

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»

Кафедра інженерії програмного забезпечення

КУРСОВА РОБОТА

(ПОЯСНЮВАЛЬНА ЗАПИСКА)

з дисципліни: «Бази даних»

на тему: «База даних каси автовокзалу»

студента II курсу групи ВТ-22-1
спеціальності 121 «Інженерія програмного
забезпечення»

Середі Івана Романовича
(прізвище, ім'я та по-батькові)

Керівник: ст. викладач кафедри ІІЗ
Чижмотря.О.В.

Дата захисту: " _ " _____ 2024р.

Національна шкала _____

Кількість балів: _____

Оцінка: ECTS _____

Члени комісії

_____	<u>Ольга КОРОТУН</u>
(підпис)	(прізвище та ініціали)
_____	<u>Світлана КРАВЧЕНКО</u>
(підпис)	(прізвище та ініціали)
_____	<u>Олексій ЧИЖМОТНЯ</u>
(підпис)	(прізвище та ініціали)
_____	<u>Інна СУГОНЯК</u>
(підпис)	(прізвище та ініціали)

Житомир – 2024

ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»
Факультет інформаційно-комп'ютерних технологій
Кафедра інженерії програмного забезпечення
Освітній рівень: бакалавр
Спеціальність 121 «Інженерія програмного забезпечення»

«ЗАТВЕРДЖУЮ»

Завідувач кафедри ІІЗ

_____ Т.А. Вакалюк

“ ____ ” _____ 2023 р.

ЗАВДАННЯ
НА КУРСОВУ РОБОТУ СТУДЕНТУ
Середі Івану Романовичу

1. Тема роботи: **База даних каси автовокзалу**,
керівник роботи: ст. викладач кафедри ІІЗ Чижмотря О.В.
2. Строк подання студентом: “28” березень 2024р.
3. Вихідні дані до роботи: Розробити базу даних для сайту автовокзалу
4. Зміст розрахунково-пояснювальної записки(перелік питань. Які підлягають розробці)
 - Аналіз інформаційних потоків та особливостей предметної області дослідження
 - Проектування бази даних за напрямком курсової роботи
 - Реалізація підсистеми обробки даних за напрямком курсової роботи
 - Адміністрування баз даних
5. Перелік графічного матеріалу(з точним зазначенням обов'язкових креслень)
 1. Презентація до КР:
 2. Посилання на репозиторій:
https://gitlab.com/sereda_ivan/kursovaadb_sereda_ivan_vt-22-1
6. Консультанти розділів проекту (роботи)

Розділ	Прізвище, ініціали та посади консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1,2,3	Чижмотря О.В., к.т.н., доц.	15.10.2023	15.10.2023

7. Дата видачі завдання “15” жовтня 2023 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/П	Назва етапів курсового проекту	Строк виконання етапів проекту	Примітки
1	Постановка задачі	10.11.2023	Виконано
2	Пошук, огляд та аналіз аналогічних розробок	20.11.2023	Виконано
3	Формулювання технічного завдання	25.11.2023	Виконано
4	Опрацювання літературних джерел	26.11.2023	Виконано
5	Проектування бази даних	05.01.2024	Виконано
6	Написання програмного коду	20.01.2024	Виконано
7	Відлагодження	20.02.2024	Виконано
8	Написання пояснювальної записки	21.03.2023	Виконано
9	Захист		

Студент

(підпис)

Іван СЕРЕДА
(прізвище та ініціали)

Керівник роботи

(підпис)

Олексій ЧИЖМОТРЯ
(прізвище та ініціали)

РЕФЕРАТ

Завданням на курсову роботу була розробка веб-додатку бази даних до сайту автовокзалу.

Пояснювальна записка до курсового роботи на тему “ База даних каси автовокзалу ” складається з вступу, чотирьох розділів, висновків, списку використаної літератури та додатку.

Текстова частина викладена на 65 сторінках друкованого тексту.

Пояснювальна записка має 88 сторінок додатків. Список використаних джерел містить 10 найменувань і займає 1 сторінку. В роботі наведено 45 рисунків. Загальний обсяг роботи – 153 сторінки.

У першому розділі було обґрунтовано створення сайту на тему “База даних автовокзалу”.

У другому розділі проведено проектування бази даних сайту.

У третьому розділі проведено тестування програмного продукту.

Висновок містить в собі результати виконаної роботи при створенні сайту на тему “База даних сайту автовокзалу”.

Ключові слова: SSMS, T-SQL, PYTHON, FLASK, JINJA2, HTML, CSS, BOOTSTRAP, JAVA SCRIPT, ООП, КЛІЄНТ, КОРИСТУВАЧ, АДМІН, АВТОРИЗАЦІЯ, РОЛІ, ДАНІ, БАЗА ДАНИХ, ЗАПИТ.

					ДУ «Житомирська політехніка».24.121.19.000 - ПЗ			
Змн.	Арк.	№ докум.	Підпис	Дата				
Розроб.	Середа І.Р.				“База даних каси автовокзалу”	Літ.	Арк.	Аркушів
Керівник	Чижмотря О.В						4	148
Н. контр.								
Зав. каф.						ФІКТ Гр. ВТ-22-1[1]		

ЗМІСТ

ВСТУП.....	7
РОЗДІЛ 1. ТЕОРЕТИЧНИЙ АНАЛІЗ ІНФОРМАЦІЙНИХ ПОТОКІВ ТА 9	9
ОСОБЛИВОСТЕЙ ПРЕДМЕТНОЇ ОБЛАСТІ ДОСЛІДЖЕННЯ	9
1.1 Аналіз інформаційних потреб та визначення предметної області дослідження.....	9
1.2 Архітектура та засоби реалізації бази даних	14
Висновки до першого розділу:	18
РОЗДІЛ 2. ПРОЕКТУВАННЯ БАЗИ ДАНИХ САЙТУ АВТОВОКЗАЛУ	20
2.1 Аналіз структури інформаційних процесів адміністраторів.....	20
2.2 Проектування бази даних сайту оголошень	21
Висновки до другого розділу.....	27
РОЗДІЛ 3. РЕАЛІЗАЦІЯ САЙТУ АВТОВОКЗАЛУ	28
3.1. Проектування інтерфейсу системи сайту автовокзалу	28
3.2. Реалізації операцій обробки даних в БД	44
Висновок до третього розділу	54
РОЗДІЛ 4. АДМІНІСТРУВАННЯ БАЗ ДАНИХ	55
4.1. Розробка заходів захисту інформації в БД.....	55
4.2. Налаштування параметрів роботи Sql Server.....	61
Список використаних джерел	65
ДОДАТКИ.....	66

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижморя О.В.				5
Змн.	Арк.	№ докум.	Підпис	Дата		

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ:

ТЗ – Технічне завдання

СУБД – система управління базою даних

CRUD – create, read, update, delete.

ПЗ – Програмне забезпечення

Рис – рисунок

СУБД – система управління базою даних

БД - база даних

АР – адмін панель

ООП – об’єкто орієнтоване програмування

JS – мова програмування Java Script

МП – мова програмування

BT - Bootstrap інструмент для використання шаблонів стилів сторінок та скриптів.

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижмотря О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		6

ВСТУП

Актуальність теми. Сучасний світ стрімко розвивається, і разом з ним зростає потреба в зручних та економних рішеннях для повсякденних завдань. Одним із таких рішень є веб-додатки, які полегшують життя користувачам у багатьох сферах, включаючи бронювання квитків на транспорт.

Веб-додаток для купівлі квитків на автобус – це зручний інструмент, який дозволяє:

- Економити час: Не потрібно йти на автовокзал або стояти в черзі, щоб купити квиток.
- Планувати подорож заздалегідь: Можна купити квиток на будь-який рейс, на будь-яку дату, з будь-якого місця.
- Обирати місце в автобусі: Більшість веб-додатків дозволяють заздалегідь вибрати місце, яке вам подобається.
- Оплачувати квиток онлайн: Це безпечний і зручний спосіб оплати, який не потребує готівки.

Веб-додаток для купівлі квитків на автобус – це універсальне рішення, яке підходить людям будь-якого віку. Молодь цінує його за зручність та економію часу, а люди похилого віку – за можливість не виходити з дому, щоб купити квиток.

Я вважаю, що веб-додатки для купівлі квитків на автобус – це майбутнє транспортної системи. Їх зручність, економність та доступність роблять їх все більш популярними, і я впевнений, що в найближчому майбутньому вони стануть основним способом бронювання квитків на автобуси.

Власна думка:

Я особисто користуюся веб-додатками для купівлі квитків на автобус вже протягом декількох років. І я можу з упевненістю сказати, що це значно полегшило мені життя. Я більше не витрачаю час на черги, можу заздалегідь спланувати подорож і вибрати місце, яке мені подобається.

