отримувати високі знижки на покупку спеціальних ліків [9].

Сама аптечна мережа, крім власного веб-сайту з усіма послугами, також розробила й додаток для мобільних пристроїв, які працюють під операційними системами Android та iOS. Так само, як і веб-сайт, мобільний додаток дає можливість здійснювати пошук потрібних ліків зі всього асортименту, визначати найближчі аптеки мережі, переглядати їх графік роботи, здійснювати онлайн-бронювання ліків. До інших функціональних можливостей цього додатку належить підтримка віртуальних бонусних карки та сервіс для нагадування про

прийом ліків. За допомогою створення особистих кабінетів користувачі можуть переглядати свою історію покупок і бронювань, отримувати знижки та персональні пропозиції на основі переглянутих товарів [10].

Іншою мережею аптек, що надає власні веб-сервіси для бронювання ліків, є мережа «1 Соціальна Аптека» [11], яка є брендом компанії «Фармастор». Цій компанії також належить мережа «Аптека Доброго Дня». Перші аптеки мережі «1 Соціальна Аптека» відкрилися в кінці 2015 року. На сьогодні в мережі числяться понад 150 аптек як у містах-мільйонниках, так і районах [11, 12]. Таким чином, «1 Соціальна Аптека» можна назвати однією з мереж, які найбільш динамічно розвиваються на українському ринку.

Мережа аптек позиціонує себе як продавців ліків та інших фармацевтичних товарів, які намагаються реалізовувати свої товари з якомога найменшою націнкою до закупівельної ціни. Для цього мережа постійно розробляє проекти з громадськими організаціями та органами влади, що мають на меті створення сприятливих умов для купівлі лікарських засобів вразливими верствами населення.

З 1 грудня 2016 року для користувачів мережі «1 Соціальна Аптека» доступний сервіс для бронювання ліків, яким можна користуватися через офіційний веб-сайт мережі. Цей сервіс надає можливість здійснювати пошук необхідних лікарських засобів та бронювати їх. Під час оформлення бронювання сервіс дозволяє вибрати місто для отримання замовлення та зі списку аптек у місті вибрати ту, за якою й буде зареєстроване бронювання та в якій користувач зможе отримати ліки [11].

### **1.1.3 Постановка завдання**

На основі розглянутих сервісів визначимо завдання щодо розробки веб-сервісу для бронювання ліків. Після аналізу деяких джерел можна сформулювати саме завдання розробки та визначити умови, за яких використовуватиметься розроблюваний веб-сервіс.

Якщо враховувати інформацію з [4], то для великих онлайн-порталів, таких, як Tabletki.ua, для підключення до сервісів пошуку й бронювання ліків більш популярними є окремі одиночні аптеки. Це пов'язано з вартістю підключення до сервісів однієї окремої аптеки. Так, для аптечних мереж вартість підключення всіх своїх аптек є занадто великою й натомість вони замовляють розробку власних веб-сервісів для бронювання ліків.

Великі аптечні мережі, наприклад «Аптека Доброго Дня», які розповсюджені в містах-мільйонниках й обласних центрах України та представлені на фармацевтичному ринку на національному рівні можуть собі дозволити розробку власних веб-сервісів для бронювання ліків. Проте регіональні мережі, які не мають такого територіального охоплення, не бачать переваг у розробці власних сервісів. У цьому й полягає особливість використання розроблюваного веб-сервісу для бронювання ліків. Оскільки розроблюваний веб-сервіс міститиме основні функції для його роботи, то його розробка та підтримка вимагатиме менше витрат, що буде доступним для аптечних мереж, таких, як, наприклад «Ромашка» або «Сальве», розміщених у невеликих регіональних містах. З цим пов'язана й інша особливість веб-сервісу, який розробляється.

Завдяки тому, що мережі регіональних аптек є досить поширеними в невеликих містах і районних центрах, то за допомогою підключення цих аптек до розроблюваного веб-сервісу люди в цих невеликих містах отримають змогу бронювати та забирати необхідні ліки в ближніх до себе аптеках. Особливо це є важливим, коли людям потрібні специфічні ліки, які можна придбати лише в аптеках, розміщених у великих містах.

Таким чином, метою роботи є розробка веб-сервісу для бронювання ліків, який містить основний функціонал та призначений для використання в невеликій регіональній мережі аптек. Переваги розроблюваного веб-сервісу полягають у збільшенні конкурентоспроможності невеликої регіональної аптечної мереж завдяки її виходу на ринок електронної комерції й наданні жителям невеликих міст доступу до купівлі необхідних ліків у межах свого міста.

## 1.2 Формування вимог до веб-сервісу для бронювання ліків

Після аналізу предметної області перейдемо до формування вимог до розроблюваного веб-сервісу для бронювання ліків.

У загальному вимоги до програмного забезпечення можна поділити на дві групи: функціональні та нефункціональні [13, 14].

Функціональні вимоги – це вимоги, які пов’язані з функціонуванням і поведінкою самого ПЗ. Вони задаються користувачем ПЗ та визначають усю функціональність усередині програмної системи [13]. Серед функціональних вимог до розроблюваного веб-сервісу для бронювання ліків визначимо наступні:

- можливість пошуку користувачем ліків за їхньою назвою чи категорією;
- можливість додавання користувачем вибраних ліків до списку бронювань;
- надання можливості перегляду цін та зображень ліків;
- забезпечення можливості вибору кількості одиниць упакувань ліків при оформленні бронювання;
- можливість оформлення й надсилання списку бронювань користувачем із вказанням контактних даних;
- надання можливості для адміністратора веб-сервісу або менеджера аптеки за допомогою адміністраторської панелі (адмін-панелі) отримання доступу до записів, розміщених у базі даних;
- надання можливості для адміністратора або менеджера виконувати CRUD-операції щодо записів у базі даних;
- можливість для адміністратора або менеджера вибору записів для виконання операцій з ними;
- обмеження доступу для адміністратора або менеджера до записів з оформленими бронюваннями: серед операцій доступний лише перегляд.



До іншої групи вимог належать нефункціональні вимоги. Нефункціональні вимоги – вимоги, не пов’язані з функціональним аспектом ПЗ. Вони є неявними характеристиками ПЗ, які користувачі припускають. Зазвичай нефункціональні вимоги – це обмеження якості, яким повинна відповідати програмна система [13, 14].

Сформуємо нефункціональні вимоги до розроблюваного веб-сервісу для бронювання ліків:

- вимоги до графічного інтерфейсу користувача: ліки повинні відображатися у вигляді окремих блоків, які містять інформацію про ліки та їх зображення;
- розділення структури веб-сервісу на дві частини: для взаємодії з користувачем і для забезпечення роботи адміністратора веб-сервісу або менеджера аптеки в адмін-панелі;
- забезпечення введення паролю при авторизації користувача в адмін-панелі;
- реалізація реєстрації користувача в адмін-панелі;
- забезпечення зберігання записів про користувачів адмін-панелі в базі даних, у яких пароль зберігається у вигляді хеш-значення, отриманого за допомогою криптографічного алгоритму хешування;
- забезпечення того, щоб дані, які вводяться користувачем, перевірялися на відповідність до певних комбінацій символів за допомогою регулярних виразів;
- забезпечення зберігання файлів усіх зображень за допомогою технології CDN.

Звернемо увагу, що подані списки функціональних і нефункціональних вимог до розроблюваного веб-сервісу для бронювання ліків не є кінцевими. У ході подальшої розробки, проектування та підтримки веб-сервісу перелік вимог до нього може доповнюватися й розширюватися.

### 1.3 Визначення зовнішніх сутностей і варіантів використання щодо веб-сервісу для бронювання ліків

Для більш точного опису функціональних вимог до розроблюваного веб-сервісу для бронювання ліків та специфікації його функціональної поведінки здійснимо визначення зовнішніх сутностей і варіантів використання веб-сервісу та побудову відповідної діаграми варіантів використання.

Визначення зовнішніх сутностей та варіантів використання дозволяє зрозуміти способи взаємодії зовнішніх користувачів з модельованою системою. У загальному, суть моделювання варіантів використання полягає в тому, що, надати можливість моделювати й розробляти систему з точки зору самого користувача [15].

Діаграма варіантів використання (англ. use case diagram) призначена якраз для візуального подання взаємодії користувача з системою, взаємозв'язків між різними варіантами використання й користувачами без відображення зайвих деталей [15, 16]. Сама діаграма складається з таких компонентів:

- зовнішні сутності (актори);
- варіанти використання (прецеденти);
- зв'язки між варіантами використання та між зовнішніми сутностями;
- межі системи.

У ході розгляду компонентів діаграми варіантів використання ми разом з тим будемо описувати компоненти, які стосуються розроблюваного веб-сервісу для бронювання ліків. Першим компонентом, який ми розглянемо, буде зовнішня сутність.

Зовнішня сутність (актор, англ. actor) – це будь-який об'єкт, суб'єкт або система, що ззовні взаємодіє з модельованою системою. Ця взаємодія полягає у виконанні варіантів використання, необхідних зовнішній сутності для досягнення своїх цілей. Для виконання певного варіанту використання системи зовнішня сутність здійснює надсилання відповідних повідомлень [16].

Для розроблюваного веб-сервісу для бронювання ліків визначимо зовнішні сутності "Користувач" та "Адміністратор".

Зовнішня сутність "Користувач" представляє звичайного користувача веб-сервісу, який використовує його для пошуку й бронювання потрібних ліків. Зовнішня сутність "Адміністратор" відображає адміністратора веб-сервісу або менеджера аптеки, який за допомогою нього має доступ до основних операцій із записами в базі даних.

Для подання опису функціональної поведінки розроблюваного веб-сервісу побудуємо дві діаграми варіантів використання: для зовнішніх сутностей "Користувач" і "Адміністратор". Діаграма варіантів використання для зовнішньої сутності "Користувач" подана на рисунку 1.1. Під час опису цієї діаграми ми водночас розглянемо інші компоненти діаграми: варіанти використання, зв'язки та межі системи.

Варіант використання (прецедент, англ. use case) відображає деяку частину функціоналу, який може виконувати зовнішня сутність при взаємодії з модельованою системою. Варіант використання призначений для специфікації загального функціонування системи й не описує її внутрішньої структури [16].



Рисунок 1.1 – Діаграма варіантів використання для зовнішньої сутності "Користувач"

Діаграма на рисунку 1.1 містить сім варіантів використання, які може виконувати "Користувач" при взаємодії з веб-сервісом для бронювання ліків. Усі варіанти використання розміщені в межах розроблюваної системи.

Межі системи на діаграмі варіантів використання позначають саму систему в цілому та використовуються для відокремлення зовнішніх сутностей і варіантів використання системи, які з пов'язані [15]. У нашому випадку системою є веб-сервіс для бронювання ліків.

На діаграмі подано, що зовнішня сутність "Користувач" при взаємодії з розробленим веб-сервісом перебуває у зв'язку лише з чотирма варіантами використання. Таким зв'язком є відношення асоціації. Відношення асоціації (англ. association relationship) – один з типів відношень на діаграмі варіантів використання, який описує взаємодію зовнішньої сутності з певним варіантом використання та вказує на те, що ця зовнішня сутність ініціює його [16].

Таким чином, "Користувач" під час взаємодії з веб-сервісом може переглядати ліки, додати ліки до списку бронювань, визначити кількість одиниць ліків для бронювання та здійснити надсилання списку бронювань. Примітка на діаграмі вказує, що "Користувач" при надсиланні списку бронювань також виконує надсилання своїх контактних даних. Як ми можемо побачити на діаграмі, решта варіантів використання перебувають у відношеннях з цими варіантами використання.

Так, наприклад, варіант використання "Визначити кількість одиниць ліків для бронювання" перебуває з "Визначити вартість бронювань" у відношенні включення. Відношення включення (англ. include relationship) – тип відношення між варіантами використання, який вказує на те, що залежний варіант використання виконується в послідовності поведінки незалежного варіанту використання [15, 16]. Відношення включення між цими двома варіантами використання означає, що під час визначення кількості одиниць ліків для бронювання зовнішня сутність "Користувач" водночас здійснює визначення вартості всіх бронювань.

Також на діаграмі показано, що варіант використання "Переглядати ліки" пов'язаний з варіантами використання "Здійснити пошук ліків" та "Сортувати ліки за різними параметрами" за допомогою відношення розширення. Відношення розширення (англ. extend relation) відображає зв'язок між двома варіантами використання, у якому один варіант використання задає свою поведінку в поведінці іншого лише за виконання певних умов [16]. Таким чином, при виконанні варіанту використання "Переглядати ліки" можливе також здійснення пошуку ліків і сортування ліків за різними параметрами. Такі можливості були визначені у функціональних вимогах до веб-сервісу.

Після опису діаграми варіантів використання розроблюваного веб-сервісу для зовнішньої сутності "Користувач" перейдемо до визначення варіантів використання для зовнішньої сутності "Адміністратор" і розгляду відповідної діаграми. Цю діаграму зображено на рисунку 1.2.

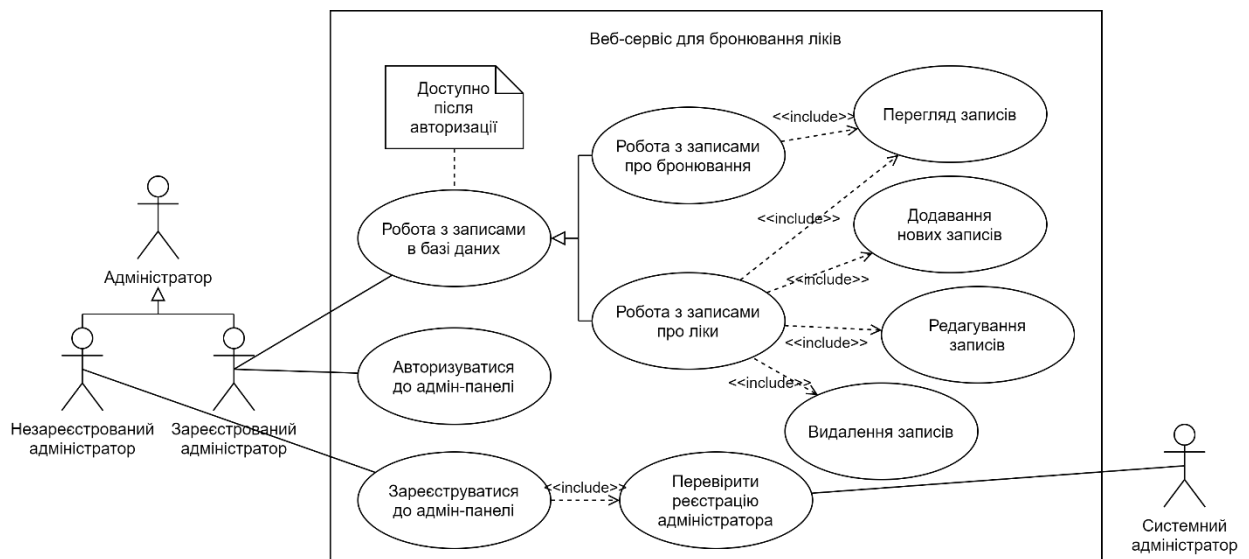


Рисунок 1.2 – Діаграма варіантів використання для зовнішньої сутності "Адміністратор"

На діаграмі з рисунку 1.2 ми бачимо, що зовнішня сутність "Адміністратор" зв'язана відношеннями узагальнення з двома іншими сутностями: "Незареєстрований адміністратор" та "Зареєстрований адміністратор". Відношення узагальнення (англ. generalization relationship) – тип зв'язку між зовнішніми су-

тностями або між варіантами використання, коли об'єкти мають спільні властивості та є окремими випадками загального об'єкта. Таким чином, відношення показує зв'язки типу «батько-нащадок» між об'єктами [15, 16]. Для нашого випадку "Адміністратор", що представляє адміністратора й менеджера аптеки, є загальним випадком для "Незареєстрованого адміністратора" й "Зареєстрованого адміністратора".

На діаграмі в межах веб-сервісу для бронювання знаходяться одинадцять варіантів використання. Для окремих випадків "Адміністратора" асоційовано три варіанти використання.

Так "Незареєстрований адміністратор" при взаємодії з розроблюваним веб-сервісом має можливість лише зареєструватися до адмін-панелі. А поведінка цього варіанту використання включає в себе вже поведінку варіанту використання "Перевірити реєстрацію користувача", який виконується іншою зовнішньою сутністю "Системним адміністратором". Якщо коротко, то це означає, що під час реєстрації незареєстрованого адміністратора до адмін-панелі системний адміністратор виконує перевірку цієї реєстрації.

Для "Зареєстрованого адміністратора" в межах розроблюваного веб-сервісу доступні варіанти використання "Авторизуватися до адмін-панелі" та "Робота з записами в базі даних". Примітка, яка прив'язана до варіанту використання "Робота з записами в базі даних" інформує про те, що веб-сервіс надає доступ до роботи з даними для адміністратора тільки після його авторизації до адмін-панелі.

На діаграмі показано, що варіант використання "Робота з записами в базі даних" є узагальненням варіантів використання "Робота з записами про бронювання" й "Робота з записами про ліки". Таким чином, при здійсненні роботи з записами в базі даних для адміністратора є доступною робота з записами про бронювання і з записами про ліки.