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижмотря О.В.				7
Змн.	Арк.	№ докум.	Підпис	Дата		

Я рекомендую всім, хто ще не користувався веб-додатками для купівлі квитків на автобус, спробувати їх. Ви будете приємно здивовані зручністю та економністю цього способу бронювання квитків.

Мета розробки. Метою розробки данного курсового проекту є створення сайту, написаного на таких мовах програмування як Python на фреймворку Flask та JS з використанням об'єктно орієнтованого програмування, створення бази даних для збереження інформації про подорожі, квитки та користувачів.

Завданням курсової роботи є розробка веб-додатку та бази даних автовокзалу.

Веб-додаток створюється з метою комфорту замовлення та купівлі квитків на автобуси.

Об'єктом дослідження є технологія розробки класів на МП Python які є схожими на таблиці БД на базі T-SQL та СУБД Microsoft SQL Server Management Studio, запис інформації в ці об'єкти та подальше їх використання за правилами ООП. А також використання МП JS для певних дій на самому сайті таких як показ маршруту на Google Map.

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижмотря О.В.				8
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 1. ТЕОРЕТИЧНИЙ АНАЛІЗ ІНФОРМАЦІЙНИХ ПОТОКІВ ТА ОСОБЛИВОСТЕЙ ПРЕДМЕТНОЇ ОБЛАСТІ ДОСЛІДЖЕННЯ

1.1 Аналіз інформаційних потреб та визначення предметної області дослідження

В даному курсовому проекті за предметну область взято базу даних, яка буде зберігати усю необхідну інформацію для сайту автовокзалу. Цей сайт надає можливість купівлі квитків онлайн, перегляду певного маршруту поїздки, авторизації користувача та закріплення за ним куплених квитків для подальшого їх перегляду. Також, на сайті є можливість купівлі квитків неавторизованим клієнтам.

Клієнти матимуть можливість купівлі квитків які відповідають певним параметрам, та подальшого їх перегляду. Для збереження історії покупок клієнт може зареєструватись, але й для користувачів є можливість покупки квитків без реєстрації. Для адміністраторів, потрібно зробити можливість видалення, додавання, оновлення даних. Для цього потрібно створити базу даних в якій буде зберігатись інформація, а для операцій з цими даними розробити адмінпанель на сайті.

База даних має бути зручною для користувачів. Це означає, що інтерфейс користувача (UI) має бути простим, інтуїтивно зрозумілим та навігабельним. Користувачі повинні мати можливість легко знаходити потрібну їм інформацію, а також додавати, видаляти та редагувати дані без зайвих зусиль.

Швидкість є однією з найважливіших характеристик бази даних. Користувачі очікують, що запити будуть оброблятися швидко, а дані – видаватися без затримок. Швидкість роботи бази даних залежить від декількох факторів, таких як апаратне забезпечення, програмне забезпечення та структура бази даних.

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижмотря О.В.				9
Змн.	Арк.	№ докум.	Підпис	Дата		

Надійність. База даних повинна бути надійною, щоб дані не втрачалися і не пошкоджувалися. Це означає, що база даних має бути захищена від збоїв живлення, апаратних помилок та несанкціонованого доступу.

Масштабованість. База даних повинна бути масштабованою, щоб вона могла рости разом з вашими потребами. Це означає, що база даних має бути здатною обробляти більше даних та користувачів без значного зниження продуктивності.

Безпека. База даних повинна бути захищеною від несанкціонованого доступу, використання, розкриття, зміни або знищення. Це означає, що база даних має мати відповідні механізми аутентифікації, авторизації та шифрування.

Сумісність. База даних повинна бути сумісною з іншими системами та програмним забезпеченням. Це означає, що база даних має бути здатною обмінюватися даними з іншими системами без проблем.

Оновлення. Дані в базі даних повинні постійно оновлюватися, щоб залишатися актуальними. Це означає, що база даних має мати механізми для імпорту та експорту даних, а також для синхронізації даних з іншими системами.

Операції з даними. Для коректного виконання запитів на отримання даних, дані повинні вноситись, видаляться та змінюватись у БД. За це відповідають такі операції:

Реєстрація користувача

Додаток повинен розрізняти користувачів та адміністраторів, щоб надати перелік певних функцій.

Перелік дій які доступні користувачу :

- Перегляд інформації
- Купівля квитка
- Перегляд маршруту

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижомотря О.В.				10
Змн.	Арк.	№ докум.	Підпис	Дата		

- Перегляд куплений квитків

Перелік дій які доступні адміністратору :

- Перегляд інформації
- Купівля квитка
- Перегляд маршруту
- Перегляд куплений квитків
- Доступ до адмінпанелі
- Редагування даних
- Видалення даних

Для цього потрібно створити таблицю з усіма користувачами як сайту так бази даних.

Вхід користувача

Авторизація користувачів після реєстрації.

Роль адміністратор

Надає можливість користуватись функціями описаними вище.

Видалення та додавання маршрутів, автобусів, міст

Необхідно для того щоб додавати та видаляти інформацію про нові подорожі

Редагування новини

Необхідно для зміни інформації про маршрути, автобуси, користувачів і тд. У випадку коли інформація може бути некоректною чи потрібно її замінити на нову.

Перегляд подорожей

Щоб купити квиток користувач матиме можливість переглянути вигідних превізників а також маршрутів безпосередньо на картах

Фільтрація

Щоб знайти певний маршрут у користувача є можливість відфільтрувати поданий список та віднайти потрібний маршрут.

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижмотря О.В.				11
Змн.	Арк.	№ докум.	Підпис	Дата		

Системні повідомлення

Щоб проінформувати користувача про помилку чи про вдалі події.

Пошук за номером квитку

Для комфортності користувача він матиме змогу ввести код квитка та швидко його переглянути.

Конкуренти:

Найближчі конкуренти схожі на додаток який розробляється це inBus та LikeBus.

inBus (<https://inbus.ua>)

Великий веб-додаток для пошуку маршрутів автобусів працюючий як в межах України так і надає інформацію про перевезення з України в країни Європи. Сайт має досить привабливий сайт та зручний зрозумілий інтерфейс та живий пошук по напрямках. Мають можливість звернутись на інформаційну лінію.

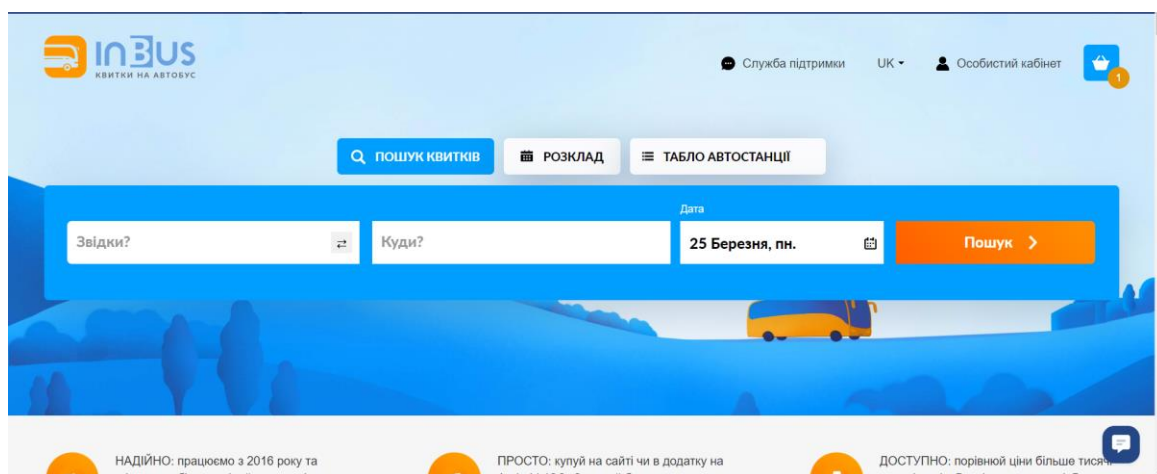


Рис.1.1. Вигляд головної сторінки на сайті inBus

З мінусів можу відмітити баги при виборі міста та не можливість вибрати період годин відправлення, а також досить довгий пошук за маршрутами і в залагом довге завантаження інших сторінок. За декілька спроб оформлення квитка були помилки такого типу. Деякі перевізники не надають можливість вибору місця. Немає фільтрації за обширнішими параметрами.

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижмотря О.В.				12
Змн.	Арк.	№ докум.	Підпис	Дата		

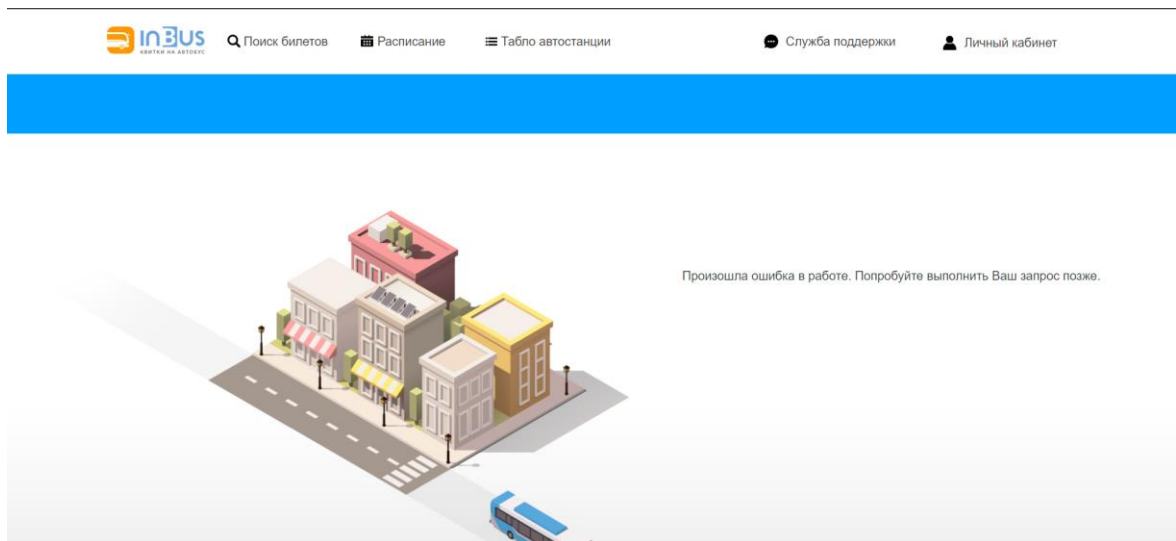


Рис.1.2.Помилка при оформленні квитка

Рис.1.3. Оформлення купівлі квитка в inBus

LikeBus (<https://likebus.ua/>)

Сайт має презентабельний вид та чудову швидкість роботи, приємний для використання інтерфейс. А також можливість зв'язатись в чаті за додатковою інформацією, вибір додаткового багажу.

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижмотря О.В.				13
Змн.	Арк.	№ докум.	Підпис	Дата		

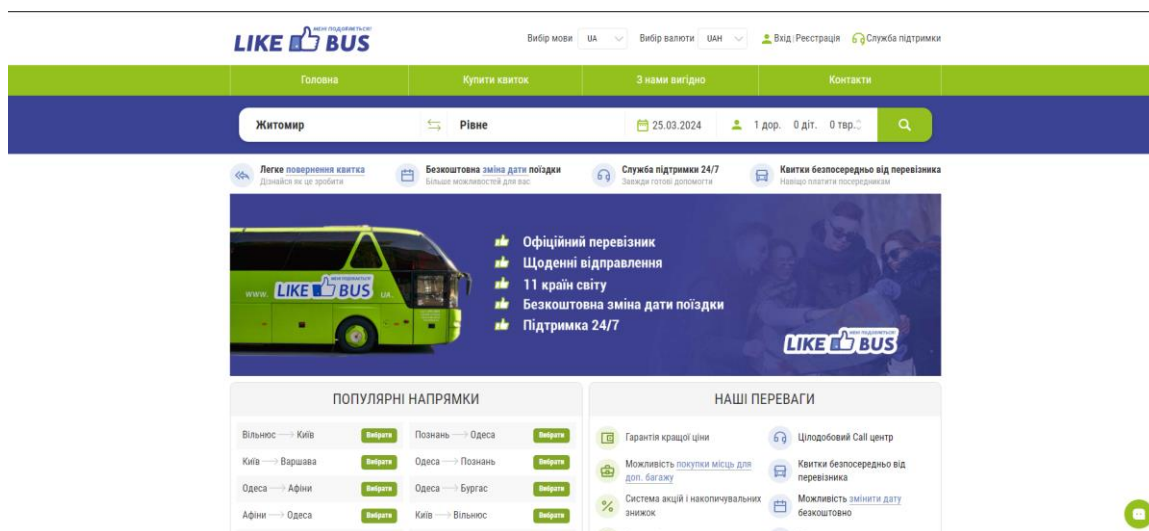


Рис.1.4. Головна сторінка LikeBus

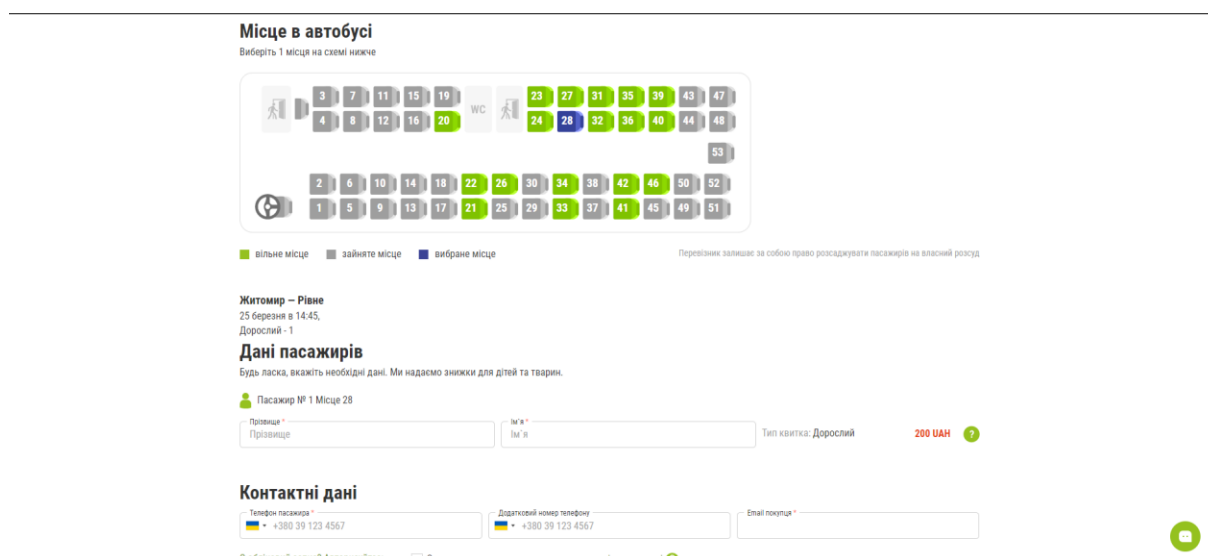


Рис.1.5. Оформлення квитка на LikeBus

З мвнусів можу помітити нагромадженість елементами інтерфейсу в яких можна загубитись. Немає фільтрації за обширнішими параметрами.

1.2 Архітектура та засоби реалізації бази даних

Для проекту, який очікує велику кількість користувачів, важливо ретельно підібрати СУБД (систему управління базами даних), щоб забезпечити максимально ефективний доступ до даних. Цей документ описує порівняльний аналіз трьох популярних СУБД: Microsoft SQL Server, Oracle та Firebase.

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижморя О.В.				14
Змн.	Арк.	№ докум.	Підпис	Дата		

Таблиця 1.1.

Функціональне порівняння СУБД

Характеристика	Microsoft SQL Server	Oracle	Firebase
Робота під керівництвом ОС	Задовільно	Добре	Відмінно
Складність	Відмінно	Задовільно	Добре
Доступ до даних	Відмінно	Добре	Добре
Використання у великих проектах	Відмінно	Відмінно	Добре
Використання у малих проектах	Добре	Добре	Відмінно
Схема даних	Відмінно	Відмінно	Добре
Підтримувані мови програмування	Відмінно	Добре	Задовільно
Підключення до Web	Відмінно	Відмінно	Відмінно
Графічні	Добре	Добре	Добре
Тригери	Відмінно	Відсутній критерій	Добре
Паралелізм	Відмінно	Відмінно	Відмінно
Одночасний доступ декількох користувачів	Добре	Добре	Відмінно
Обробка даних мультимедіа	Слабо	Відмінно	Відмінно

Пошук за регулярними виразами	Відмінно	Відмінно	Задовільно
Підтримка об'єктно-орієнтованої парадигми	Добре	Відмінно	Задовільно
Засоби аналізу	Відмінно	Відмінно	Відмінно

Пояснення таблиці:

Microsoft SQL Server та Oracle це реляційні БД, а Firebase використовує сховище документів

1. Робота під керівництвом ОС:

- Microsoft SQL Server: Windows, Linux.
- Oracle: Windows, Linux, Solaris, HP-UX, OS X.
- Firebase: Не потребує розміщення.

2. Складність:

- Microsoft SQL Server: Легко освоїти, але складні задачі потребують певного досвіду.
- Oracle: Складний для вивчення та потребує знання SQL.
- Firebase: Простий у використанні, але має обмежені можливості.