Самий варіант використання "Робота з записами про бронювання" включає у свою поведінку "Перегляд записів". Це означає, що для роботи з записа-

ми про бронювання веб-сервіс дозволяє "Адміністратору" здійснювати лише перегляд цих записів, що й було описано у функціональних вимогах. Натомість, відношення включення, у яких перебуває варіант використання "Робота з записами про ліки", показують, що цей варіант використання містить у собі функціональну поведінку, яка надає можливість виконувати чотири основні CRUD-операції з записами про ліки: "Перегляд записів", "Додавання нових записів", "Редагування записів" та "Видалення записів".

Після того, як ми визначили, які зовнішні сутності взаємодіють з розробленим веб-сервісом для бронювання ліків і які варіанти використання він їм надає, ми можемо перейти до опису архітектури веб-сервісу та визначення його базової структури.

#### **1.4 Проектування архітектури веб-сервісу для бронювання ліків**

Вибір архітектурної моделі для розробленого веб-сервісу допомагає визначити концепцію структури веб-сервісу та описати зв'язки між його структурними елементами. При виборі архітектурної моделі зупинимося на архітектурі «клієнт-сервер».

Архітектура «клієнт-сервер» є найпопулярнішою концепцією для створення програмних систем у розподілених мережах, таких, як Інтернет. Ця архітектурна модель передбачає розділення системи на два типи компонентів [17]:

- сервери, що здійснюють надання послуг;
- клієнти, які використовують надані послуги.

Таким чином, архітектура «клієнт-сервер» розділяє завдання між серверами та клієнтами. Клієнти виконують ініціацію запитів до серверів на певні послуги, а сервери здійснюють надання цих послуг. Взаємодія між клієнтами та серверами здійснюється за допомогою мережі [18].

До основних переваг цієї архітектурної моделі можна віднести:

- доступ до послуг у будь-який час, у будь-якому місці;

- платформи-незалежність;
- покращена масштабованість;
- високий рівень продуктивності за рахунок розподілення системи;
- незалежність вузлів, яка полегшує їх заміну чи переміщення.

Одним з прикладів популярності моделі «клієнт-сервер» є її використання в мережі всесвітнього павутиння [18, 19]. Саме на основі цього та на основі переваг для проектування структури розроблюваного веб-сервісу для бронювання ліків ми вибираємо архітектуру «клієнт-сервер».

За функціональністю архітектура «клієнт-сервер» може мати декілька різних рівнів. Залежно від їх кількості та призначення існують наступні типи цієї моделі: однорівнева, дворівнева, трирівнева, N-рівнева архітектури [19].

Для нашого веб-сервісу для бронювання ліків використаємо найбільш поширений тип архітектури «клієнт-сервер» – трирівневу архітектуру (англ. three-tier architecture), який розділяє програмну систему на три рівні [19, 20]:

- рівень представлення (англ. presentation tier) – містить інтерфейс користувача, призначений для взаємодії кінцевого користувача з системою; відповідає за відображення інформації й збирання даних від користувача;
- прикладний рівень (англ. application tier) – серце програмної системи; здійснює обробку даних, зібраних на рівні представлення, на основі бізнес-логіки; виконує операції з даними за допомогою взаємодії з рівнем даних;
- рівень даних (англ. data tier) – відповідає за доступ до даних; здійснює збереження та управління даними, які обробляються в програмній системі.

Таким чином, головною перевагою трирівневої архітектури є фізичне й логічне розділення функціоналу між рівнями системи. Це дає можливість виконувати роботу на різних платформах, здійснювати незалежне оновлення та масштабування. Також це сприяє покращенню безпеки за рахунок реалізації взаємодії між рівнями представлення й даних через прикладний рівень [20].

Тепер опишемо структуру розроблюваного веб-сервісу для бронювання ліків. Структурна схема подана на рисунку 1.3.



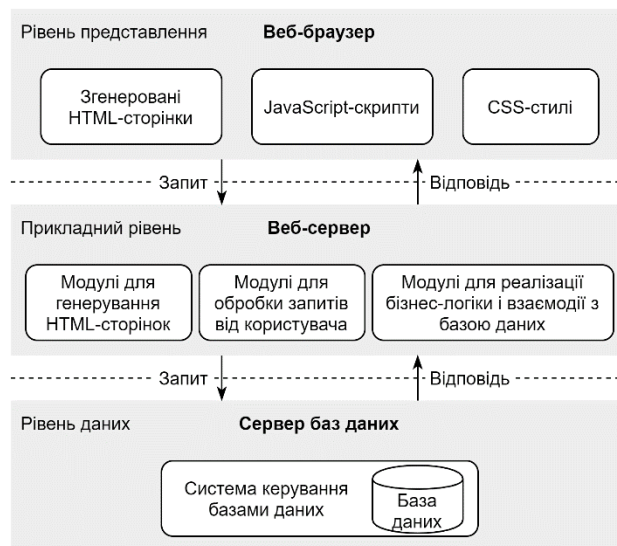


Рисунок 1.3 – Схема веб-сервісу для бронювання ліків

З рисунку ми бачимо, що на рівні представлення розроблюваного веб-сервісу перебуває веб-браузер, який відображає інформацію для кінцевого користувача за допомогою найпопулярніших засобів: HTML-сторінок, CSS-стилів і JavaScript-скриптів. JavaScript-скрипти також ініціюють запити для передачі даних від користувача для їх обробки на прикладному рівні.

Прикладним рівнем є веб-сервер, у якому містяться модулі для генерування сторінок та обробки запитів, отриманих від рівня представлення. Крім цього на цьому рівні знаходяться модулі, що взаємодіють з базою даних і виконують усю бізнес-логіку веб-сервісу: отримання переліку ліків, надсилання бронювань, отримання записів про ліки й бронювання тощо.

За рівень даних відповідає сервер баз даних, на якому працює СКБД, яка надає доступ до бази даних веб-сервісу для бронювання ліків. СКБД обробляє запитів, отримані від прикладного рівня, виконує операції над даними та повертає до прикладного рівня результати виконання їх виконання [20].

Під час проектування веб-сервісу ми логічно об'єднуємо прикладний рівень і рівень даних, так що вони утворюють сервер у моделі «клієнт-сервер», який приймає запити від клієнта та виконує бізнес-логіку. Рівень представлення, таким чином, відповідає за клієнта, що отримує дані від користувача й надсилає запити до сервера для реалізації функціоналу [17, 19].

## 2 РОЗРОБКА, РОЗГОРТАННЯ ТА ВИКОРИСТАННЯ ВЕБ-СЕРВІСУ ДЛЯ БРОНЮВАННЯ ЛІКІВ

### 2.1 Обґрунтування вибору засобів розробки веб-сервісу для бронювання ліків

У ході розробки веб-сервісу для бронювання ліків ключове значення має вибір засобів розробки. Для розробки веб-сервісу, визначеного під час постановки завдання, необхідно здійснити вибір засобів для функціонування на кожному рівні: для реалізації інтерфейсу користувача, виконання бізнес-логіки і зберігання даних.

Нижче наведено перелік засобів, які були вибрані для розробки веб-сервісу для бронювання ліків:

- PHP версії 7.1 – для реалізації бізнес-логіки і створення інтерфейсу користувача;
- Bootstrap версії 4.5 – для побудови елементів графічного інтерфейсу;
- JavaScript з jQuery версії 3.5 – для реалізації взаємодії користувача з графічним інтерфейсом;
- MySQL версії 5.6 – для зберігання даних й роботи з базою даних.

Розглянемо особливості та переваги цих засобів, що вплинули на їх вибір під час розробки веб-сервісу для бронювання ліків.

Так, PHP – це одна з найпоширеніших скриптових мов програмування з відкритим вихідним кодом. Основним способом використання PHP є запуск скриптів на веб-сервері і їх обробка інтерпретатором для генерування HTML-сторінок, які можна переглядати у веб-браузері.

До причин популярності мови PHP у веб-розробці належить простота її освоєння: вона спеціально спроектована, щоб її код вбудовуватися в HTML-розмітку й оброблятися інтерпретатором для генерування сторінки [21].

Іншою перевагою PHP підтримка багатьох операційних систем (Linux, Windows, macOS, Solaris, FreeBSD) і сучасних веб-серверів (Apache, Internet Information Services (IIS), Nginx). Ще однією вагомою перевагою PHP є простота роботи з базами даних і підтримка широкого вибору СКБД. Так, бібліотеки для підключення й роботи з БД підтримуються як з основні СКБД (SQL Server, MySQL, Oracle), так і NoSQL-СКБД, як MongoDB [22, 23].

Таким чином, з урахування популярності та наведених переваг ми вибрали PHP для реалізації бізнес-логіки розроблюваного веб-сервісу та створення інтерфейсу користувача за допомогою генерування HTML-сторінок.

Водночас для реалізації інтерфейсу користувача веб-сервісу для бронювання ліків використовуємо Bootstrap – один з популярних безкоштовних фреймворків для простого і швидкого графічного оформлення веб-сайтів. Фреймворк складається HTML- та CSS-шаблонів, які надають інструментарій для побудови й оформлення графічних елементів інтерфейсу користувача: кнопок, навігаційних панелей, форм, блоків, таблиць. Bootstrap містить також деякі JS-скрипти, що використовуються для анімації графічних елементів і взаємодії користувача з ними [24].

Простота та наявність великої кількості шаблонів визначили наш вибір фреймворку Bootstrap для реалізації графічних елементів інтерфейсу користувача при розробці веб-сервісу для бронювання ліків.

Для забезпечення взаємодії користувача з інтерфейсом розроблюваного веб-сервісу використовуємо JavaScript, що є найпопулярнішою скриптовою мовою програмування, призначеною для побудови інтерактивних веб-сторінок. На стороні веб-браузера розширення JavaScript дозволяють скриптам здійснювати управління елементами веб-сторінок на основі моделі DOM (Document Object Model) та реагувати на дії користувача: клацання мишею, заповнення форми, навігація сторінкою [25].

Таким чином, для розробки веб-сервісу разом з JavaScript ми будемо використовувати бібліотеку jQuery. jQuery є легкою та швидкою JS-бібліотекою,

яка надає інструменти для спрощеної розробки скриптів з написанням меншої кількості коду. До основних можливостей, які надає jQuery, належать [26]:

- управління елементами веб-сторінки за моделлю DOM через селектори CSS і XPath та зручна реалізація обробки подій від користувача;
- вбудовані візуальні ефекти й анімації;
- підтримка взаємодії веб-браузера з сервером на основі AJAX;
- крос-браузерна підтримка JS-скриптів.

Зважаючи на популярність, широкий набір інструментів і простоту, для реалізації взаємодії користувача з інтерфейсом розроблюваного веб-сервісу ми вибрали скриптову мову JavaScript та її бібліотеку jQuery.

При розробці веб-сервісу для бронювання ліків у якості середовища для зберігання даних і здійснення доступу до них ми використовуємо систему керування реляційними базами даних (СКРБД) MySQL, яка розробляється й підтримується компанією Oracle та є однією з найпопулярніших СКБД. Таким чином, система MySQL стала одним з невід’ємних компонентів для зберігання даних для веб-розробки [27].

До переваг MySQL належать: висока продуктивність і великий набір функціональних можливостей, використання стандартної мови SQL для роботи з БД, підтримка багатопотоковості, висока масштабованість та можливість використання великих БД.

Крім цього, до сильних сторін системи MySQL є підтримка великої кількості платформ й операційних систем. Так, сервери MySQL можуть запускатися в середовищах macOS, Linux, Windows, Solaris, FreeBSD тощо. Ще однією сильною стороною MySQL є широкий вибір мов програмування, у яких наявний інтерфейс для доступу до баз даних MySQL. До таких мов належать C, C++, C#, PHP, Java, Perl, Python, Ruby [27, 28].

Таким чином, для вибору MySQL у якості СКБД при розробці веб-сервісу для бронювання ліків значно вплинули популярність у веб-розробці й підтримка інтерфейсів у багатьох мовах програмування.

## 2.2 Розробка моделі даних веб-сервісу для бронювання ліків

При виборі СКБД MySQL для функціонування на рівні даних в архітектурі розроблюваного веб-сервісу ми визначили реляційну БД для зберігання даних. Таким чином, для концептуального опису та розробки бази даних на основі предметної області найкращим варіантом є використання моделі «сутність-зв'язок» (англ. entity-relationship). Модель «сутність-зв'язок» призначена для опису структури БД. Основними складовими ER-моделі є типи сутності, їхні атрибути та зв'язки між ними [29].

На рисунку 2.1 зображена діаграма, яка використовується для візуального подання ER-моделі бази даних розроблюваного веб-сервісу. Під час опису цієї діаграми й розглянемо основні складові ER-моделі.

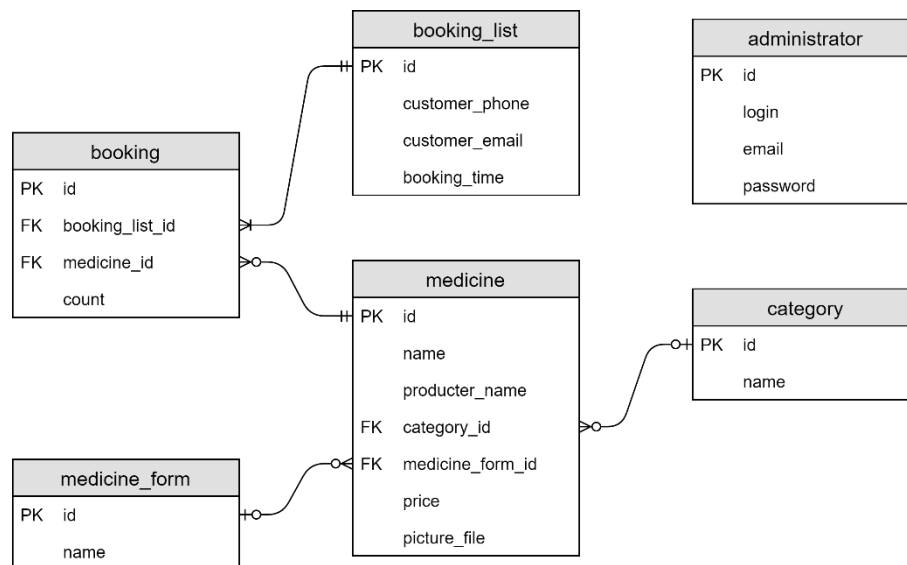


Рисунок 2.1 – ER-діаграма бази даних веб-сервісу для бронювання ліків

Тип сутності (англ. entity type) – набір однорідних об'єктів предметної області, для якої розробляється база даних [30]. На діаграмі з рисунку 2.1 типи сутності відображаються у вигляді прямокутників. У цій ER-моделі ми виділили шість наступних типів сутності: medicine, category, medicine\_form, booking, booking\_list і administrator.

Кожен з цих типів сутності має свій перелік атрибутів (англ. attribute). Атрибути використовуються для опису властивостей типу сутності. Вони характеризують конкретний тип сутності. На ER-діаграмі (рисунок 2.1) атрибути зображуються списками у прямокутниках типів сутності. Виділяють також два особливі види атрибутів: первинні й зовнішні ключі. Первинний ключ (англ. primary key, PK) може однозначно ідентифікувати конкретний екземпляр типу сутності. Зовнішній ключ (англ. foreign key, FK) використовується для реалізації зв'язку між двома типами сутності. Він позначає конкретний екземпляр типу сутності, який має зв'язок з екземпляром іншого типу сутності [30, 31].

Зв'язок (англ. relationship) визначає взаємодію між двома типами сутності. На діаграмі на рисунку 2.1 зв'язки позначаються у вигляді ліній, що з'єднують типи сутності. Для позначення зв'язків на цій діаграмі використовується нотація «вороняча лапка» (англ. crow's foot).

Залежно від виду зв'язку виділяють слабкий тип сутності, екземпляри якого через зовнішні ключі залежать від інших типів сутності. Сильні типи сутності, натомість, не залежать від інших типів сутності та їх екземпляри можуть однозначно визначатися без використання зв'язків [29, 30].

Види зв'язку між типами сутності в ER-моделі визначаються його кардинальністю. Кардинальність вказує можливу кількість екземплярів одного типу сутності, що відповідають екземпляру іншого типу сутності, який перебуває у зв'язку [31]. Кардинальності зв'язків розберемо під час опису типів сутностей ER-моделі бази даних розроблюваного веб-сервісу для бронювання ліків.

Розглянемо тип сутності medicine. Цей тип сутності представляє ліки в нашій базі даних. Відповідно до діаграми на рисунку 2.1 він містить наступні атрибути:

- id – ідентифікатор (первинний ключ);
- name – назва ліків;
- producter\_name – назва виробника;
- category\_id – ідентифікатор категорії (зовнішній ключ);

- `medicine_form_id` – ідентифікатор форми пакування (зовнішній ключ);
- `price` – ціна;
- `picture_file` – посилання, за яким зберігається файл з зображенням.