3. Доступ до даних:

- Microsoft SQL Server: Гнучкий контроль доступу на основі ролей.
- Oracle: Повністю відкритий доступ до даних для користувачів.
- Firebase: Контроль доступу на основі правил Firebase.

4. Використання у великих проектах:

- Microsoft SQL Server: Масштабований, висока продуктивність.
- Oracle: Ідеально підходить для складних систем з високими навантаженнями.

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижморя О.В.				16
Змн.	Арк.	№ докум.	Підпис	Дата		

- Firebase: Не рекомендується для проектів з великим об'ємом даних.

5. Використання у малих проектах:

- Microsoft SQL Server: Складний для налаштування.
- Oracle: Зайве громіздкий для малих проектів.
- Firebase: Простий у налаштуванні та підключенні.

6. Схема даних:

- Microsoft SQL Server: Фіксована схема, реляційна модель.
- Oracle: Фіксована та безсхемна схема, реляційна модель.
- Firebase: Не використовує схеми, модель NoSQL.

7. Підтримувані мови програмування:

- Microsoft SQL Server: Широкий спектр мов, C#, Java, Python etc.
- Oracle: C, C#, Go, Java, JavaScript, Python.
- Firebase: C, JS, Objective-C.

8. Графічні інструменти:

- Microsoft SQL Server: SSMS (SQL Server Management Studio).
- Oracle: SQL Developer Data Modeler.
- Firebase: Firebase Console.

9. Тригери:

- Microsoft SQL Server: Повністю підтримуються.
- Oracle: Немає підтримки.
- Firebase: Автоматичне відстеження змін.

10. Паралелізм:

- Microsoft SQL Server: Висока ступінь паралелізму.
- Oracle: Масштабований паралелізм.
- Firebase: Повністю підтримується.

11. Одночасний доступ декількох користувачів:

- Microsoft SQL Server: Масштабований, залежить від ліцензії.
- Oracle: Висока пропускна здатність.

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижмоторя О.В.				17
Змн.	Арк.	№ докум.	Підпис	Дата		

- Firebase: Без обмежень, залежить від тарифу.

12. Обробка даних мультимедіа:

- Microsoft SQL Server: Складне налаштування.
- Oracle: Вбудована підтримка.
- Firebase: Просте зберігання та обробка.

13. Пошук за регулярними виразами:

- Microsoft SQL Server: Повністю підтримується.
- Oracle: Повністю підтримується.
- Firebase: Обмежена підтримка.

14. Засоби аналізу:

- Microsoft SQL Server: SSRS (SQL Server Reporting Services).
- Oracle: Oracle Business Intelligence Suite.
- Firebase: Firebase Analytics (вбудовані).

Висновок:

- Microsoft SQL Server: Ідеально підходить для великих проектів, де потрібна висока масштабованість, доступність, безпека та аналітика.
- Oracle: Рекомендується для складних систем з високими навантаженнями, де потрібна максимальна продуктивність та обробка даних мультимедіа.
- Firebase: Найкращий вибір для швидкої розробки додатків, особливо для малих та середніх проектів. Firebase відрізняється простотою налаштування та використання, але має обмеження щодо масштабованості складних запитів та підтримки об'єктно-орієнтованої парадигми.

Висновки до першого розділу:

Конкурентоспроможність

На ринку проект буде вигравати в конкурентів більшими можливостями за рахунок особливостей яких немає в конкурентів.

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижмотря О.В.				18
Змн.	Арк.	№ докум.	Підпис	Дата		

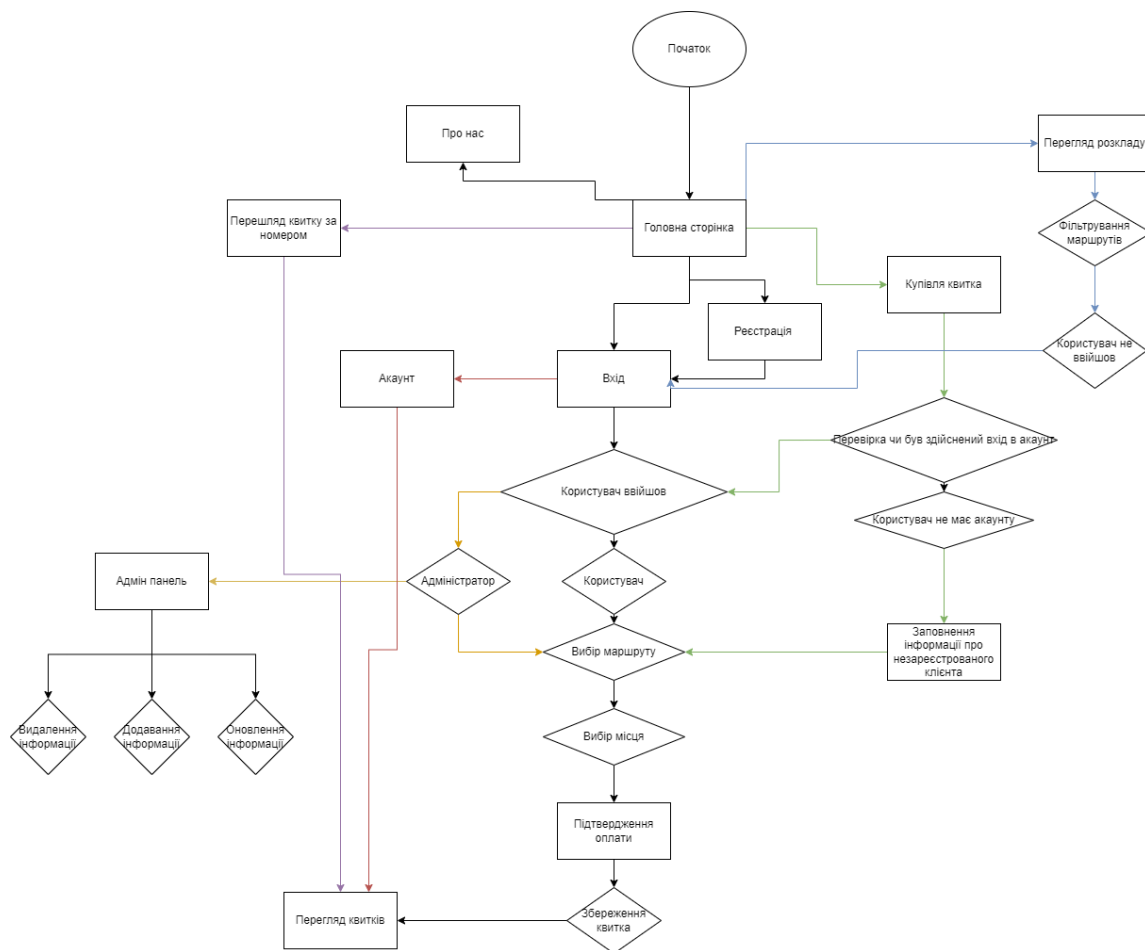
Архітектура

Виходячи з потреб конструювання та комфорту як програмування так використання для розробки найбільше підходить БД Microsoft SQL Server та СУБД Server SQL Menegment Studio. Вона безпечна, масштабуєма, високо продуктивна, не працює з медіа, дозволяє надавати ролі для доступу до даних що в рамках веб-додатку автовокзалу практично ідеально підходить.

Мова програмування вибрана Python так як це доволі сильна, проста, зрозуміла та швидка МП. Для ядра серверу вибрано фреймворк Flask так як він є простим в реалізації але й разом з цим доволі потужним.

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижмотря О.В.				19
Змн.	Арк.	№ докум.	Підпис	Дата		

2.1 Аналіз структури інформаційних процесів адміністраторів



Користувачі будуть мати змогу працювати з веб-додатком як після аутентифікації, так і без неї. Наприклад переглядати подорожі та їх напрямки, перевірку квитків.

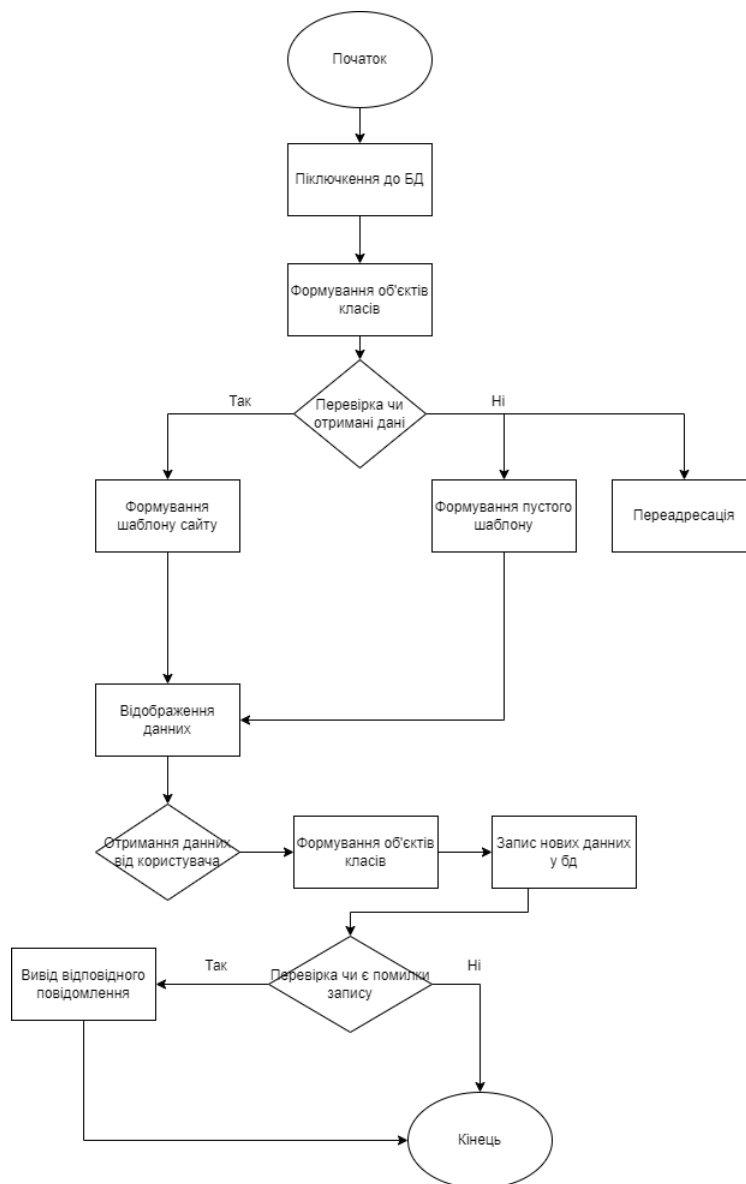


Рис.2.2. Структурна схема життєвого циклу запиту

2.2 Проектування бази даних сайту оголошень

Після формування функціоналу додатку було створено базу даних. Для проектування було використано SSMS.

Список спроектованих таблиць

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижмотря О.В.				21
Змн.	Арк.	№ докум.	Підпис	Дата		








- +  dbo.buses
- +  dbo.cities
- +  dbo.clients
- +  dbo.non_authorized_users
- +  dbo.orders
- +  dbo.tickets
- +  dbo.timetable

Рис.2.3. Список створених таблиць





















- +  dbo.delete_null_journey
- +  dbo.delete_null_tickets
- +  dbo.delete_order
- +  dbo.delete_orders
- +  dbo.deleteBus
- +  dbo.deleteCity
- +  dbo.deleteClient
- +  dbo.deleteNonAthUser
- +  dbo.deleteOrder
- +  dbo.deleteTicket
- +  dbo.deleteTimetable
- +  dbo.drop_FK_orders_ticket
- +  dbo.drop_FK_tickets_timetable
- +  dbo.drop_FK_timetable_buses
- +  dbo.get_all_tickets_per_day
- +  dbo.get_all_tickets_per_month
- +  dbo.get_all_tickets_per_year
- +  dbo.get_income_per_day
- +  dbo.get_income_per_month
- +  dbo.get_income_per_year

Рис.2.4. Список створених процедур

Внаслідок проектування до бази даних включено таблиці:

- buses
- cities
- clients
- non_authorized_users
- orders

- tickets
- timetable

Ще була створена діаграма для комфорту представлення зв'язків

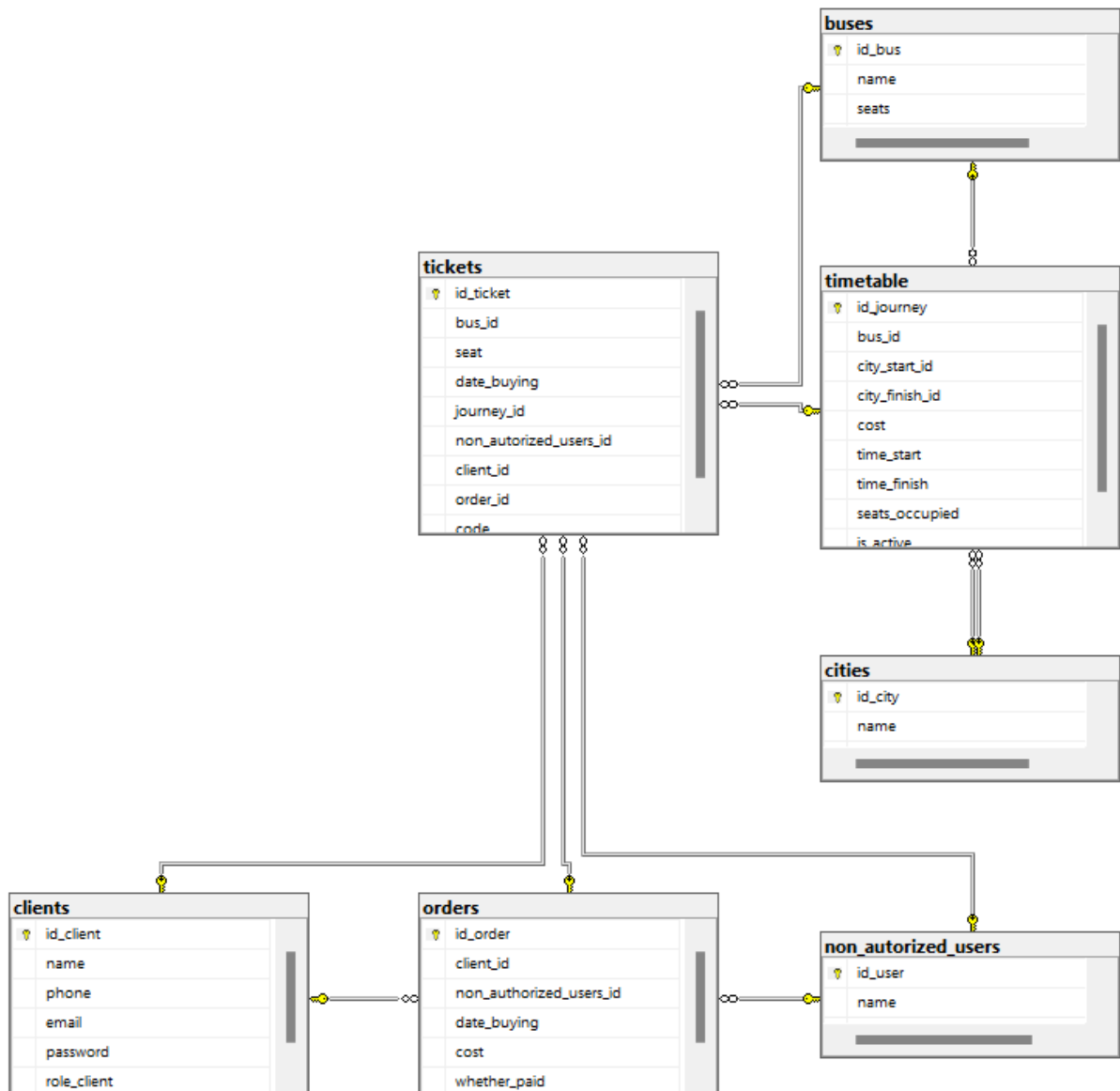


Рис.2.5. Логічна ERD діаграма

Для збереження інформації про автобус була створена таблиця “buses”. В данну таблицю вводяться всі необхідні дані про автобус такі як назва, кількість місць, номер.

Структура таблиці наведена нижче:

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижморя О.В.				23
Змн.	Арк.	№ докум.	Підпис	Дата		

Таблиця 2.1. Структура таблиці "buses"

Назва	Тип даних	ПК	ЗК	Опис поля
id_bus	int	+	-	Id автобуса
name	nvarchar	-	-	Назва автобуса
seats	int	-	-	Кількість місць
bus_number	nvarchar	-	-	Номер автобуса

Для збереження інформації про міста була створена таблиця "cities". В данну таблицю вводяться всі необхідні дані про місто такі як назва та країна.

Структура таблиці наведена нижче:

Таблиця 2.2. Структура таблиці "cities"

Назва поля	Тип даних	ПК	ЗК	Опис поля
id_city	int	+	-	Id міста
name	nvarchar	-	-	Назва
country	nvarchar	-	-	Країна

Для збереження інформації про клієнтів була створена таблиця "clients". В данну таблицю вводяться всі необхідні дані такі як ім'я, номер телефону, email, пароль та роль клієнта яка визначає користувач це чи адмін.