Тип сутності `medicine` має два зовнішні ключі. Таким чином, він є слабким. Ці ключі визначають залежність `medicine` від типів сутності `category` та `medicine_form`. Типи сутності `category` і `medicine_form` використовуються для реалізації механізму категоріальних атрибутів `medicine` категорії ліків і форми пакування. Так, `category` відображає категорію ліків, а `medicine_form` – відображає форму пакування ліків. Обидва типи сутності мають однакові атрибути:

- `id` – ідентифікатор (первинний ключ);
- `name` – назва (категорії або форми пакування).

Оскільки серед перелічених атрибутів немає зовнішніх ключів, то типи сутності `category` та `medicine_form` є сильними. Відносно типу сутності `medicine` він перебуває у зв'язку "Належить до" або "Має" з типами сутності `category` та `medicine_form`.

Кардинальність зв'язку – «нуль або багато до нуля або одного». Це означає, що нулю або кільком екземплярам `medicine` відповідає нуль або один екземпляр типів сутностей `category` і `medicine_form`. Таким чином один екземпляр ліків може належати до однієї категорії або не належати до жодної категорії, яка існує, а також він може мати одну форму пакування або не мати жодної форми пакування з тих, які існують у базі даних [30, 31].

Розберемо тепер тип сутності `booking`. Він використовується для представлення одного бронювання ліків. За ER-діаграмою до переліку його атрибутів входять:

- `id` – ідентифікатор (первинний ключ);
- `booking_list_id` – ідентифікатор списку бронювань (зовнішній ключ);
- `medicine_id` – ідентифікатор ліків (зовнішній ключ);
- `count` – кількість одиниць ліків.

Тип сутності booking має два зовнішні ключі і, таким чином, залежить від двох інших типів сутності. Через це booking є слабким типом сутності.

За допомогою зовнішнього ключа medicine\_id тип сутності booking перебуває у зв'язку "Містить" з типом сутності medicine з кардинальністю «нуль або багато до одного й тільки одного». Так, екземпляр бронювання містить лише один екземпляр ліків, а екземпляр ліків може міститися в кількох бронюваннях або не міститися в жодному бронюванні, що зберігається в БД.

Зовнішній ключ booking\_list\_id використовується для зв'язку "Входить до" між типами сутності booking і booking\_list. Кардинальністю цього зв'язку є «один або багато до одного й тільки одного». Тобто, один екземпляр бронювання може входити до складу тільки одного списку бронювань (booking\_list), а кожен список бронювань може складатися з одного або кількох екземплярів бронювання [31].

Як ми вже вказали, тип сутності booking\_list використовується для представлення списку бронювань. Нижче подано список його атрибутів:

- id – ідентифікатор (первинний ключ);
- customer\_phone – номер мобільного телефону користувача;
- customer\_email – E-Mail адреса користувача;
- booking\_time – час і дата створення списку бронювань.

Звідси ми бачимо, що серед атрибутів booking\_list немає зовнішніх ключів. Це вказує на те, що, booking\_list є сильним типом сутності.

Тип сутності administrator призначений для зберігання записів про адміністраторів під час їх авторизації до адмін-панелі. Тому цей тип сутності не має зв'язків з іншими типами сутностей і, таким чином, є сильним. До атрибутів administrator належать дані, необхідні для реєстрації й авторизації адміністратора:

- id – ідентифікатор (первинний ключ);
- login – логін адміністратора;
- email – E-Mail адреса адміністратора;



- password – хеш-значення паролю адміністратора.

На основі описаної ER-моделі в СКБД MySQL ми за допомогою SQL-запитів створюємо відповідну реляційну базу даних pharm\_booking.

У рамках створеної реляційної бази даних усі розглянуті типи сутності подаються у вигляді таблиць з відповідними назвами, а їх атрибути – у вигляді стовпців відповідних таблиць. Зв'язки між таблицями реалізуються за допомогою обмежень на первинні й зовнішні ключі для деяких стовпців [30, 31].

Назви створеної бази даних обумовлена початковою назвою розроблюваного веб-сервісу для бронювання ліків – «PharmBooking».

### **2.3 Розробка компонентів веб-сервісу для бронювання ліків**

У ході розробки веб-сервісу для бронювання ліків його компоненти можна умовно поділити на дві частини. Одна частина відповідає за функціональні можливості, пов'язані із взаємодією веб-сервісу з користувачем відповідно до розробленої діаграми варіантів використання на рисунку 1.1. Друга частина відповідає за функціонал веб-сервісу при взаємодії з адміністратором за допомогою адмін-панелі. Цей функціонал було описано в діаграмі варіантів використання на рисунку 1.2.

Далі ми дослідимо поведінку компонентів та їх взаємодію між собою під час роботи розроблюваного веб-сервісу для бронювання ліків. Спершу розглянемо частину компонентів, які працюють при взаємодії з користувачем. У файловій структурі розроблюваного веб-сервісу (додаток А) до цих компонентів входять майже всі файли кореневого каталогу. До компонентів, пов'язаних із взаємодією з адміністратором, належать файли до каталогу «admin».

Функціонування компонентів, що відповідають за взаємодію з користувачем, зручно аналізувати в контексті описаної тривірневої архітектури веб-сервісу. На рисунку 2.2 зображено схему розміщення файлів, які використовуються при взаємодії з користувачем, на різних рівнях архітектури.

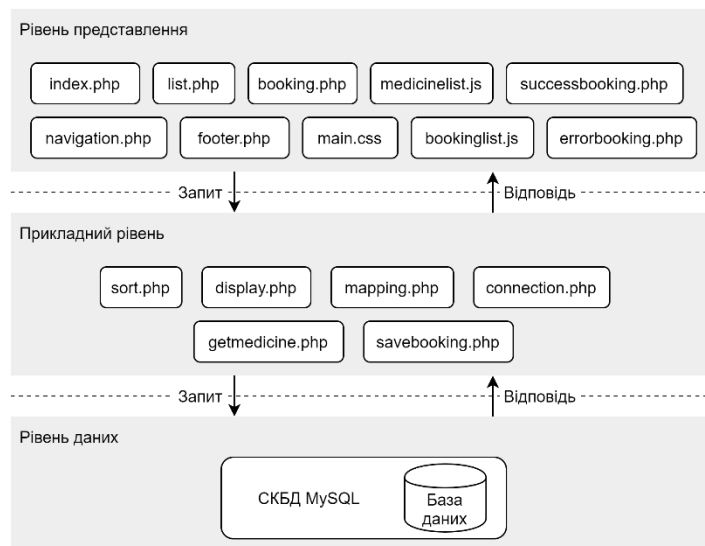


Рисунок 2.2 – Схема файлів, які пов’язані із взаємодією веб-сервісу з користувачем

Як ми розглядали раніше, архітектура розроблюваного веб-сервісу складається з трьох рівнів: рівня представлення, прикладного рівня та рівня даних. Раніше ми зазначали, що рівень даних зберігає й управляє даними. На цьому рівні працює СКБД MySQL, яка містить базу даних веб-сервісу.

Рівень представлення використовується для відображення інтерфейсу користувача та отримання даних від нього. Розберемо, які файли, що пов’язані із взаємодією з користувачем, містяться на цьому рівні. До них належать PHP-файли, які відображають окремі сторінки розроблюваного веб-сервісу для бронювання ліків. Код цих файлів подано в додатку В.

Першим таким файлом є «`index.php`» (лістинг В.1) – індексний файл, що відображає головну сторінку веб-сервісу. Код цього файлу виводить на головну сторінку кнопки, що дають можливість вибрати ліки за категорією. До цього файлу, як і до інших, підключаються файли «`navigation.php`» і «`footer.php`» з каталогу «`templates`», які відображають навігаційну панель і футер усіх сторінок веб-сервісу. У навігаційній панелі розміщується поле для пошуку ліків за назвою, а також посилання на головну сторінку (файл «`index.php`») та сторінку зі списком бронювань (файл «`booking.php`»).

Перегляд ліків за категорією або через їх пошук за назвою реалізується файлом «list.php» (лістинг В.2). Для отримання записів про ліки він використовує файл «getmedicine.php» з каталогу «database». Для відображення ліків у вигляді блоків використовується файл «display.php».

До «list.php» підключається файл «medicinelist.js» з каталогу «scripts», скрипт містить функціонал, що реагує на вибір способу сортування ліків і надсилає AJAX-запити до файлу «sort.php» для отримання відсортованих записів про ліки. У «medicinelist.js» також наявний функціонал для додавання вибраних ліків до списку бронювань шляхом створення cookie у поточній сесії веб-браузеру. Крім цього до «list.php» підключається файл «main.css» з каталогу «styles». Цей CSS-файл задає загальні стилі для всіх сторінок веб-сервісу.

Як зазначалося, файл «booking.php» (лістинг В.3) призначений для відображення списку бронювань та реалізації надсилання цього списку. Для виведення списку бронювань використовується механізм cookie, файли «getmedicine.php» і «display.php».

До «booking.php» також підключається файл «bookinglist.js» з каталогу «scripts», код якого використовується для відображення загальної суми бронювань, видалення ліків зі списку бронювань та перевірки правильності введення користувачем номеру телефону, E-Mail адреси у формі для надсилання списку бронювань. Файли «successbooking.php» та «errorbooking.php», розміщені в каталозі «templates» відображають сторінки з повідомленням про успішність або помилку надсилання списку бронювань.

Перейдемо до розгляду файлів, розміщених на прикладному рівні. На цьому рівні виконується бізнес-логіка веб-сервісу, обробка даних, отриманих від користувача, взаємодія з СКБД.

До таких файлів належить «display.php», який, як вже було вказано, використовується для відображення записів про ліки на сторінках перегляду ліків (файл «list.php») і списку бронювань (файл «booking.php»). Файл «sort.php» використовується для отримання AJAX-запиту від «medicinelist.js» та виведе-

ння на сторінці «list.php» відсортованих записів про ліки у вигляді блоків. Для отримання записів з БД «sort.php» використовує файл «getmedicine.php» з каталогу «database». У файлі «mapping.php» зберігаються асоціативні масиви, які необхідні для роботи багатьох файлів веб-сервісу.

У каталозі «database» (додаток А) зберігаються файли, які можуть взаємодіяти з базою даних в MySQL. Файл «connection.php» (лістинг В.4) містить код для підключення до БД. У файлі «getmedicine.php» (лістинг В.5) зберігається код, призначений для отримання з БД записів про ліки для їх відображення на сторінках перегляду ліків і списку бронювань. Для підключення до БД «getmedicine.php» використовує файл «connection.php».

Файл «savebooking.php» (лістинг В.6) використовується для збереження списку бронювань. Для кращого опису процесу надсилання списку бронювань скористаємося діаграмою діяльності (англ. activity diagram), призначеною для візуалізації послідовності дій, які виконуються у процесах системи [32]. Діаграма діяльності процесу надсилання списку бронювань подана на рисунку Б.1 (додаток Б). Основні елементи цієї діаграми ми розглянемо в ході її розбору.

Діаграма діяльності на рисунку Б.1 складається з чотирьох доріжок. Доріжка (англ. swimlane) відображає область, що обмежує дії, які виконуються одним елементом системи. Так, надсилання та збереження списку бронювань здійснюється за допомогою наступних чотирьох файлів: «booking.php», «savebooking.php», «successbooking.php», «errorbooking.php».

Виконання процесу надсилання списку бронювань починається у початковому стані (англ. initial state), що позначається зафарбованим кружком. Від початкового стану стрілка веде до заокругленого прямокутника.

Стрілками зображуються потоки управління (англ. control flow), які вказують на перехід між діяльностями. У заокруглених прямокутниках записуються діяльності (англ. activity), що виконуються в процесі. Тоді, першою діяльністю є надсилання списку бронювань, яке виконується у файлі «booking.php».

Далі стрілка потоку управління веде до ромба, з якого йдуть два розгалуження. Ромбом на діаграмі позначається рішення (англ. decision) – точка розгалуження потоку управління відповідно до умов. Умови, за яких виконується перехід тим чи іншим шляхом, позначаються біля розгалужень у квадратних дужках. Для процесу надсилання списку бронювань це означає перевірку коректності надісланих даних.

Якщо дані були коректними, то потік управління веде до їх збереження у БД файлом «savebooking.php». Далі здійснюється перехід до файлу «successbooking.php», що відображає повідомлення про успіх надсилання. Виконання процесу надсилання списку бронювань після цього завершується в кінцевому стані (final state), що позначається кружком, зафарбованим у колі.

В іншому випадку, якщо дані є некоректними, здійснюється перехід до файлу «errorbooking.php», що відображає сторінку з повідомленням про помилку. Після цього в кінцевому стані й закінчується саме виконання процесу надсилання списку бронювань [32, 33].

Розглянемо тепер компоненти, що відповідають за взаємодію веб-сервісу з адміністратором, тобто, які відповідають за роботу адмін-панелі. Для цього також використовуємо описану трирівневу архітектуру. Усі файли, пов'язані з адмін-панеллю, містяться в каталозі «admin» (додаток А). Схема розміщення цих файлів на рівнях архітектури розроблюваного веб-сервісу подана на рисунку 2.3. Код файлів адмін-панелі міститься в лістингах додатку Г.

Спочатку розберемо файли, розміщені на рівні представлення. До них належать файли, які відображають окремі сторінки адмін-панелі, файли з CSS-стилями й JS-скриптами.

Робота адмін-панелі починається з файлу «index.php» (лістинг Г.1), розміщеного у каталозі «admin». Цей файл виводить форму для авторизації користувача до адмін-панелі. У цій формі також відображається кнопка для переходу до реєстрації адміністратора. До файлу «index.php», як і до інших файлів,

підключаються файли «navigation.php» і «footer.php» з каталогу «templates», що відображають на сторінках навігаційне меню й футер.

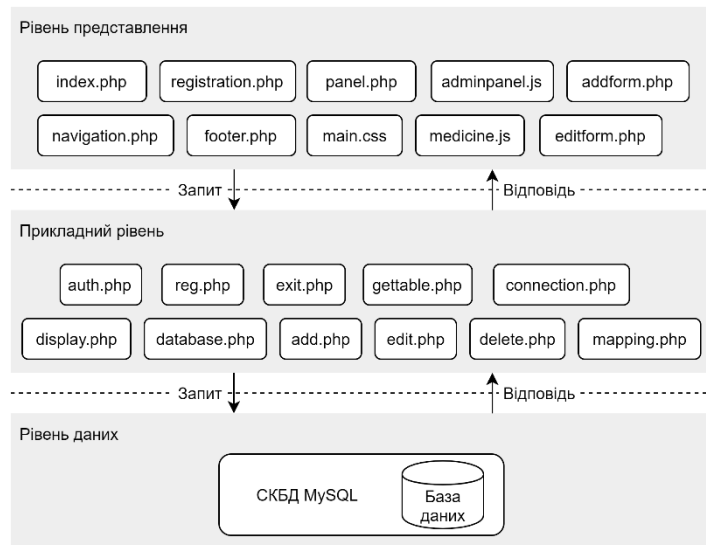


Рисунок 2.3 – Схема файлів, які пов’язані із взаємодією веб-сервісу з адміністратором (адмін-панель)

У «registration.php» відображається сторінка з формою для реєстрації користувача до адмін-панелі. Файл «panel.php» (лістинг Г.2) відображає робочу сторінку адмін-панелі, у якій здійснюється перегляд записів про ліки і про бронювання.

До цього файлу підключається також файл «adminpanel.js» у каталозі «scripts» каталогу «admin» (додаток А). «adminpanel.js» містить JS-скрипт, що обробляє дію вибору користувачем таблиці для перегляду: з записами про ліки або з записами про бронювання. Для виведення потрібної таблиці «adminpanel.js» здійснює надсилання AJAX-запиту до файлу «gettable.php». Крім цього, «adminpanel.js» реалізує також виконання інших CRUD-операцій з записами про ліки: створення, редагування й видалення.

Тоді, файли «addform.php» та «editform.php» у каталозі «medicine» використовуються для відображення форм додавання й редагування запису про ліки відповідно. До обох цих файлів підключається файл «medicine.js» з катало-

гу «scripts». Скрипт цього файлу використовується для перевірки значень, які вводяться у полях форм на сторінках файлів «addform.php» і «editform.php».

Розглянемо файли, що працюють на прикладному рівні. Одним з таких файлів є «auth.php». Він відповідає за авторизацію користувача до адмін-панелі. Авторизація користувача реалізовується шляхом створення відповідного cookie після перевірки наявності запису в базі даних. Для перевірки даних використовується хеш-значення введеного користувачем паролю.

Схожим чином працює файл «reg.php», призначений для реєстрації користувача до адмін-панелі. Він здійснює зберігання нового запису про адміністратора в БД і зберігає відповідний cookie, тим самим авторизуючи щойно зареєстрованого адміністратора до адмін-панелі. Механізм cookie забезпечує авторизовану сесію користувача в адмін-панелі. При встановленні авторизованої сесії будь-який перехід на головну сторінку адмін-панелі перенаправляє до сторінки файлу «panel.php», де виконується робота в адмін-панелі.

За реалізацію закриття авторизованої сесії відповідає файл «exit.php», що запускається після натискання кнопки виходу з адмін-панелі у навігаційному меню. Під час свого виконання файл «exit.php» просто видаляє cookie авторизованого користувача й виконує перехід до головної сторінки адмін-панелі, на якій виводиться форма авторизації.