Структура таблиці наведена нижче:

Таблиця 2.3. Структура таблиці "clients"

Назва поля	Тип даних	ПК	ЗК	Опис поля
id_client	int	+	-	Id клієнта
name	nvarchar	-	-	Ім'я
phone	nvarchar	-	-	Номер телефону
email	nvarchar	-	-	Ел.адреса
password	nvarchar	-	-	Пароль
role_client	nvarchar	-	-	Роль клієнта

Для збереження інформації про не зареєстрованих клієнтів була створена таблиця “non_authorized_clients”. В данну таблицю вводяться всі необхідні дані такі як ім’я, номер телефону.

Структура таблиці наведена нижче:

Таблиця 2.4. Структура таблиці " non_authorized_clients"

Назва поля	Тип даних	ПК	ЗК	Опис поля
id_user	int	+	-	Id користувача
name	nvarchar	-	-	Ім'я
phone	nvarchar	-	-	Номер телефону

Для збереження інформації про поїздки була створена таблиця “timetable”. В данну таблицю вводяться всі необхідні дані такі як id автобусу, id мвста відправлення, id міста прибуття, вартість квитка, час відправки та прибуття, кількість зайнятих місць та чи активна дана подорож.

Структура таблиці наведена нижче:

Таблиця 2.5. Структура таблиці " timetable "

Назва поля	Тип даних	ПК	ЗК	Опис поля
id_journey	int	+	-	Id маршруту
bus_id	int	-	-	Id автобуса
city_start_id	int	-	+	Id міста відправлення
city_finish_id	int	-	+	Id міста прибуття
cost	float	-	-	Вартість
time_start	datetime	-	-	Час відправлення
time_finish	datetime	-	-	Час прибуття
seats_occupied	int	-	-	Кількість зайнятих місць
is_active	bit	-	-	Чи активний маршрут

Для збереження інформації про оплату була створена таблиця “orders”. В данну таблицю вводяться всі необхідні дані такі як id клієнта, id не авторизованого клієнта, час купівлі, вартість транзакції, і чи оплачено.

Структура таблиці наведена нижче:

Таблиця 2.6. Структура таблиці "orders"

Назва поля	Тип даних	ПК	ЗК	Опис поля
id_order	int	+	-	Id замовлення
client_id	int	-	+	Id клієнта
non_authorized_users_id	int	-	+	Id незареєстрованого користувача
date_buying	datetime	-	-	Дата оплати
cost	float	-	-	Вартість
whether_paid	bit	-	-	Оплачено чи ні

Для збереження інформації про квиток було створено таблицю “tickets”. В данну таблицю вводяться всі необхідні дані такі як id автобуса, вибране місце, час купівлі, id подорожі, id не авторизованого користувача, id клієнта, id оплати, код квитка.

Структура таблиці наведена нижче:

Таблиця 2.7. Структура таблиці "tickets"

Назва поля	Тип даних	ПК	ЗК	Опис поля
id_ticket	int	+	-	Id квитка
bus_id	int	-	+	Id автобуса
seat	int	-	-	Номер місця
date_buying	datetime	-	-	Дата покупки
journey_id	int	-	+	Id маршруту
non_authorized_users_id	int	-	+	Id незареєстрованого користувача

client_id	int	-	+	Id клієнта
order_id	int	-	+	Id замовлення
code	int	-	-	Код квитка

Висновки до другого розділу :

У результаті виконання даного розділу було проаналізовано інформаційні процеси, спроектовано структуру бази даних, описано приблизну логіку функцій для роботи з базою на клієнті, поисано деяку логіку обробки, збереження та генерації даних для бази. Також було описано як буде відбуватись побудова та генерація статистичної інформації про сайт. Крім цього, було описано логіку взаємодії між даними

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижмотря О.В.				27
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 3. РЕАЛІЗАЦІЯ САЙТУ АВТОВОКЗАЛУ

3.1. Проектування інтерфейсу системи сайту автовокзалу

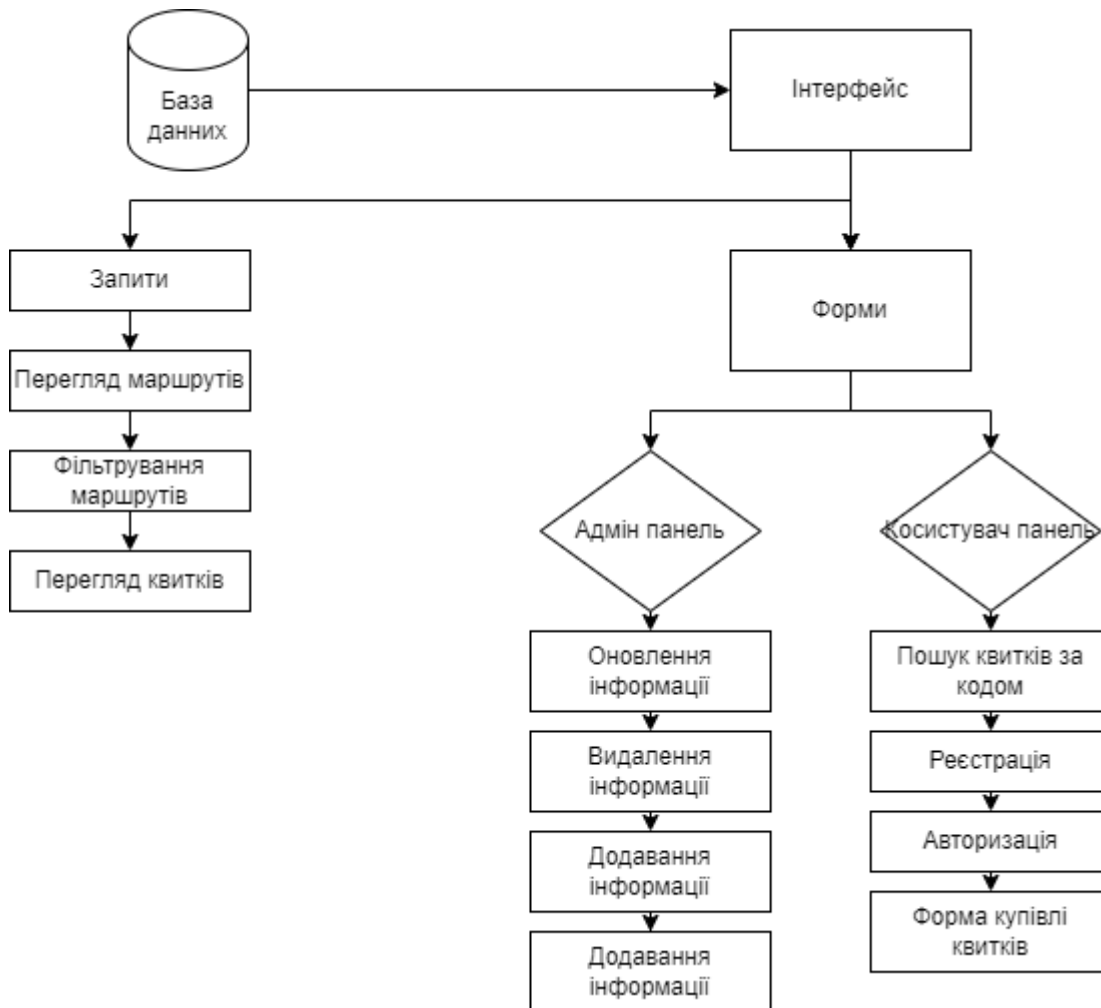


Рис.3.1. Структура інтерфейсу

На рисунку 11 зображено загальну схему інтерфейсу сайту автовокзалу. Далі будуть зображені форми та сторінки, що показують фінальний результат виконання проектування інтерфейсу веб-сайту. Форми супроводжуються кодом виконання та запису даних в БД.

Для розробки інтерфейсу було використано мову гіпертекстової розмітки – HTML, стилі – CSS, а також інструмент Bootstrap.

Форми та сторінки інтерфейсів:

Відкриваємо наш сайт за адресою: <http://127.0.0.1:5000/home> . При відкритті сайту нас зустрічає головна сторінка. На головній сторінці ми можемо помітити декілька посилань на інші сторінки, такі як “Головна”, “Розписання всіх автобусів”, “Купити квитки”, “Про нас”, форма для пошуку квитків за кодом, та блок з авторизацією. Вміст самої сторінки та посилання на відповідні сторінки.

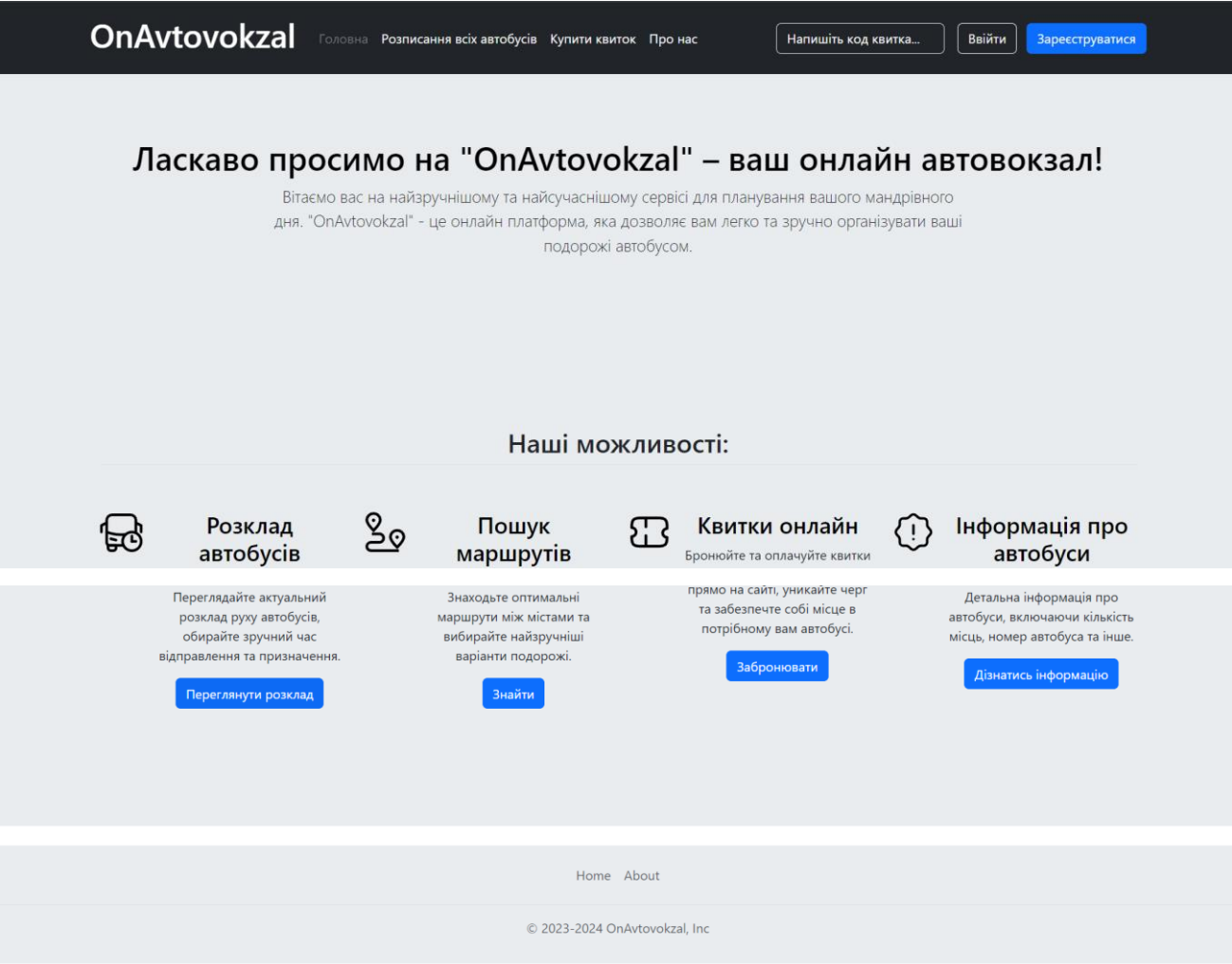


Рис.3.2. Головна сторінка сайту

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижмотря О.В.				29
Змн.	Арк.	№ докум.	Підпис	Дата		

Реєструємось на сайті. Форма реєстрації приймає ім'я та прізвище, пошту, номер телефону, пароль та повторний пароль. Перед відправкою форми на сервер відбувається перевірка того, чи співпадають паролі та чи існує користувач із такою самою поштою, в випадку неспівпадіння, користувача повідомляють про це і попросять змінити дані. Після реєстрації користувач автоматично входить в акаунт.

OnAvtovokzal

[Головна](#) [Розписання всіх автобусів](#) [Купити квиток](#) [Про нас](#)

Напишіть код квитка...

Ввійти

Зареєструватися

Реєстрація

Ім'я та прізвище

Введіть email

Введіть телефон

Пароль

Повторіть пароль

Зареєструватися

©2024

Home

About

Рис.3.3. Форма для реєстрації користувача в систему

При спробі входу появляється вікно з параметрами входу: телефон, email. Відповідно користувач вибирає за чим входити і надає данні. Після успішного входу користувача переадресовують на головну сторінку та змінюється блок авторизації, що показано на рисунку 17. При невдалом вході показується помилка та переадресовується для авторизації заново, представлено на рисунку 18.

OnAvtovokzal

[Головна](#) [Розписання всіх автобусів](#) [Купити квиток](#) [Про нас](#)

Напишіть код квитка...

Ввійти

Зареєструватися

Через email

Ввійти

Через телефон

Home

About

© 2023-2024 OnAvtovokzal, Inc

Рис.3.4. Вибір за якими параметрами будемо входити в акаунт

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижмотря О.В.				30
Змн.	Арк.	№ докум.	Підпис	Дата		