Файл «gettable.php» (лістинг Г.3) призначений для відображення відповідної таблиці з записами на сторінці файлу «panel.php». Для своєї роботи файл використовує функціонал файлу «display.php» (лістинг Г.4), який розміщений у каталозі «admin».

Для детальнішого розгляду роботи файлу «gettable.php» і виконання процесу вибору таблиці для перегляду використаємо діаграму діяльності, подану на рисунку Б.2 (додаток Б).

Ця діаграма складається з трьох доріжок, які відповідають файлам «panel.php», «adminpanel.js» і «gettable.php». Самий процес вибору таблиці починається з початкового стану у файлі «panel.php», що веде до вибору користува-

чем таблиці, яку він хоче переглянути. Далі потоки управління ведуть до файлу «adminpanel.js», що виконує надсилання AJAX-запиту до «gettable.php» для отримання вибраних записів з БД.

У файлі «gettable.php» йде розгалуження потоків управління, за яким з БД отримуються записи вибраної таблиці: або записи про ліки, або записи про бронювання. Після цього стрілки ведуть до одного ромба, який відображає злиття (англ. merge) потоків управління [32]. Так, після виконання файлу «gettable.php» таблиця з записами назад повертається до «adminpanel.js».

Далі «adminpanel.js» виконує відображення отриманої таблиці на сторінці файлу «panel.php», після чого йде кінцевий стан, у якому процес вибору таблиці для перегляду закінчує своє виконання.

У каталозі «medicine» містяться файли, відповідальні за виконання основних операцій з записами про ліки: додавання, створення та видалення. Файл «database.php» (лістинг Г.5) реалізує виконання перелічених операцій із записами про ліки в БД. Файли «add.php» і «edit.php» використовуються для обробки форм у файлах «addform.php» і «editform.php» та для додавання й редагування записів про ліки в БД. Файл «delete.php» призначений для видалення вибраного запису про ліки з БД.

На цьому закінчується розробка компонентів веб-сервісу для бронювання ліків. Далі ми переходимо до розгортання розробленого веб-сервісу й розміщення його веб-хостингу для подальшого використання

## **2.4 Розгортання веб-сервісу для бронювання ліків**

Розгортання веб-сервісу для бронювання ліків складається з трьох етапів. На першому етапі ми здійснюємо пересилання файлів зображень на CDN-хостинг. Далі, на другому етапі, переносимо файли розробленого веб-сервісу на веб-хостинг. І на останньому, третьому етапі виконуємо перенесення створеної бази даних на веб-хостинг.



Далі ми опишемо процес розгортання розробленого веб-сервісу для бронювання ліків. Зауважимо, що перед розгортанням веб-сервісу для того, щоб наповнити його вмістом, до розробленої бази даних `pharm_booking` додаємо записи про адміністраторів, про ліки, їх категорії та форми пакування.

### 2.4.1 Перенесення зображень на CDN-хостинг

Для зберігання файлів з фотографіями ліків використовуємо розподілену мережу CDN, яка дозволяє швидко доставляти до веб-сторінок статичні ресурси, такі, як CSS-, JavaScript-файли, зображення. У якості CDN-хостингу вибираємо `Sirv` [34], що спеціально призначений для зберігання, обробки й оптимізації відображення зображень на веб-сайтах. Для використання послуг CDN-хостингу `Sirv` створюємо в ньому обліковий запис з безкоштовним тарифом.

Для надсилання файлів зображень на CDN-хостинг будемо використовувати протокол передачі файлів FTP шляхом надсилання файлів на FTP-сервер CDN-хостингу за допомогою FTP-клієнта. Для цього в налаштуваннях `Sirv` знаходимо параметри підключення до FTP-сервера. Це подано на рисунку 2.4.

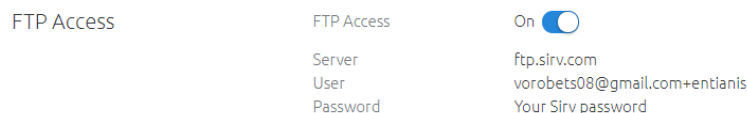


Рисунок 2.4 – Параметри підключення до FTP-сервера `Sirv`

Для пересилання файлів зображень на FTP-сервер CDN-хостингу `Sirv` використовуємо FTP-клієнт з відкритим вихідним кодом `FileZilla`. У графічному інтерфейсі `FileZilla` вводимо отримані параметри і відкриваємо з'єднання з FTP-сервером CDN-хостингу `Sirv`. У файловій системі FTP-сервера створюємо каталог «`pharmbooking`» та пересилаємо до цього всі файли з зображеннями ліків. Після цього закриваємо з'єднання з FTP-сервером.

Далі переходимо у веб-інтерфейс CDN-хостингу `Sirv` і переглядаємо переслані файли з зображеннями ліків. При перегляді файлів з зображеннями ми

можемо отримати URL-посилання, за якими можна отримати доступ до цих зображень. Це можна побачити на рисунку 2.5.

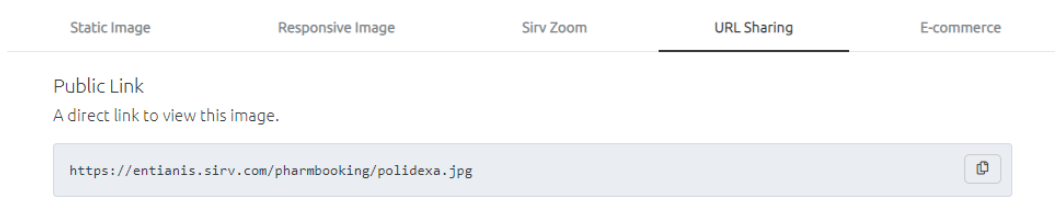


Рисунок 2.5 – Отримання URL-посилань на файли з зображеннями

Отримані значення URL-посилань ми зберігаємо в записах про ліки у створеній базі даних `pharm_booking`. Вони використовуватимуться для виведення зображень ліків під час роботи розробленого веб-сервісу.

#### 2.4.2 Пересилання файлів веб-сервісу на веб-хостинг

Для розміщення та запуску розробленого веб-сервісу на сервері у все-світній мережі використовуватимемо веб-хостинг 000webhost [35]. Вагомою перевагою веб-хостингу 000webhost, яка посприяла його вибору, є наявність безкоштовного тарифу та таких особливостей платних веб-хостингів, як підтримка PHP, MySQL і cPanel.

Таким чином, у веб-хостингу 000webhost ми створюємо обліковий запис і реєструємо веб-сайт за безкоштовним тарифом. Зареєстрований веб-сайт на 000webhost є доступним за адресою `https://pharm-booking.000webhostapp.com`.

Перенесення файлів веб-сервісу на веб-хостинг ми також будемо здійснювати за допомогою протоколу FTP. Тоді у налаштуваннях веб-сайту на веб-хостингу 000webhost знаходимо параметри FTP-підключення. Це зображено на рисунку 2.6.

Для перенесення файлів веб-сервісу знову використовуємо FTP-клієнт FileZilla. У вікні цієї програми вводимо параметри підключення та встановлюємо з'єднання з FTP-сервером 000webhost. Далі у файловій системі FTP-сервера у каталог «`public_html`» пересилаємо файли веб-сервісу.

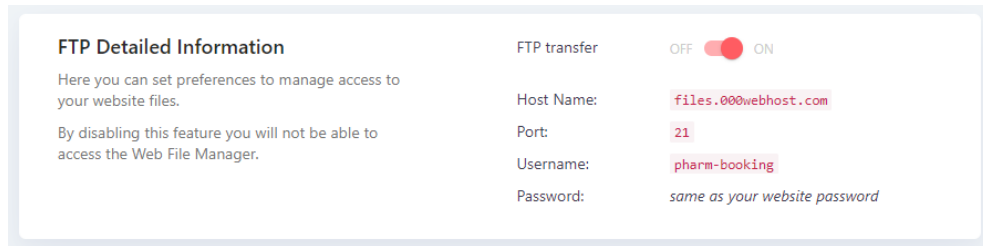


Рисунок 2.6 – Параметри підключення до FTP-сервера 000webhost

На цьому ми й закінчуємо перенесення файлів розробленого веб-сервісу на веб-хостинг 000webhost та закриваємо встановлене FTP-з'єднання.

### 2.4.3 Перенесення бази даних веб-сервісу на веб-хостинг

Як ми вже раніше вказували, веб-хостинг 000webhost підтримує СКБД MySQL. Для створення бази даних MySQL у 000webhost переходимо до менеджера баз даних і натискаємо кнопку створення бази даних. У вікні, яке з'явилось, вводимо назву БД, яку хочемо створити, ім'я користувача та пароль (рисунок 2.7). Далі натискаємо кнопку для створення бази даних.

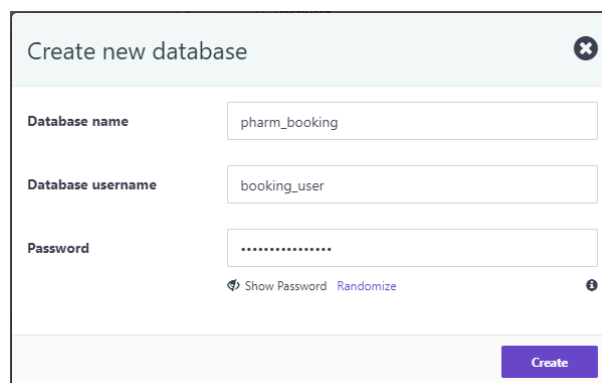


Рисунок 2.7 – Створення бази даних на веб-хостингу 000webhost

Після закінчення процесу створення БД у менеджері баз даних відображається основна інформація про БД – параметри для підключення до неї. Ці параметри нам потрібно прописати всередині розробленого веб-сервісу для того, щоб він міг підключатися саме до тієї БД, яка міститься на веб-хостингу.

Таким чином, отримані параметри підключення додаємо до коду файлу «connection.php» з каталогу «database» (лістинг В.4, додаток В), що й відповідає за підключення до БД. Після зміни файлу «connection.php» у локальному середовищі за допомогою FileZilla знову з'єднуємося з FTP-сервером веб-хостингу й надсилаємо оновлену версію файлу «connection.php».

Залишається лише перенести весь вміст локальної БД до БД на веб-хостингу. Оскільки в локальному середовищі та на веб-хостингу для адміністрування СКБД MySQL використовується веб-застосунок phpMyAdmin, то ми можемо скористатися його функціоналом імпорту-експорту баз даних для перенесення вмісту нашої БД.

Для цього спочатку ми відкриваємо phpMyAdmin в локальному середовищі, вибираємо БД pharm\_booking. Далі у вкладці «Експорт» зберігаємо файл «pharm\_booking.sql» в локальному середовищі. У цьому файлі містяться SQL-запити для створення всіх таблиць БД і заповнення їх певними значеннями.

Після цього запускаємо phpMyAdmin на веб-сервері, вибираємо БД, яку хочемо перенести дані, і відкриваємо вкладку «Імпорт». Це показано на рисунку 2.8.

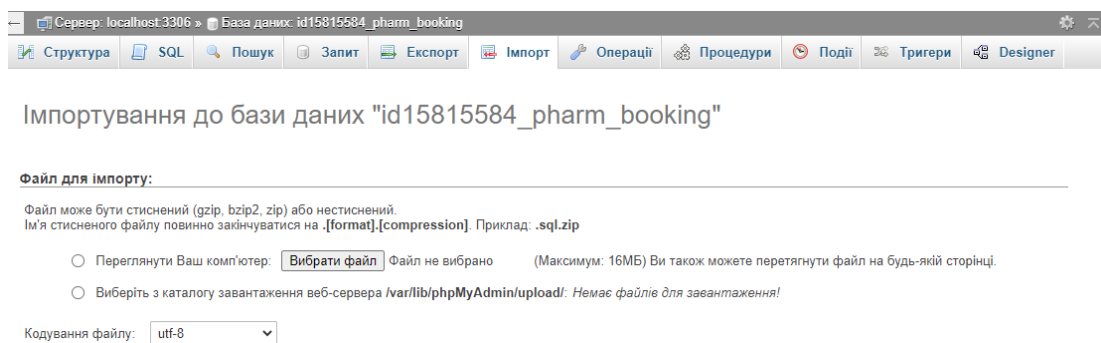


Рисунок 2.8 – Перенесення вмісту бази даних на веб-хостинг

У цій вкладці ми завантажуюємо файл з нашою БД «pharm\_booking.sql». Тоді на веб-хостингу phpMyAdmin запускає виконання запитів, що містяться в цьому файлі, і, таким чином, додає необхідний вміст до своєї БД.

## 2.5 Демонстрація роботи веб-сервісу для бронювання ліків

Після завершення розгортання веб-сервісу для бронювання ліків можемо перейти до демонстрації його роботи. Знімки екрана, які демонструють роботу розробленого веб-сервісу «PharmBooking», подані в додатку Д.

Спочатку опишемо демонстрацію роботи веб-сервісу для бронювання ліків зі звичайним користувачем. Робота з веб-сервісом починаємо з головної сторінки (рисунок Д.1).

На ній відображається список кнопок, призначених для перегляду ліків за певною категорією. Зверху, у навігаційній панелі, розміщене поле для пошуку ліків за назвою. Для перегляду ліків, що належать до категорії протизастудних, натискаємо кнопку «Протизастудні».

Після цього з'являється нове вікно, у якому відображаються ліки у вигляді блоків (рисунок Д.2). У кожному блоці міститься назва ліків, їх форма пакування, виробник, ціна та фото, а також кнопка для бронювання ліків. Натискаємо цю кнопку для двох ліків.

Після цього з'являються повідомлення, які вказують, що вибрані ліки було додано до списку бронювань (рисунок Д.3). Далі в навігаційній панелі натискаємо кнопку для переходу до списку бронювань.

На сторінці списку бронювань відображаються ліки, які були додані до нього раніше (рисунок Д.4). Для кожних ліків у списку бронювань вибираємо кількість одиниць для бронювання. Зауважимо, що при цьому внизу змінюється відображення загальної вартості бронювань.

Для надсилання списку записуємо у відповідних полях номер телефону й E-Mail (рисунок Д.5). Далі натискаємо кнопку «Надіслати». Після цього завантажуються сторінка, яка містить повідомлення про успішне надсилання списку бронювань (рисунок Д.6).

Розглянемо демонстрацію роботи веб-сервісу з адміністратором, тобто демонстрацію роботи в адмін-панелі. Для переходу до адмін-панелі натискаємо кнопку «Адмін-панель» на футері будь-якої сторінки веб-сервісу.

На головній сторінці адмін-панелі відображається форма авторизації (рисунок Д.7). Внизу, під формою авторизації, відображається кнопка для реєстрації користувача. Записуємо в полях форми авторизації логін і пароль адміністратора.

Оскільки логін та пароль були правильними, то завантажується робоча сторінка адмін-панелі (рисунок Д.8). На цій сторінці відображається лише випадний список, у якому можна вибрати таблицю для перегляду.

Вибираємо в цьому списку таблицю з записами про ліки. Відразу після цього вибору на сторінці з'являється таблиця, що містить записи про ліки (рисунок Д.9).

У кінці таблиці відображається кнопка для додавання нового запису в таблиць. У кожному рядку таблиці відображаються посилання для редагування або видалення відповідного запису.

Для виходу з адмін-панелі натискаємо кнопку «Вихід», що міститься в навігаційній панелі. Після натискання цієї кнопки відбувається перехід на сторінку авторизації до адмін-панелі, що вказує на закриття авторизованої сесії.

## **3 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ**

### **3.1 Психологічні чинники небезпеки**

Діяльність людини за своїми функціями поділяється на дві групи: фізичну та розумову діяльність. Фізична діяльність людини пов'язана з роботою її м'язів та зі здійсненням конкретних дій. Розумова ж діяльність визначається такими психічними процесами, у ході яких людина використовує мовні символи й образи для планування певних своїх дій. Складовими розумової діяльності є навчання, творчість, науки, управління, викладання – роботи, необхідним для яких є прийняття й обробка інформації. Виконання таких робіт вимагає напруження уваги, пам'яті, сенсорної системи, активної діяльності мислення та емоційного апарату [36, 37].

Виділяють перелік психологічних чинників, які збільшують індивідуальні ризики небезпеки, як під час розумової, так і під час фізичної діяльності людини. Трудова діяльність людини, яка має несприятливі прояви, підвищує рівень небезпеки для її здоров'я. Так, психологи виділяють спеціальний розділ психології – психологію безпеки, що вивчає психологічні особливості, які виникають під час трудової діяльності [38].

У психології безпеки важливими для вивчення є психологічні чинники, які впливають на збільшення рівня небезпеки. Загалом вони поділяються на дві групи: чинники, які стійко підвищують індивідуальну ймовірність наразитися на небезпеку, та чинники, що тимчасово підвищують індивідуальну ймовірність наразитися на небезпеку.

Так, до стійких чинників належить порушення зв'язку між сенсорними та руховими центрами вищих відділів нервової системи. Ці порушення впливають на діяльність людини так, що вона не може адекватно реагувати на подразники, які сприймаються органами чуття. Важливе значення у зменшенні їх впливу має доведений до автоматизму рівень відпрацювання навиків [39].

Також, до стійких чинників відносять вади органів чуття, як, наприклад часткова втрата слуху або зору. Найкращим методом зменшення небезпеки для людей, які мають такі вади, є набуття більшої кількості практичних знань і наявність відповідального відношення до роботи [40].

Ймовірність наразитися на небезпеку може підвищуватися й через незадоволеність роботою, відсутність мотивації до неї, які сприяють розсіяності уваги та неможливості психічного налаштування на роботу, стають причинами багатьох нещасних випадків. Таким чином, вагомим для усунення небезпеки є здійснення людиною вибору діяльності, що якомога найкраще відповідає її інтересам.

Слід зазначити ще один стійкий чинник підвищення індивідуальної ймовірності наразитися на небезпеку – пристрасть до алкоголю та наркотиків, які негативно діють на психічне здоров'я людини: викликають сповільнення темпу мислення, погіршення самоконтролю, зменшення швидкості реакції та загалом зумовлюють деградацію людини [38, 39].

Іншим стійким чинником, який впливає на підвищення рівня небезпеки, є неврівноваженість емоційних процесів: раптові зміни радості і злості, висока емоційна збудливість, сильні емоційні реакції на зовнішні подразники. Для уникнення неврівноваженості емоційних процесів потрібно постійно займатися самовихованням та розвивати самовладання [40].

До іншої групи психологічних чинників небезпеки відносяться чинники, які тимчасово підвищують індивідуальну ймовірність наразитися на небезпеку. Такі чинники можуть проявлятися тільки під час певних періодів трудової діяльності людини і впливати на її поведінку лише протягом короткого часу.

Тимчасовим чинником небезпеки є необережність. Особливістю цього чинника є те, що він підвищує ймовірність нещасного випадку не тільки для самого працівника, але й для всього робочого колективу. Головним способом боротьби з необережністю є вироблення обережності, яка характеризується



дисциплінованістю, правильним використанням автоматизму дій, раціональним керуванням увагою та підтримкою порядку на робочому місці [39].

Іншим чинником, що тимчасово впливає на підвищення рівня небезпеки, є недостатність досвіду. Так, завдяки відсутності вміння розподіляти увагу, недосвідченість працівника сприяє збільшенню ймовірності помилок та приводить до швидкого стомлення. Підвищення рівня досвідченості може досягатися стажем роботи й визначається навичками та особистими якостями працівника.

Також на підвищення ймовірності наразитися на небезпеку впливають й емоційні явища та потрясіння: сімейні чвари, сварки та інші конфліктні ситуації. Вони продукують хвилювання та нестриманість, що проявляються протягом деякого часу й негативно впливають на увагу, мислення. Крім негативних переживань, до емоційних явищ, які можуть збільшувати ймовірність небезпеки, відносять також і раптові відчуття радості, які зменшують уважність та призводять до нещасних випадків [40].

Серед тимчасових чинників небезпеки потрібно вказати і втому. Психологічними проявами втоми є сповільненість мислення, зниження вміння до запам'ятовування й концентрації уваги, послаблена реакція на подразники. Головним і найдієвішим способом позбутися втоми є тривалий відпочинок [39].

Психологічні чинники небезпеки сильно впливають на розумову діяльність, що є основною частиною роботи розробників програмного забезпечення й інших працівників сфери інформаційних технологій. Значущу роль в ІТ-сфері відіграють такі чинники, як втома, різні емоційні процеси і явища.

Для зменшення рівня небезпеки для здоров'я, керівникам і працівникам підприємств, що працюють в ІТ-сфері, потрібно враховувати прояви психологічних чинників небезпеки у процесі виконання роботи, у тому числі й під час розробки веб-сервісу для бронювання ліків, та застосовувати різноманітні способи їх усунення.

### 3.2 Вимоги ергономіки до організації робочого місця оператора ПК

Трудова діяльність людини визначається як засобами праці, працездатністю самої людини, гігієнічними факторами виробничого середовища, так і організацією виробничого місця.

Раціональне облаштування робочого місця, забезпечення зручного розміщення працівника (зручна поза, свобода рухів) та використання обладнання, що відповідає вимогам ергономіки підвищують ефективність роботи, зменшують ймовірність втоми й появи професійних хвороб [41].

Це стосується й робочого місця оператора ПК, яке складається з відповідного обладнання: екранного пристрою ПК, до якого можуть підключатися пристрої взаємодії з користувачем (клавіатура, миша), інші периферійні пристрої, друкувальні пристрої і пристрої зв'язку, з робочого крісла, робочого столу (поверхні) та виробничого середовища [42].

Організація робочого місця оператора ПК визначається Вимогами щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями, затвердженими наказом Міністерства соціальної політики України № 207 від 14.02.2018, і Державними санітарними правилами і нормами роботи з візуальними дисплейними терміналами електронно-обчислювальних машин, затвердженими постановою Головного державного санітарного лікаря України № 7 від 10.12.1998 (ДСанПіН 3.3.2-007-98) [43].

Згідно нормативними документами, організація робочого місця оператора ПК має відповідати як ергономічним вимогам щодо відповідності й розташування всіх елементів, так і особливостям трудової діяльності. Таким чином, задаються нормативні вимоги щодо площі одного робочого місця – 6 м<sup>2</sup> і його об'єму – 20 м<sup>3</sup> [42, 44].

Робочий стіл або поверхня, відповідно до вимог, повинні мати достатній розмір, щоб допускати гнучкість під час розміщення екрану ПК, клавіатури, відповідного обладнання. У загальному задаються наступні норми параметрів

робочого столу (поверхні), які забезпечують досяжність працівника для виконання робочих операцій: висота – 68-80 см, ширина – 80-140 см та глибина – 80-100 см [45].

Для робочого сидіння (крісла) оператора ПК використовуються наступні вимоги. Воно повинно дозволяти легко здійснювати рухи й займати зручне положення. Робоче сидіння повинно бути підйомно-поворотним, мати стаціонарні або змінні підлокітники для того, щоб усувати статичне напруження м'язів рук. Також за вимогами, при необхідності, разом з робочим сидінням рекомендується використовувати підніжку.

Відповідно до вимог освітлення, на робочому місці має бути створений контраст між екраном ПК й навколишнім середовищем. Робоче місце повинно розташовуватися біля вікон так, щоб природне світло могло попадати збоку від працівника, переважно з лівої сторони [42, 43].

Разом з тим, ДСанПіН 3.3.2-007-98 визначають і вимоги щодо організації кількох робочих місць операторів ПК, розташування їх між собою. Так, задаються вимоги до відстані між бічними поверхнями екранів ПК – не менше 1,2 м і вимоги відстані між тильними поверхнями екранів – не менше 2,5 м. При цьому, у разі потреби високої концентрації уваги під час виконання робіт вимагається встановлення перегородок висотою 1,5-2 м між суміжними робочими місцями операторів [45].

Вимоги ергономіки до робочого місця оператора ПК, які задаються нормативними документами, також стосуються робочих місць в ІТ-підприємствах, оскільки робота більшості їхніх працівників пов'язана насамперед з використанням ПК.

Виконання вимог ергономіки в ІТ-підприємствах дозволяє працівникам підвищувати рівень ефективності праці, зменшувати ризики отримання професійних захворювань. Саме тому на вимоги ергономіки слід зважати при здійсненні діяльності в ІТ-сфері, до якої входить і розробка веб-сервісу для бронювання ліків.

## ВИСНОВКИ

У результаті виконання кваліфікаційної роботи було спроектовано й розроблено веб-сервіс для бронювання, який можна використовувати при реалізації ліків невеликими аптечними мережами регіонального рівня.

У першому розділі кваліфікаційної роботи:

- проведено аналіз предметної області, розглянуто веб-сервіси для бронювання ліків, представлені на українському фармацевтичному ринку, що дало можливість здійснити постановку завдання та визначити особливості розроблюваного веб-сервісу;

- визначено вимоги до розроблюваного веб-сервісу та виділено зовнішні сутності й варіанти використання, за допомогою чого розділено функціонал веб-сервісу на дві частини: для звичайного користувача і для адміністратора;

- здійснено проектування архітектури розроблюваного веб-сервісу на основі трирівневої моделі «клієнт-сервер».

У другому розділі кваліфікаційної роботи:

- визначено засоби розробки веб-сервісу для бронювання ліків й обґрунтовано їх вибір, що зумовило особливості розробки веб-сервісу;

- розроблено модель даних, яка використовується для проектування бази даних веб-сервісу, здійснено розробку компонентів веб-сервісу та наведено опис їх взаємодії;

- виконано розгортання розробленого веб-сервісу для бронювання ліків на вибраному веб-хостингу, що дозволило провести опис демонстрації роботи веб-сервісу на прикладі звичайного користувача й адміністратора.

У розділі «Безпека життєдіяльності, основи охорони праці»:

- висвітлено основні психологічні чинники небезпеки та визначено способи їх усунення при розробці веб-сервісу для бронювання ліків;

- розглянуто вимоги ергономіки до організації робочого місця оператора ПК та їх роль у виконанні розробки веб-сервісу для бронювання ліків.

## ПЕРЕЛІК ДЖЕРЕЛ

1. Аналіз використання цифрових технологій у медицині та фармації / А.С. Немченко, Ю.Є. Куриленко, В.М. Назаркіна, Ю.О. Шеншина. // Від експериментальної та клінічної патофізіології до досягнень сучасної медицини і фармації: матеріали ІІ науково-практичної конференції студентів та молодих вчених з міжнародною участю. – Харків: НФаУ, 2020. – С. 159–160.
2. Сахнацька Н.М. Проблеми дистанційної реалізації ліків у контексті надання якісної та доступної фармацевтичної допомоги населенню / Н.М. Сахнацька. // Сучасна фармація: історія, реалії та перспективи розвитку: Матеріали науково-практичної конференції з міжнародною участю, присвяченої 20-й річниці заснування Дня фармацевтичного працівника України. – Київ: НМУ імені О.О. Богомольця, 2019. – С. 343–344.
3. Дмитрик К. Аптечний ринок: консолідація та розвиток бізнесу в епоху digital [Електронний ресурс] / Катерина Дмитрик // Аптека.ua. – 2019. – Режим доступу до ресурсу: <https://www.apteka.ua/article/517257>.
4. Будина Л. “Мы меняем фарм-рынок”. История основателей прайс-агрегатора Tabletki.ua [Електронний ресурс] / Людмила Будина // Retailers. – 2018. – Режим доступу до ресурсу: <https://retailers.ua/news/tehnologii/8425-my-menyaem-farm-ryinok-istoriya-osnovateley-prays-agregatora-tabletkiua>.
5. Tabletki.ua - пошук і бронювання ліків в аптеках, інструкції по застосуванню, тисячі аптек по всій Україні [Електронний ресурс] – Режим доступу до ресурсу: <https://tabletki.ua/uk>.
6. Tabletki.ua Traffic, Ranking & Marketing Analytics | SimilarWeb [Електронний ресурс] – Режим доступу до ресурсу: <https://www.similarweb.com/website/tabletki.ua>.
7. Пошук ліків в аптеках Києва та всієї України | Аптека поруч [Електронний ресурс] – Режим доступу до ресурсу: <https://geoapteka.ua/ua>.

8. GEO-Аптека! Сервис поиска лекарств в аптеках Украины!!! - Программный комплекс "Аптека" [Электронный ресурс] – Режим доступа до ресурсу: <https://pharmbase.com.ua/ru/geo-apteka-servis-poiska-lekarstv-v-apteках-ukrainy>.

9. Аптека Доброго Дня - Интернет аптека ліків і медикаментів [Электронный ресурс] – Режим доступа до ресурсу: <https://www.add.ua/ua>.

10. «Аптека Доброго Дня» выпустила мобильное приложение [Электронный ресурс] // Аптека.ua. – 2018. – Режим доступа до ресурсу: <https://www.apteka.ua/article/477096>.

11. 1 Соціальна Аптека - офіційний сайт, ціни на ліки, адреси аптек [Электронный ресурс] – Режим доступа до ресурсу: <https://1sa.com.ua>.

12. Дорош В. В этом году в сети «1 социальная аптека» открыли 84 аптеки [Электронный ресурс] / Валерия Дорош // Retailers. – 2017. – Режим доступа до ресурсу: <https://retailers.ua/news/menedjment/6770-v-etom-godu-v-seti-persha-sotsilna-apteka-otkryili-84-apteki>.

13. Software Engineering | Classification of Software Requirements [Электронный ресурс]. – 2018. – Режим доступа до ресурсу: <https://www.geeksforgeeks.org/software-engineering-classification-of-software-requirements>.

14. Software Requirements - Tutorialspoint [Электронный ресурс] – Режим доступа до ресурсу: [https://www.tutorialspoint.com/software\\_engineering/software\\_requirements.htm](https://www.tutorialspoint.com/software_engineering/software_requirements.htm).

15. What is Use Case Diagram? [Электронный ресурс] – Режим доступа до ресурсу: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-use-case-diagram>.

16. Дідловська М.В. Діаграма варіантів використання [Электронный ресурс] / М.В. Дідловська – Режим доступа до ресурсу: [http://mmsa.kpi.ua/sites/default/files/disciplines/Розробка%20і%20тестування%20Програм/didkovska\\_m\\_v\\_testing\\_lecture\\_3.pdf](http://mmsa.kpi.ua/sites/default/files/disciplines/Розробка%20і%20тестування%20Програм/didkovska_m_v_testing_lecture_3.pdf).

17. Змерзлий І. Клієнт-серверна архітектура та ролі серверів [Електронний ресурс] / Іван Змерзлий // Medium – Режим доступу до ресурсу: <https://medium.com/@IvanZmerzlyi/клієнт-серверна-архітектура-та-ролі-серверів-9893d8048229>.

18. What is Client-Server? Definition and FAQs [Електронний ресурс] – Режим доступу до ресурсу: <https://www.omnisci.com/technical-glossary/client-server>.

19. Client-Server Architecture | Components, Types, Examples [Електронний ресурс] – Режим доступу до ресурсу: <https://teachcomputerscience.com/client-server-architecture>.

20. What is Three-Tier Architecture | IBM [Електронний ресурс] – Режим доступу до ресурсу: <https://www.ibm.com/cloud/learn/three-tier-architecture>.

21. PHP: What is PHP? - Manual [Електронний ресурс] – Режим доступу до ресурсу: <https://www.php.net/manual/en/intro-what-is.php>.

22. Kevin Tatroe. Programming PHP: Creating Dynamic Web Pages / Kevin Tatroe, Peter MacIntyre. – O'Reilly Media, Inc. – 2020. – 544 с. ISBN 1492054089, 9781492054085.

23. Скриптова мова PHP - Web-технології та Web-дизайн [Електронний ресурс] – Режим доступу до ресурсу: <https://sites.google.com/site/da21svietlova/skriptova-mova-php>.

24. Bootstrap · The most popular HTML, CSS, and JS library in the world. [Електронний ресурс] – Режим доступу до ресурсу: <https://getbootstrap.com>.

25. Introduction - JavaScript | MDN [Електронний ресурс] – Режим доступу до ресурсу: <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Introduction?retiredLocale=uk>.

26. jQuery - Overview - Tutorialspoint [Електронний ресурс] – Режим доступу до ресурсу: <https://www.tutorialspoint.com/jquery/jquery-overview.htm>.

27. MySQL - Introduction - Tutorialspoint [Електронний ресурс] – Режим доступу до ресурсу: <https://www.tutorialspoint.com/mysql/mysql-introduction.htm>.

28. MySQL :: MySQL Reference Manual :: 1.2 Overview of the MySQL Database Management System [Електронний ресурс] – Режим доступу до ресурсу: <https://dev.mysql.com/doc/refman/8.0/en/what-is.html>.

29. Entity Relationship Diagram - ER Diagram in DBMS [Електронний ресурс] – Режим доступу до ресурсу: <https://beginnersbook.com/2015/04/e-r-model-in-dbms>.

30. Бази даних. Теорія і практика | BestProg [Електронний ресурс] – Режим доступу до ресурсу: [https://www.bestprog.net/uk/sitemap\\_ua/бази-даних](https://www.bestprog.net/uk/sitemap_ua/бази-даних).

31. What is Entity Relationship Diagram (ERD)? [Електронний ресурс] – Режим доступу до ресурсу: <https://www.visual-paradigm.com/guide/data-modeling/what-is-entity-relationship-diagram>.

32. Activity Diagram Tutorial | How to Draw an Activity Diagram [Електронний ресурс] – Режим доступу до ресурсу: <https://creately.com/blog/diagrams/activity-diagram-tutorial>.

33. Activity Diagram - Activity Diagram Symbols, Examples, and More [Електронний ресурс] – Режим доступу до ресурсу: <https://www.smartdraw.com/activity-diagram>.

34. Image CDN: Image Optimization, Processing & Hosting • Sirv [Електронний ресурс] – Режим доступу до ресурсу: <https://sirv.com>.

35. Free Web Hosting - Host a Website for Free with Cpanel, PHP [Електронний ресурс] – Режим доступу до ресурсу: <https://www.000webhost.com>.

36. Желібо Є.П. Безпека життєдіяльності: навч. посіб. / Є.П. Желібо, Н.М. Заверуха, В.В. Зацарний. – Київ: Каравела, 2008. – 344 с.

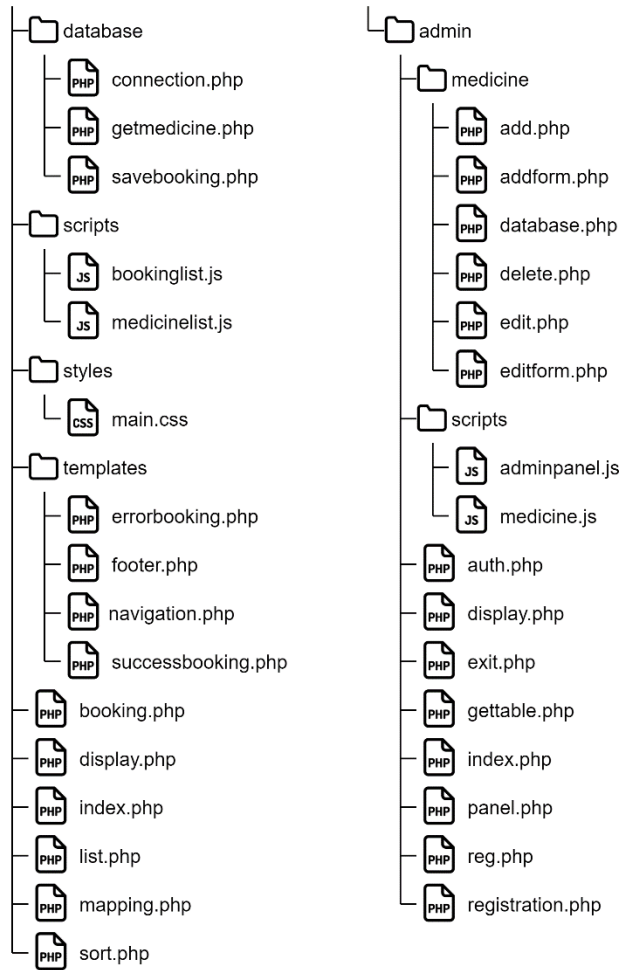
37. Толлок А.О. Безпека життєдіяльності: навч. посіб. / А.О. Толлок, О.А. Крюковська. – Дніпродзержинськ: ДДТУ, 2011. – 215 с.



38. Бердій Я.І. Безпека життєдіяльності: Навчальний посібник / Я.І. Бердій. – Київ: Кондор, 2009. – 286 с.
39. Березюк О.В. Безпека життєдіяльності [Електронний ресурс] / О.В. Березюк, М.С. Лемешев – Режим доступу до ресурсу: [https://web.posibnyky.vntu.edu.ua/fmbt/berezyuk\\_bezpeka\\_zhittyediyalnosti](https://web.posibnyky.vntu.edu.ua/fmbt/berezyuk_bezpeka_zhittyediyalnosti).
40. Коцан І. Безпека життєдіяльності / Ігор Коцан. – Харків: Фоліо, 2014. – 462 с.
41. Геврик Є.О. Охорона праці: Навчальний посібник для студентів вищих навчальних закладів / Є.О. Геврик. – Київ: Ельга, Ніка-Центр, 2003. – 280 с.
42. Про затвердження Вимог щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями. Наказ Міністерства соціальної політики України № 207 від 14.02.2018 [Електронний ресурс] // Верховна Рада України. Законодавство України – Режим доступу до ресурсу: <https://zakon.rada.gov.ua/laws/show/z0508-18>.
43. Вимоги охорони праці при виконанні робіт на персональному комп'ютері [Електронний ресурс] – Режим доступу до ресурсу: <https://selidovo-rada.gov.ua/novini/vimogi-okhoroni-pratsi-pri-vikonanni-robit-na-personalnomu-komp-yuteri>.
44. Гандзюк М.П. Основи охорони праці / М.П. Гандзюк, Є.П. Желібо, М.О. Халімовський. – Київ: Каравела, 2004. – 408 с.
45. Грибан В.Г. Охорона праці. Навчальний посібник / В.Г. Грибан, О.В. Негодченко. – Київ: Центр учбової літератури, 2011. – 280 с.

# ДОДАТКИ

Схема файлової структури веб-сервісу для бронювання ліків



Діаграми діяльності деяких процесів  
у веб-сервісі для бронювання ліків

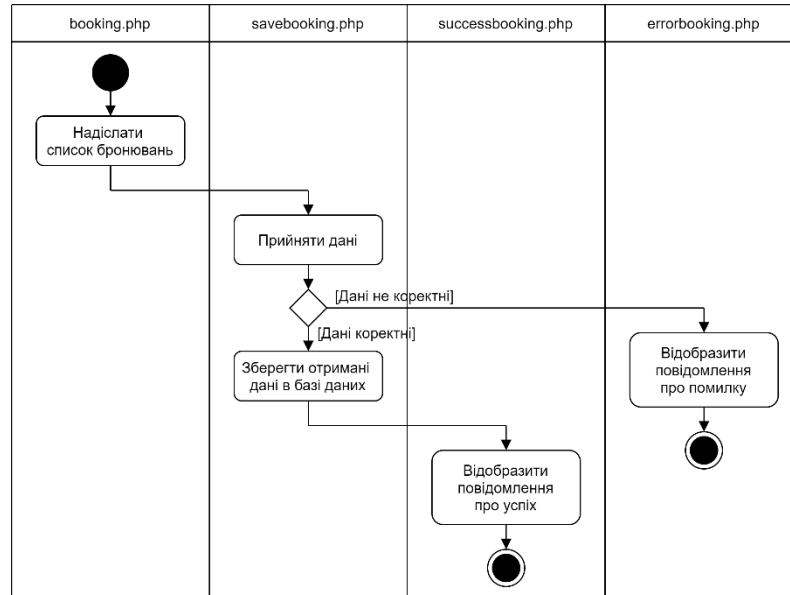


Рисунок Б.1 – Діаграма діяльності процесу надсилання списку бронювань

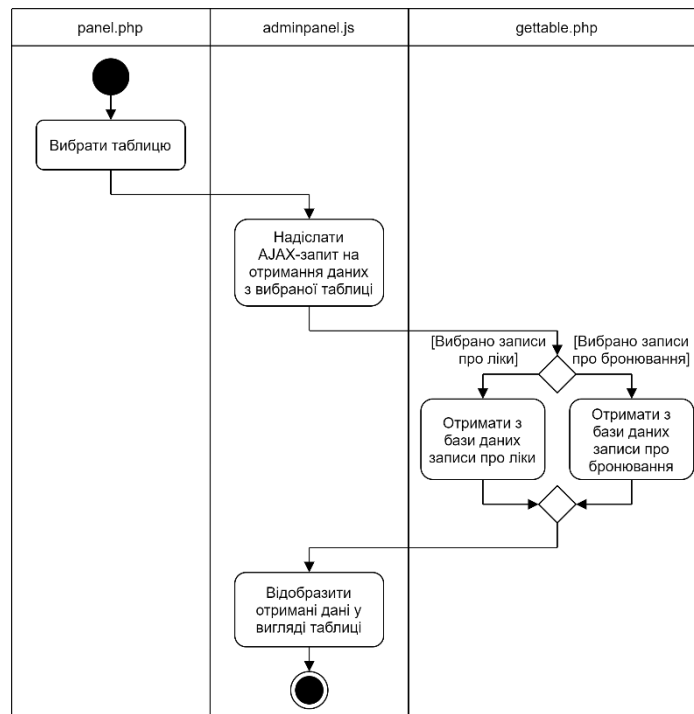


Рисунок Б.2 – Діаграма діяльності процесу вибору таблиці для перегляду

## Лістинги файлів компонентів, що відповідають за взаємодію з користувачем

### Лістинг В.1 – Код файлу «index.php»

```
<?php
// Визначення файлу сторінки
define('PAGE_FILE', "index.php");
require_once "mapping.php";
?>
<!DOCTYPE html>
<html lang="uk">
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1">
  <title><?=$title_map[PAGE_FILE] . " - PharmBooking"?></title>
  <link rel="stylesheet" type="text/css"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootst
rap.min.css">
  <link rel="stylesheet" type="text/css" href="styles/main.css">
  <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.m
in.js"></script>
  <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstra
p.min.js"></script>
</head>
<body>
  <!-- NAVIGATION -->
  <?php require "templates/navigation.php"; // Навігаційна
панель ?>
  <!-- /NAVIGATION -->

  <!-- MAIN SECTION -->
  <main class="bg-light pt-sm-3 pb-sm-3" id="page">
    <div class="container">
      <form action="list.php" method="get" class="mx-auto w-75
pt-3 pb-5">
        <h3 class="pb-3">Категорії ліків</h3>
        <?php foreach (array_keys($category_map) as $key): ?>
          <button type="submit" class="col mb-2 btn btn-info
btn-lg" name="category" value="<?=$key?>">
            <?=$category_map[$key]?>
          </button>
        <?php endforeach; ?>
      </form>
    </div>
```

```

</main>
<!-- /MAIN SECTION -->

<!-- FOOTER -->
<?php require "templates/footer.php"; // Футер ?>
<!-- /FOOTER -->
</body>
</html>

```

## Лістинг В.2 – Код файлу «list.php»

```

<?php
// Визначення файлу сторінки
define('PAGE_FILE', "list.php");
require_once "mapping.php";
?>
<!DOCTYPE html>
<html lang="uk">
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1">
  <title><?=$title_map[PAGE_FILE] . " - PharmBooking"?></title>
  <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootst
rap.min.css">
  <link rel="stylesheet" href="styles/main.css">
  <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.m
in.js"></script>
  <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstra
p.min.js"></script>
  <script src="scripts/medicinelist.js"></script>
</head>
<body>
  <!-- NAVIGATION -->
  <?php
require "templates/navigation.php"; // Навігаційна панель
require_once "display.php";
?>
  <!-- /NAVIGATION -->

  <!-- MAIN SECTION -->
  <main class="bg-light pt-sm-3 pb-sm-3" id="page">
    <div class="container">
      <div class="clearfix mb-sm-3">
        <?php if (isset($_GET['category'])): // Вибрано певну
категорію ?>
          <div class="float-sm-left">
            <h3><?=$category_map[$_GET['category']]?></h3>

```

```

        </div>
        <?php endif; ?>
        <div class="form-inline float-sm-right">
            <label for="#sort-by" class="mr-sm-2">Спосіб
сортування</label>
            <select name="sort-by" id="sort-by" class="form-
control">
                <option value="price-asc">Спочатку
найдешевші</option>
                <option value="price-desc">Спочатку
найдорожчі</option>
                <option value="name-asc">За алфавітом</option>
            </select>
            <input type="hidden" id="query"
value="<?=$_GET['query'] . " ">">
            <input type="hidden" id="category"
value="<?=$_GET['category']?>">
        </div>
    </div>
    <?php
require_once "database/getmedicine.php";

// Якщо здійснено пошук за назвою
if (isset($_GET['query'])) {
    $name = filter_var(trim($_GET['query']),
FILTER_SANITIZE_STRING);

    // Пошук ліків за назвою
    $medicines = search($name);
}
// Якщо вибрано певну категорію
elseif (isset($_GET['category'])) {
    $category = filter_var(trim($_GET['category']),
FILTER_SANITIZE_STRING);

    // Отримання ліків за категорією
    $medicines = get_by_category($category);
}

// Якщо отримано якісь ліки за запитом
if (!empty($medicines)) {
    display_list($medicines);
}
// Якщо ліків за запитом не було отримано
else {
    echo '<div class="centered-text pt-sm-5">';
    echo '<h4>Ліків не знайдено</h4></div>';
}
?>
</div>
</main>
<!-- /MAIN SECTION -->

```

```

<!-- FOOTER -->
<?php require "templates/footer.php"; // Футер ?>
<!-- /FOOTER -->
</body>
</html>

```

### Лістинг В.3 – Код файлу «booking.php»

```

<?php
// Визначення файлу сторінки
define('PAGE_FILE', "booking.php");
require_once "mapping.php";
?>
<!DOCTYPE html>
<html lang="uk">
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1">
  <title><?=$title_map[PAGE_FILE] . " - PharmBooking"?></title>
  <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootst
rap.min.css">
  <link rel="stylesheet" href="styles/main.css">
  <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.m
in.js"></script>
  <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstra
p.min.js"></script>
  <script src="scripts/bookinglist.js"></script>
</head>
<body>
  <!-- NAVIGATION -->
  <?php
require "templates/navigation.php"; // Навігаційна панель
require_once "display.php";
?>
  <!-- /NAVIGATION -->

  <!-- MAIN SECTION -->
  <main class="bg-light pt-sm-3 pb-sm-3" id="page">
    <div class="container">
      <div class="mb-sm-3">
        <h3>Список бронювань</h3>
      </div>
      <form action="database/savebooking.php" method="post">
        <?php
          require_once "database/getmedicine.php";

```



```

// Якщо встановлено cookie "medicine"
if (isset($_COOKIE['medicine'])) {
    // ID ліків на бронювання
    $medicine_ids = array_keys($_COOKIE['medicine']);
    // Отримання ліків на бронювання
    $booking_medicines = get_booking_list($medicine_ids);
}
// Якщо є вибрані ліки для бронювання
if (!empty($booking_medicines)):
    // Відображення списку бронювань
    display_booking_list($booking_medicines);
    // Відображення форми для надсилання списку бронювань
?>
<div class="form-row pb-sm-3 pt-sm-2">
    <div class="col">
        <input type="text" id="phone" class="form-control"
placeholder="Введіть номер телефону (без +38)" name="phone"
required>
    </div>
    <div class="col">
        <input type="email" id="email" class="form-
control" placeholder="Введіть E-Mail" name="email">
    </div>
</div>
<div class="clearfix pl-sm-2">
    <div class="float-sm-left">
        <h4 id="total-price">Загалом: <small></small></h4>
    </div>
    <div class="form-inline float-sm-right">
        <button class="btn btn-info" id="send-bookings"
type="submit" disabled>Надіслати</button>
    </div>
</div>
<?php else: // Якщо ліків на бронювання не було
отримано ?>
    <div class="centered-text pt-sm-5"><h4>Список
бронювань порожній</h4></div>
    <?php endif; ?>
</form>
</div>
</main>
<!-- /MAIN SECTION -->

<!-- FOOTER -->
<?php require "templates/footer.php"; // Футер ?>
<!-- /FOOTER -->
</body>
</html>

```

#### Лістинг В.4 – Код файлу «connection.php»

```
<?php
// Параметри підключення до БД
$db_params = array(
    'hostname' => "localhost",
    'username' => "id15815584_booking_user",
    'password' => "XXXXXXXXXXXXXXXXXX",
    'dbname' => "id15815584_pharm_booking"
);

// Встановлює з'єднання з БД
function connect()
{
    global $db_params;

    $mysqli_conn = new mysqli($db_params['hostname'],
        $db_params['username'],
            $db_params['password'], $db_params['dbname']);

    if ($mysqli_conn->connect_error) {
        echo 'Помилка підключення: ' . $mysqli_conn->connect_error;
        exit();
    }
    return $mysqli_conn;
}

// Повертає асоціативний масив від об'єкта mysqli_result
function result_array($query_result)
{
    $entries = array();
    for ($i = 0; $entry = $query_result->fetch_assoc(); $i++) {
        $entries[$i] = $entry;
    }
    return $entries;
}
```

#### Лістинг В.5 – Код файлу «getmedicine.php»

```
<?php
require_once "connection.php";
require_once "../mapping.php";

// Здійснює отримання ліків з БД через пошук за назвою
function search($name, $sortby = "price-asc")
{
    global $sortby_map;
    $conn = connect(); // Підключення до БД
    // SQL-запит до БД
    $query = "SELECT * FROM view_medicine
        WHERE view_medicine.name LIKE CONCAT(?, '%')
        ORDER BY " . $sortby_map[$sortby];
```

```

if ($stmt = $conn->prepare($query)) { // Підготовка виразу
    // Встановлення змінних до параметрів
    $stmt->bind_param('s', $name);
    $stmt->execute(); // Виконання

    $result = $stmt->get_result(); // Результати виконання

    // Отримання асоціативного масиву ліків
    $medicines = result_array($result);

    // Закриття підготовленого виразу
    $stmt->close();
}
else {
    echo 'Помилка виконання запиту: ' . $conn->connect_error;
    exit();
}

$conn->close(); // Закриття підключення до БД
return $medicines; // Повернення масиву ліків
}

// Здійснює отримання ліків з БД за певною категорією
function get_by_category($category, $sortby = "price-asc")
{
    global $category_map, $sortby_map;
    $conn = connect(); // Підключення до БД

    // Назва категорії
    $category_name = $category_map[$category];

    // SQL-запит до БД
    $query = "SELECT * FROM view_medicine
        WHERE view_medicine.category = '$category_name'
        ORDER BY " . $sortby_map[$sortby];

    // Отримання результатів виконання запиту
    if ($result = $conn->query($query)) {
        // Отримання асоціативного масиву ліків
        $medicines = result_array($result);
        $result->free(); // Звільнення пам'яті
    }
    else {
        echo 'Помилка виконання запиту: ' . $conn->connect_error;
        exit();
    }

    $conn->close(); // Закриття підключення до БД
    return $medicines; // Повернення масиву ліків
}

```

```

// Здійснює отримання з БД ліків на бронювання
function get_booking_list($medicine_ids)
{
    $conn = connect(); // Підключення до БД

    $ids_list = ""; // Список з ID ліків на бронювання
    foreach ($medicine_ids as $medicine_id) {
        $ids_list .= "$medicine_id, ";
    }
    $ids_list = substr($ids_list, 0, -2);

    // SQL-запит до БД
    $query = "SELECT id, `name`, medicine_form, producter_name,
price
            FROM view_medicine
            WHERE view_medicine.id IN ($ids_list)";

    // Отримання результатів виконання запиту
    if ($result = $conn->query($query)) {
        // Отримання асоціативного масиву ліків
        $booking_medicines = result_array($result);
        $result->free(); // Звільнення пам'яті
    }
    else {
        echo 'Помилка виконання запиту: ' . $conn->connect_error;
        exit();
    }

    $conn->close(); // Закриття підключення до БД
    return $booking_medicines; // Повернення масиву ліків на
бронювання
}

```

### Лістинг В.6 – Код файлу «savebooking.php»

```

<?php
require_once "connection.php";

if (isset($_POST['booking'], $_POST['count'], $_POST['phone'],
$_POST['email'])) {
    $booking_ids = array_keys($_POST['booking']); // ID ліків на
бронювання
    $booking_counts = $_POST['count']; // Кількості ліків на
бронювання
    // Номер телефону
    $phone_number = filter_var(trim($_POST['phone']),
FILTER_SANITIZE_STRING);
    // E-Mail
    $email = filter_var(trim($_POST['email']),
FILTER_SANITIZE_STRING);
}

```

```

// Якщо E-Mail введено
if ($email != "") {
    // Збереження списку бронювань в БД з номером телефону й E-
Mail
    $booking_list_id = save_booking_list($phone_number, $email);
}
else {
    // Збереження списку бронювань в БД тільки з номером
телефону
    $booking_list_id = save_booking_list($phone_number);
}

// Якщо масиви ID ліків на бронювання і їх
// кількостей не порожні та їхні розміри однакові
if (!empty($booking_ids) && !empty($booking_counts) &&
(count($booking_ids) == count($booking_counts))) {
    // Збереження бронювань
    save_bookings($booking_ids, $booking_counts,
$booking_list_id);
    // Видалення відповідних cookie
    erase_medicine_cookies($booking_ids);
    // Перенаправлення на сторінку з повідомленням
// про успішність надсилання списку бронювань
    header("Location: /templates/successbooking.php");
}
else {
    // Перенаправлення на сторінку з повідомленням про помилку
    header("Location: /templates/errorbooking.php");
}
}
else {
    // Перенаправлення на сторінку з повідомленням про помилку
    header("Location: /templates/errorbooking.php");
}

// Видаляє cookie для ліків за вказаним ID
function erase_medicine_cookies($medicine_ids)
{
    foreach ($medicine_ids as $medicine_id) {
        // Видалення cookie - встановлення його з часом існування на
1 годину назад
        setcookie("medicine[$medicine_id]", 0, time() - 3600, "/");
    }
}

// Здійснює створення нового списку
// бронювань у БД та повернення його ID
function save_booking_list($phone, $email = NULL)
{
    $conn = connect(); // Підключення до БД

```

```

    if ($email != NULL) {
        // Якщо задано E-Mail, то SQL-запит призначений для
створення
        // нового списку із вказанням номеру телефону й E-Mail
        $query = "CALL set_booking_list(?, ?)";
    }
    else {
        // Якщо E-Mail не задано, то SQL-запит призначений для
створення
        // нового списку тільки із вказанням номеру телефону
        $query = "CALL set_booking_list(?, NULL)";
    }

    if ($stmt = $conn->prepare($query)) { // Підготовка виразу
        if ($email != NULL) {
            // Якщо що задано E-Mail, то до параметрів підготовленого
// виразу встановлюється номер телефону й E-Mail
            $stmt->bind_param('ss', $phone, $email);
        }
        else {
            // Якщо E-Mail не задано, то до параметрів підготовленого
// виразу встановлюється тільки номер телефону
            $stmt->bind_param('s', $phone);
        }
        $stmt->execute(); // Виконання підготовленого виразу

        // Результати виконання
        $result = $stmt->get_result();

        // Отримання ID створеного списку
        $id_array = result_array($result)[0];
        $stmt->close(); // Закриття підготовленого виразу
    }
    else {
        echo 'Помилка виконання запиту: ' . $conn->connect_error;
        exit();
    }

    $conn->close(); // Закриття підключення до БД
    return $id_array['id']; // Повернення ID створеного списку
}

// Здійснює збереження бронювань у БД за вказаними масивами
// ID ліків, їхніми кількостями та ID списку бронювань
function save_bookings($ids, $counts, $list_id)
{
    $conn = connect(); // Підключення до БД

    // Початок SQL-запиту до БД
    $query = "INSERT INTO booking
        (booking_list_id, medicine_id, `count`)
        VALUES ";

```

```

// Додавання до SQL-запиту місця для
// додавання заданої кількості бронювань
for ($i = 0; $i < count($ids); $i++) {
    $query .= "(?, ?, ?), ";
}
$query = substr($query, 0, -2);

if ($stmt = $conn->prepare($query)) { // Підготовка виразу
    $types = array(""); // Рядок для типів значень змінних
    $vars = array(); // Масив для значень змінних

    foreach ($ids as $key => $id) {
        $types[0] .= "iii";
        array_push($vars, $list_id, $ids[$key], $counts[$id]);
    }
    $vars = array_merge($types, $vars);

    // Параметрів методу bind_param()
    $params = array();
    foreach ($vars as $key => $var) {
        $params[$key] = &$vars[$key];
    }

    // Виклик методу bind_param() з відповідними параметрами
    // для встановлення змінних до підготовленого виразу
    call_user_func_array(array($stmt, 'bind_param'), $params);
    $stmt->execute(); // Виконання підготовленого виразу
    $stmt->close(); // Закриття підготовленого виразу
}
else {
    echo 'Помилка виконання запиту: ' . $conn->connect_error;
    exit();
}
$conn->close(); // Закриття підключення до БД
}

```

## Лістинги файлів компонентів, що відповідають за взаємодію з адміністратором

### Лістинг Г.1 – Код файлу «index.php» у каталозі «admin»

```
<?php
// Визначення файлу сторінки
define('PAGE_FILE', "admin/");
require_once "../mapping.php";

if (isset($_COOKIE['administrator'])) {
    // Якщо встановлено cookie "administrator", тобто
    адміністратор авторизований,
    // то виконується перенаправлення на сторінку роботи в адмін-
    панелі
    header("Location: panel.php");
}
?>
<!DOCTYPE html>
<html lang="uk">
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1">
    <title><?=$title_map[PAGE_FILE] . " - PharmBooking"?></title>
    <link rel="stylesheet" type="text/css"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootst
rap.min.css">
    <link rel="stylesheet" type="text/css"
href="/styles/main.css">
    <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.m
in.js"></script>
    <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstra
p.min.js"></script>
</head>
<body>
    <!-- NAVIGATION -->
    <?php require "../templates/navigation.php"; // Навігаційна
панель ?>
    <!-- /NAVIGATION -->

    <!-- MAIN SECTION -->
    <main class="bg-light pt-sm-3 pb-sm-3" id="page">
        <div class="container">
            <form action="auth.php" method="post">
                <div class="mt-sm-5 mb-sm-4 centered-text">
```



```

        <h3>PharmBooking</h3>
        <h4>Адмін-панель</h4>
    </div>
    <div class="mx-auto" style="width: 30%;">
        <div class="form-group mb-sm-3">
            <input type="text" name="input-login" class="form-
control" placeholder="Введіть логін" required>
        </div>
        <div class="form-group mb-sm-3">
            <input type="password" name="input-password"
class="form-control" placeholder="Введіть пароль" required>
        </div>
        <div class="form-group mb-sm-3">
            <button type="submit" class="btn btn-info btn-
block">Авторизуватися</button>
        </div>
        <div class="form-group">
            <a href="registration.php" class="btn btn-link btn-
block">Реєстрація</a>
        </div>
    </div>
</form>
</div>
</main>
<!-- /MAIN SECTION -->

<!-- FOOTER -->
<?php require "../templates/footer.php"; // Футер ?>
<!-- /FOOTER -->
</body>
</html>

```

## Лістинг Г.2 – Код файлу «panel.php»

```

<?php
// Визначення файлу сторінки
define('PAGE_FILE', "admin/panel.php");
require_once "../mapping.php";

if (!isset($_COOKIE['administrator'])) {
    // Якщо не встановлено cookie "administrator", тобто
адміністратор не авторизований,
    // то виконується перенаправлення на сторінку авторизації в
адміністративній панелі
    header("Location: .");
}
?>
<!DOCTYPE html>
<html lang="uk">
<head>
    <meta charset="utf-8">

```

```

    <meta name="viewport" content="width=device-width, initial-
scale=1">
    <title><?=$title_map[PAGE_FILE] . " - PharmBooking"?></title>
    <link rel="stylesheet" type="text/css"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootst
rap.min.css">
    <link rel="stylesheet" type="text/css"
href="/styles/main.css">
    <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.m
in.js"></script>
    <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstra
p.min.js"></script>
    <script src="scripts/adminpanel.js"></script>
</head>
<body>
    <!-- NAVIGATION -->
    <?php require "../templates/navigation.php"; // Навігаційна
панель ?>
    <!-- /NAVIGATION -->

    <!-- MAIN SECTION -->
    <main class="bg-light pt-sm-3 pb-sm-3" id="page">
        <div class="container">
            <div class="form-inline mb-sm-3" id="choose-table">
                <select id="table-name" class="form-control">
                    <option value="">Виберіть таблицю для
перегляду</option>
                    <option value="medicine">Ліки</option>
                    <option value="booking">Бронювання</option>
                </select>
            </div>
            <table class="table table-striped table-hover"></table>
        </div>
    </main>
    <!-- /MAIN SECTION -->

    <!-- FOOTER -->
    <?php require "../templates/footer.php"; // Футер ?>
    <!-- /FOOTER -->
</body>
</html>

```

### Лістинг Г.3 – Код файлу «gettable.php»

```

<?php
require_once "../database/connection.php";
require_once "display.php";

```

```

if (isset($_GET['tablename'])) {
    // Назва таблиці
    $tablename = filter_var(trim($_GET['tablename']),
FILTER_SANITIZE_STRING);

    if ($tablename == "medicine") {
        // Якщо задано таблицю з ліками, то виводиться таблиця з
ліками
        get_medicines();
    }
    elseif ($tablename == "booking") {
        // Якщо задано таблицю з бронюваннями, то виводиться таблиця
з бронюваннями
        get_bookings();
    }
}

// Виводить таблицю з ліками, отриманими з БД
function get_medicines()
{
    $conn = connect(); // Підключення до БД

    // SQL-запит до БД на отримання ліків
    $query = "SELECT * FROM view_medicine ORDER BY `name` ASC";

    // Отримання результатів виконання запиту
    if ($result = $conn->query($query)) {
        $medicines = result_array($result); // Отримання
асоціативного масиву
        $result->free(); // Звільнення пам'яті
    }
    else {
        echo 'Помилка виконання запиту: ' . $conn->connect_error;
        exit();
    }

    // Закриття підключення до БД
    $conn->close();

    // Відображення таблиці з бронюваннями
    display_medicine_table($medicines);
}

// Виводить таблицю з бронюваннями, отриманими з БД
function get_bookings()
{
    $conn = connect(); // Підключення до БД

    // SQL-запит до БД на отримання бронювань
    $query = "SELECT * FROM view_booking ORDER BY booking_time
DESC";
}

```

```

// Отримання результатів виконання запиту
if ($result = $conn->query($query)) {
    $bookings = result_array($result); // Отримання
асоціативного масиву
    $result->free(); // Звільнення пам'яті
}
else {
    echo 'Помилка виконання запиту: ' . $conn->connect_error;
    exit();
}

// Закриття підключення до БД
$conn->close();

// Відображення таблиці з бронюваннями
display_booking_table($bookings);
}

```

#### Лістинг Г.4 – Код файлу «display.php» у каталозі «admin»

```

<?php
// Відображає таблицю з бронюваннями
function display_booking_table($bookings)
{
    echo '<thead><tr>';
    echo '<th>Час</th>';
    echo '<th>Номер телефону</th>';
    echo '<th>E-Mail</th>';
    echo '<th>Назва ліків</th>';
    echo '<th>Кількість одиниць</th>';
    echo '</tr></thead>';

    // Відображення окремих бронювань
    echo '<tbody>';
    foreach ($bookings as $booking) {
        echo '<tr><td>' .
date_format(date_create($booking['booking_time']), "d.m.Y
H:i:s") . '</td>';
        echo '<td>' . $booking['customer_phone'] . '</td>';
        echo '<td>' . $booking['customer_email'] . '</td>';
        echo '<td>' . $booking['medicine_name'] . '</td>';
        echo '<td>' . $booking['medicine_count'] . '</td></tr>';
    }
    echo '</tbody>';
}

// Відображає таблицю з ліками
function display_medicine_table($medicines)
{
    echo '<thead><tr>';
    echo '<th>Назва</th>';

```

```

echo '<th>Виробник</th>';
echo '<th>Категорія</th>';
echo '<th>Ціна</th>';
echo '<th>URL зображення</th>';
echo '</tr></thead>';

// Відображення окремих ліків
echo '<tbody>';
foreach ($medicines as $medicine) {
    display_medicine_row($medicine);
}
echo '</tbody>';
}

// Відображає окремий рядок з ліками в таблиці
function display_medicine_row($medicine)
{
    echo '<tr id="' . $medicine['id'] . '">';
    echo '<td>' . $medicine['name'] . ' (' .
$medicine['medicine_form'] . ')</td>';
    echo '<td>' . $medicine['producer_name'] . '</td>';
    echo '<td>' . $medicine['category'] . '</td>';
    echo '<td>' . $medicine['price'] . ' грн</td>';
    echo '<td><a href="' . $medicine['picture_file'] . '">' .
$medicine['picture_file'] . '</a></td>';
    echo '<td><a href="medicine/editform.php?id=' .
$medicine['id'] . '">Редагувати</a><br>';
    echo '<a href="#" class="del-btn" data-id="' . $medicine['id']
. '">Видалити</a></td></tr>';
}

```

### Лістинг Г.5 – Код файлу «database.php»

```

<?php
require_once "../../database/connection.php";

// Повертає ID категорії з БД за вказаною назвою
function get_category_id($name)
{
    $conn = connect(); // Підключення до БД

    // SQL-запит
    $query = "SELECT id FROM category WHERE `name` = ?";

    if ($stmt = $conn->prepare($query)) { // Підготовка виразу
        // Встановлення назви категорії до параметрів підготовленого
виразу
        $stmt->bind_param('s', $name);
        $stmt->execute(); // Виконання підготовленого виразу

        $result = $stmt->get_result(); // Результати виконання

```

```

        $id_array = result_array($result)[0]; // Отримання ID
категорії
        $stmt->close(); // Закриття підготовленого виразу
    }
    else {
        echo 'Помилка виконання запиту: ' . $conn->connect_error;
        exit();
    }

    $conn->close(); // Закриття підключення до БД
    return $id_array['id']; // Повернення ID категорії
}

// Повертає ID форми пакування з БД за вказаною назвою
function get_medicine_form_id($name)
{
    $conn = connect(); // Підключення до БД

    // SQL-запит
    $query = "SELECT id FROM medicine_form WHERE `name` = ?";

    if ($stmt = $conn->prepare($query)) { // Підготовка виразу
        // Встановлення назви форми пакування до параметрів
        підготовленого виразу
        $stmt->bind_param('s', $name);
        $stmt->execute(); // Виконання підготовленого виразу

        $result = $stmt->get_result(); // Результати виконання
        $id_array = result_array($result)[0]; // Отримання ID форми
пакування
        $stmt->close(); // Закриття підготовленого виразу
    }
    else {
        echo 'Помилка виконання запиту: ' . $conn->connect_error;
        exit();
    }
    mysqli_close($conn); // Закриття підключення до БД
    return $id_array['id']; // Повернення ID форми пакування
}

// Повертає запис ліків з БД за вказаним ID
function get_medicine_by_id($id)
{
    $conn = connect(); // Підключення до БД

    // SQL-запит до БД
    $query = "SELECT * FROM view_medicine WHERE id = ?";

    if ($stmt = $conn->prepare($query)) { // Підготовка запиту
        // Встановлення ID запису до параметрів підготовленого
виразу
        $stmt->bind_param('i', $id);

```

```

    $stmt->execute(); // Виконання підготовленого виразу
    $result = $stmt->get_result(); // Результати виконання
    $medicine_array = result_array($result); // Отримання
асоціативного масиву
    $stmt->close(); // Закриття підготовленого виразу
}
else {
    echo 'Помилка виконання запиту: ' . $conn->connect_error;
    exit();
}

$conn->close(); // Закриття підключення до БД
return $medicine_array[0]; // Повернення масиву із записом
}

// Додає запис ліків у БД із вказаними полями
function add_medicine($entry)
{
    $conn = connect(); // Підключення до БД

    // SQL-запит для додавання запису ліків у БД
    $query = "INSERT INTO medicine (`name`, producter_name,
category_id,
        medicine_form_id, price, picture_file) VALUES (?, ?, ?,
?, ?, ?)";

    if ($stmt = mysqli_prepare($conn, $query)) { // Підготовка
виразу
        // Поля для додавання нового запису
        $name = $entry['name'];
        $producter = $entry['producter'];
        $category_id = $entry['category-id'];
        $medicine_form_id = $entry['medicine-form-id'];
        $price = $entry['price'];
        $picture = $entry['picture-file'];

        // Встановлення параметрів підготовленого виразу
        $stmt->bind_param('sssiids', $name, $producter, $category_id,
            $medicine_form_id, $price, $picture);

        $stmt->execute(); // Виконання виразу
        $stmt->close(); // Закриття підготовленого виразу
    }
    else {
        echo 'Помилка виконання запиту: ' . $conn->connect_error;
        exit();
    }

    // Закриття підключення до БД
    $conn->close();
}

```

```

// Редагує запис ліків у БД із вказаними полями
function edit_medicine($entry)
{
    $conn = connect(); // Підключення до БД

    // SQL-запит для додавання запису ліків у БД
    $query = "UPDATE medicine SET `name` = ?, producter_name = ?,
category_id = ?,
        medicine_form_id = ?, price = ?, picture_file = ? WHERE
id = ?";

    if ($stmt = $conn->prepare($query)) { // Підготовка виразу
        // Поля для додавання нового запису
        $id = $entry['id'];
        $name = $entry['name'];
        $producter = $entry['producter'];
        $category_id = $entry['category-id'];
        $medicine_form_id = $entry['medicine-form-id'];
        $price = $entry['price'];
        $picture = $entry['picture-file'];

        // Встановлення параметрів підготовленого виразу
        $stmt->bind_param('ssiidsi', $name, $producter,
$category_id,
            $medicine_form_id, $price, $picture, $id);

        $stmt->execute(); // Виконання виразу
        $stmt->close(); // Закриття підготовленого виразу
    }
    else {
        echo 'Помилка виконання запиту: ' . $conn->connect_error;
        exit();
    }

    // Закриття підключення до БД
    $conn->close();
}

// Видаляє запис ліків з БД за вказаним ID
function delete_medicine($id)
{
    $conn = connect(); // Підключення до БД

    // SQL-запит до БД для видалення запису ліків
    $query = "DELETE FROM medicine WHERE id = ?";

    if ($stmt = $conn->prepare($query)) { // Підготовка виразу
        // Встановлення ID до параметрів виразу
        $stmt->bind_param('i', $id);
        $stmt->execute(); // Виконання виразу
        $stmt->close(); // Закриття підготовленого виразу
    }
}

```



```
else {
    echo 'Помилка виконання запиту: ' . $conn->connect_error;
    exit();
}

// Закриття підключення до БД
$conn->close();
}
```

## Знімки екрана демонстрації роботи веб-сервісу

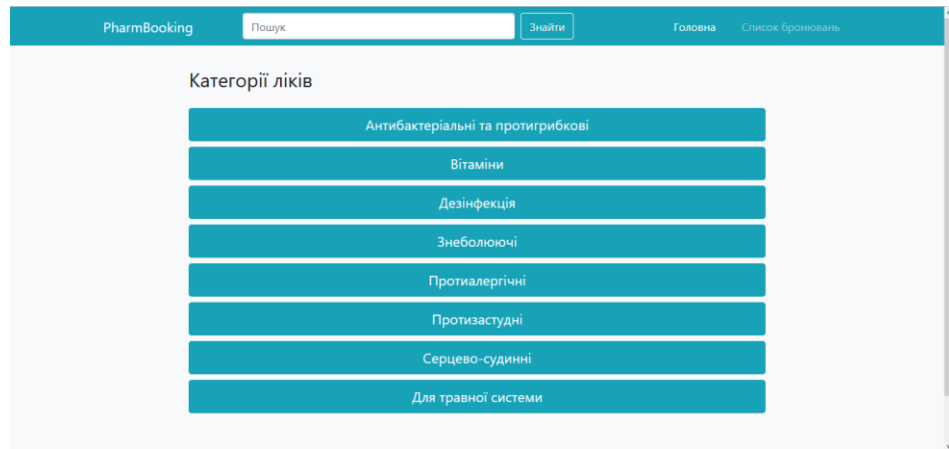


Рисунок Д.1 – Головна сторінка веб-сервісу для бронювання ліків

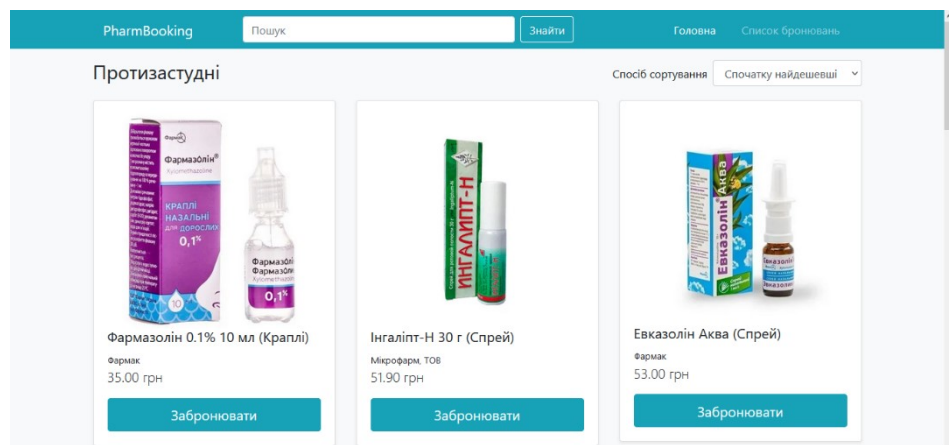


Рисунок Д.2 – Перегляд ліків, належать до категорії протизастудних

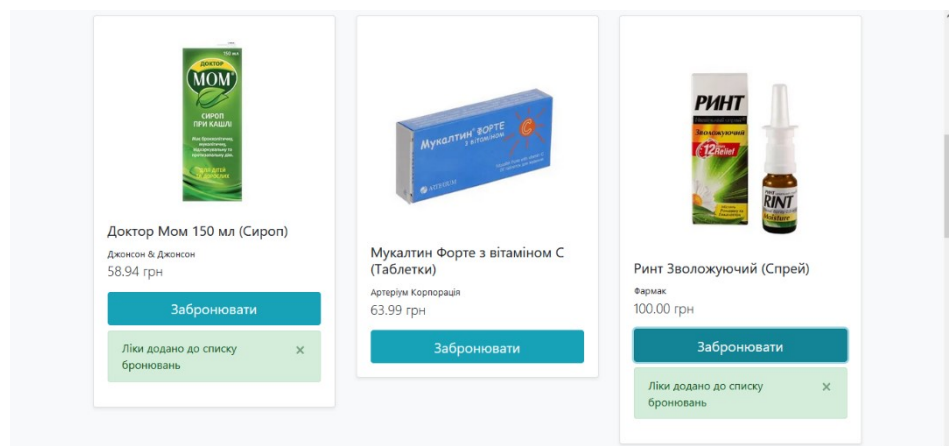


Рисунок Д.3 – Додавання ліків до списку бронювань

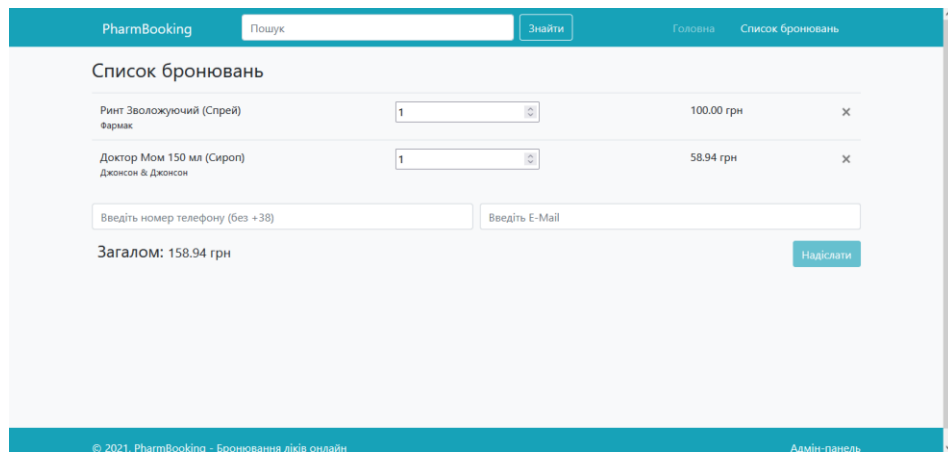


Рисунок Д.4 – Перегляд списку бронювань

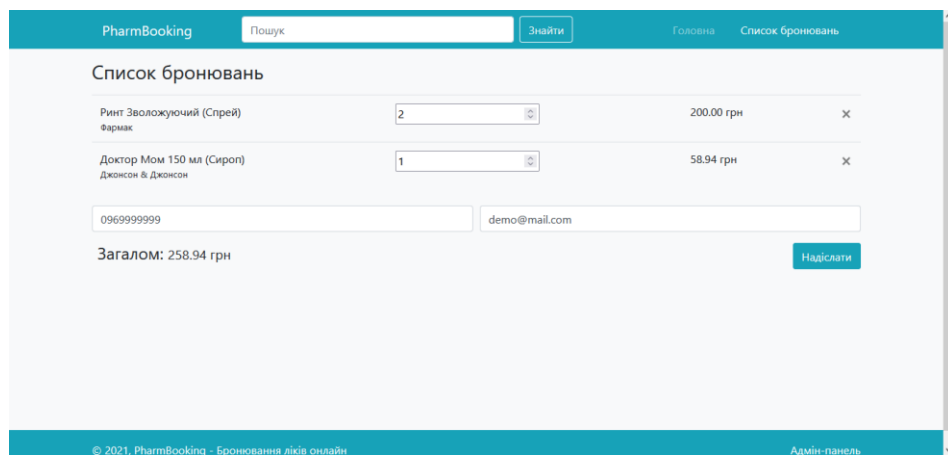


Рисунок Д.5 – Надсилання списку бронювань

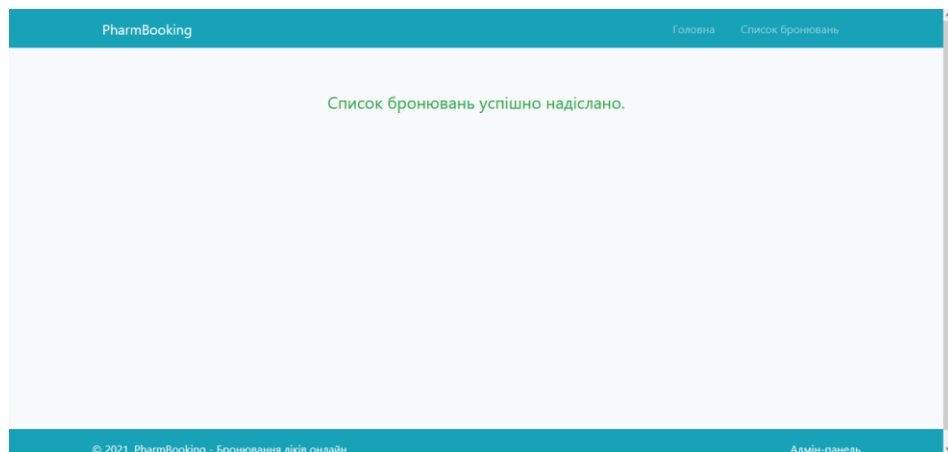


Рисунок Д.6 – Повідомлення про успішне надсилання списку бронювань

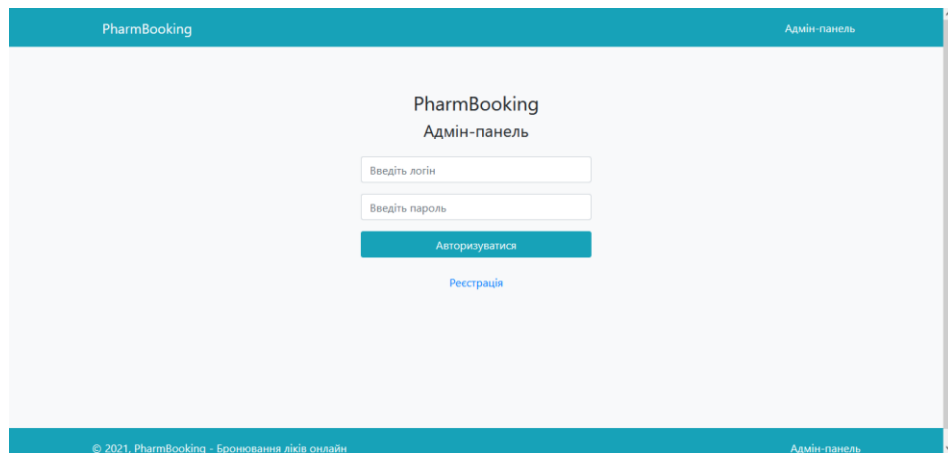


Рисунок Д.7 – Сторінка авторизації до адмін-панелі

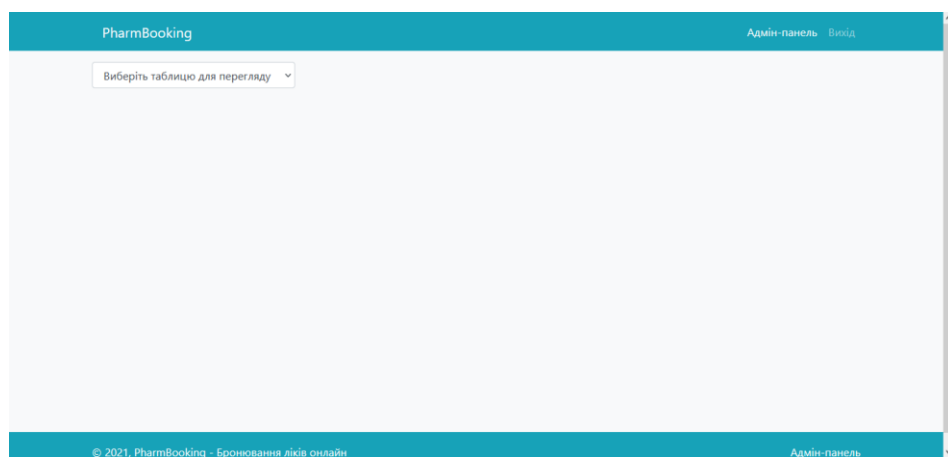


Рисунок Д.8 – Робоча сторінка адмін-панелі

Назва	Виробник	Категорія	Ціна	URL зображення	
Інгаліпт-Н 30 г (Спрей)	Мікрофарм, ТОВ	Протизастудні	51.90 грн	<a href="https://entianis.sivv.com/pharmbooking/ingalpt_n.png">https://entianis.sivv.com/pharmbooking/ingalpt_n.png</a>	Редагувати Видалити
Аміксин ІС 0.125 г (Таблетки)	Інтерхім	Протизастудні	175.00 грн	<a href="https://entianis.sivv.com/pharmbooking/amixin_ic.jpg">https://entianis.sivv.com/pharmbooking/amixin_ic.jpg</a>	Редагувати Видалити
Анальгін-Дарниця 500 мг (Таблетки)	Дарниця	Знеболюючі	10.50 грн	<a href="https://entianis.sivv.com/pharmbooking/analgin_darnyia.jpeg">https://entianis.sivv.com/pharmbooking/analgin_darnyia.jpeg</a>	Редагувати Видалити
АЦЦ Лонг 600 мг (Таблетки)	Сандоз	Протизастудні	137.25 грн	<a href="https://entianis.sivv.com/pharmbooking/acc_long.jpg">https://entianis.sivv.com/pharmbooking/acc_long.jpg</a>	Редагувати Видалити
Доктор Мом 150 мл (Сироп)	Джонсон & Джонсон	Протизастудні	58.94 грн	<a href="https://entianis.sivv.com/pharmbooking/doctor_mom.jpg">https://entianis.sivv.com/pharmbooking/doctor_mom.jpg</a>	Редагувати Видалити
Евказолін Аква (Спрей)	Фармак	Протизастудні	53.00 грн	<a href="https://entianis.sivv.com/pharmbooking/eukasolin_aqua.jpg">https://entianis.sivv.com/pharmbooking/eukasolin_aqua.jpg</a>	Редагувати Видалити
Мікселін Флора з вітаміном С	Дарниця	Протизастудні	63.00	<a href="https://entianis.sivv.com/pharmbooking/mixelin_flora.jpg">https://entianis.sivv.com/pharmbooking/mixelin_flora.jpg</a>	Редагувати Видалити

Рисунок Д.9 – Перегляд таблиці з записами про ліки