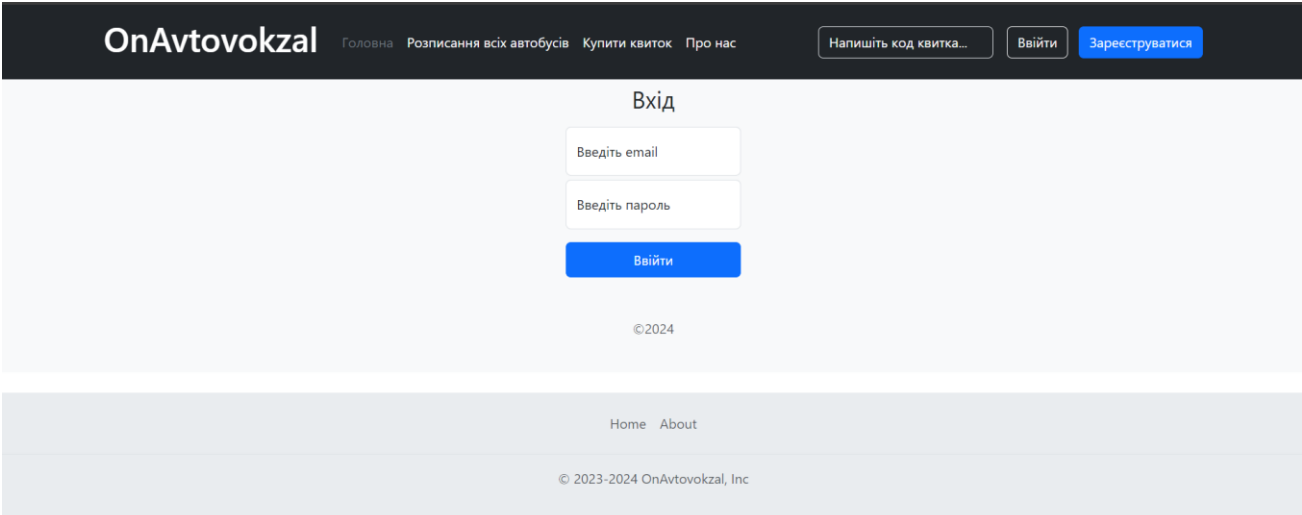


Рис.3.5. Сторінка авторизації через email

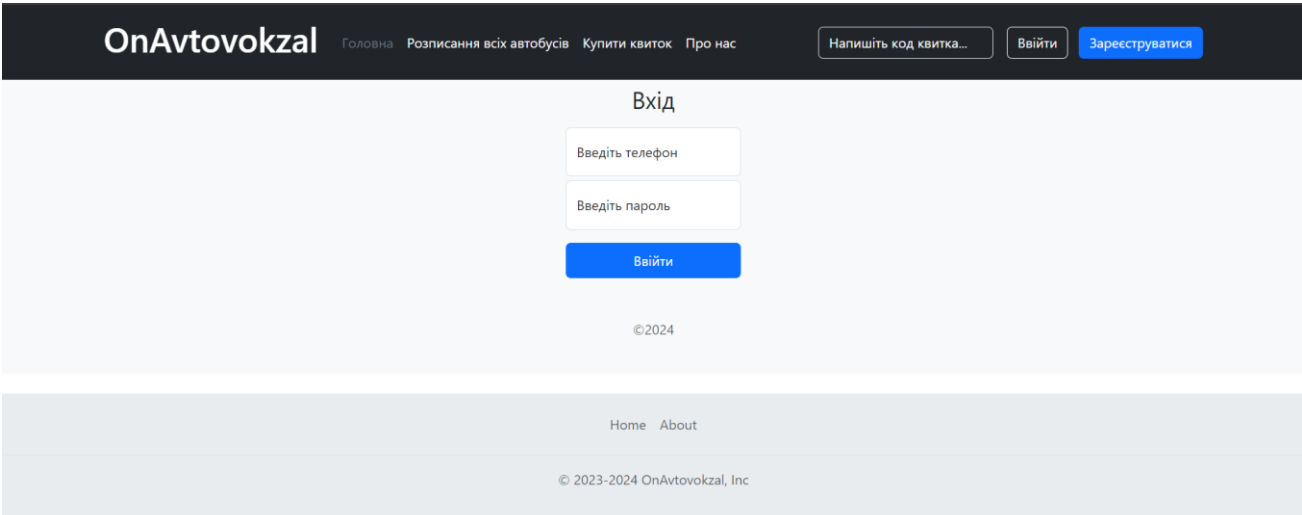


Рис.3.6. Сторінка авторизації через телефон

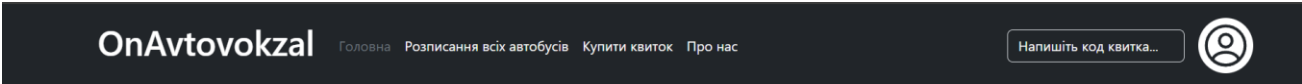


Рис.3.7. Успішний вхід



Рис.3.8. Вхід невдалий

На сторінці “Розписання всіх автобусів” ми можемо переглянути табло всіх маршрутів, яке формується за допомогою запитів до БД та отримання інформації. У випадку коли користувач не авторизований сайт вимагає авторизації, в іншому

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижмотря О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		31

випадку можна перейти до покупки. Ще є можливість фільтрації маршрутів за місцем відправлення та прибуття, за датою відправлення та прибуття, також присутнє сортування по спаданню та зростанню за такими параметрами : за назвою автобусу в алфавітному порядку, за іменем міста прибуття\відправлення, за часом прибуття\відправлення. Фільтрація відбувається за допомогою select запитів в базу даних.

OnAvtovokzal

ГоловнаРозписання всіх автобусівКупити квитокПро нас

Напишіть код квитка...

Розклад всіх автобусів

Фільтр

Відпр: ▾Ідем в ▾Назва автобусу ▾По спаданню ▾Відправлення від: dd.mm.yyyy --:--До: dd.mm.yyyy --:--Фільтрувати

#	Автобус	Місто відправлення	Місто прибуття	Час відправлення	Час прибуття	Ціна	Замовити
1	щатцщущт	Рівне	Житомир	2024-01-28 21:04:00	2035-11-11 04:47:00	200.0	Купити
2	щатцщущт	Рівне	Житомир	2024-01-28 21:04:00	2024-01-28 22:04:00	300.0	Купити
3	щатцщущт	Рівне	Житомир	2024-01-28 22:26:00	2024-01-28 03:26:00	488.0	Купити
4	щатцщущт	Рівне	Житомир	2024-01-28 21:04:00	2024-01-28 22:04:00	200.0	Купити
5	щатцщущт	Рівне	Житомир	2024-01-28 21:28:00	2024-01-28 00:28:00	483.0	Купити

Рис.3.9. Сторінка “Розписання автобусів” коли користувач авторизований

OnAvtovokzal

ГоловнаРозписання всіх автобусівКупити квитокПро нас

Напишіть код квитка...

Ввійти

Зареєструватися

Розклад всіх автобусів

Фільтр

Відпр: ▾Ідем в ▾Назва автобусу ▾По спаданню ▾Відправлення від: dd.mm.yyyy --:--До: dd.mm.yyyy --:--Фільтрувати

#	Автобус	Місто відправлення	Місто прибуття	Час відправлення	Час прибуття	Ціна	Замовити
1	щатцщущт	Рівне	Житомир	2024-01-28 21:04:00	2035-11-11 04:47:00	200.0	Ввійти
2	щатцщущт	Рівне	Житомир	2024-01-28 21:04:00	2024-01-28 22:04:00	300.0	Ввійти
3	щатцщущт	Рівне	Житомир	2024-01-28 22:26:00	2024-01-28 03:26:00	488.0	Ввійти
4	щатцщущт	Рівне	Житомир	2024-01-28 21:04:00	2024-01-28 22:04:00	200.0	Ввійти
5	щатцщущт	Рівне	Житомир	2024-01-28 21:28:00	2024-01-28 00:28:00	483.0	Ввійти

Рис.3.10. Сторінка “Розписання автобусів” коли користувач не авторизований

Переглядаємо сторінку “Купити квиток”. Коли користувач авторизований

йому надається можливість вибрати шлях і час поїздки, в іншому випадку йому потрібно ще ввести свої контактні дані. Тут користувач може переглянути маршрут і приблизний час подорожі.

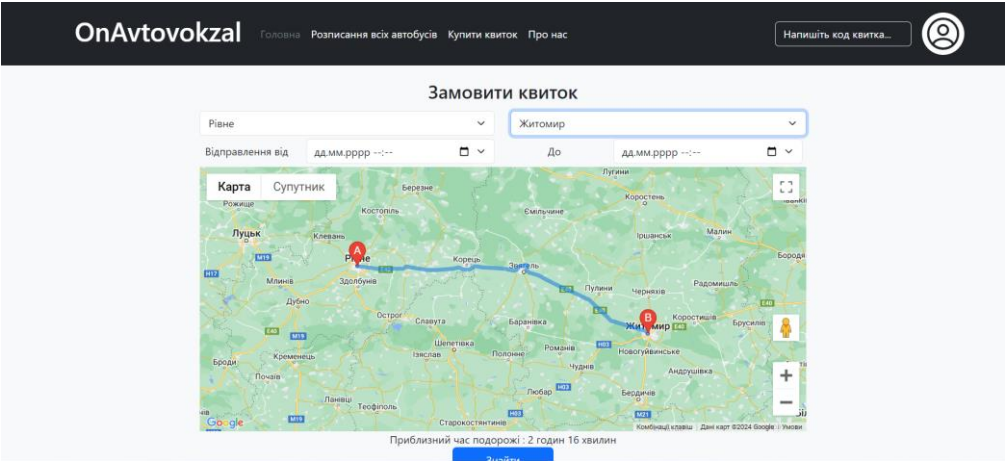


Рис.3.11. Сторінка “Купити квиток” коли користувач авторизований

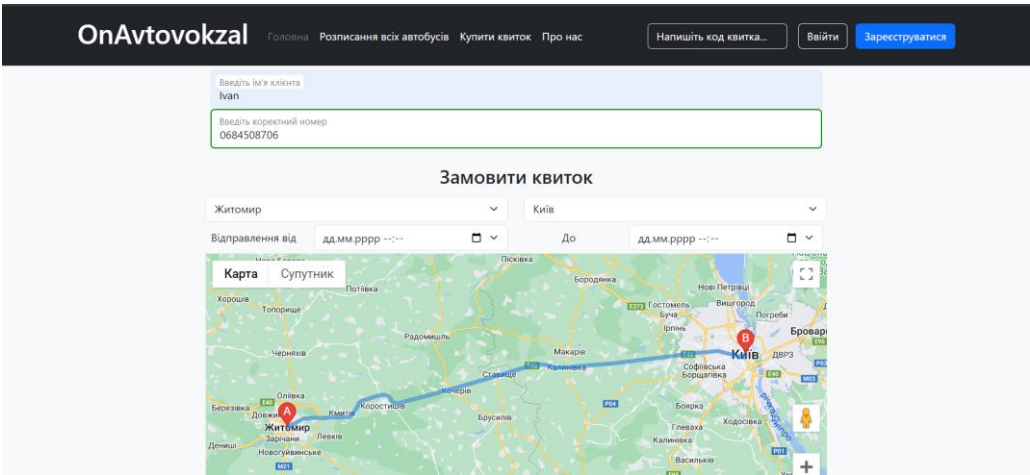


Рис.3.12. Сторінка “Купити квиток” коли користувач не авторизований

Після вибору шляху та часу робиться get запит з аргументами на сервер, сервер в свою чергу звертається до БД та за допомогою select запитів шукає потрібну інформацію та повертає всі маршрути з відповідними полями.

OnAvtovokzal						
Головна Розписання всіх автобусів Купити квиток Про нас						
Напишіть код квитка...						
#	Автобус	Місто відправлення	Місто прибуття	Час відправлення	Час прибуття	Ціна
1	Київ-Рівне	Київ	Рівне	2024-03-29 10:30:00	2024-03-29 15:30:00	400.0
Купити						
Home About						
© 2023-2024 OnAvtovokzal, Inc						

Рис.3.13. Сторінка після вибору напрямку та часу

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижморя О.В.				33
Змн.	Арк.	№ докум.	Підпис	Дата		

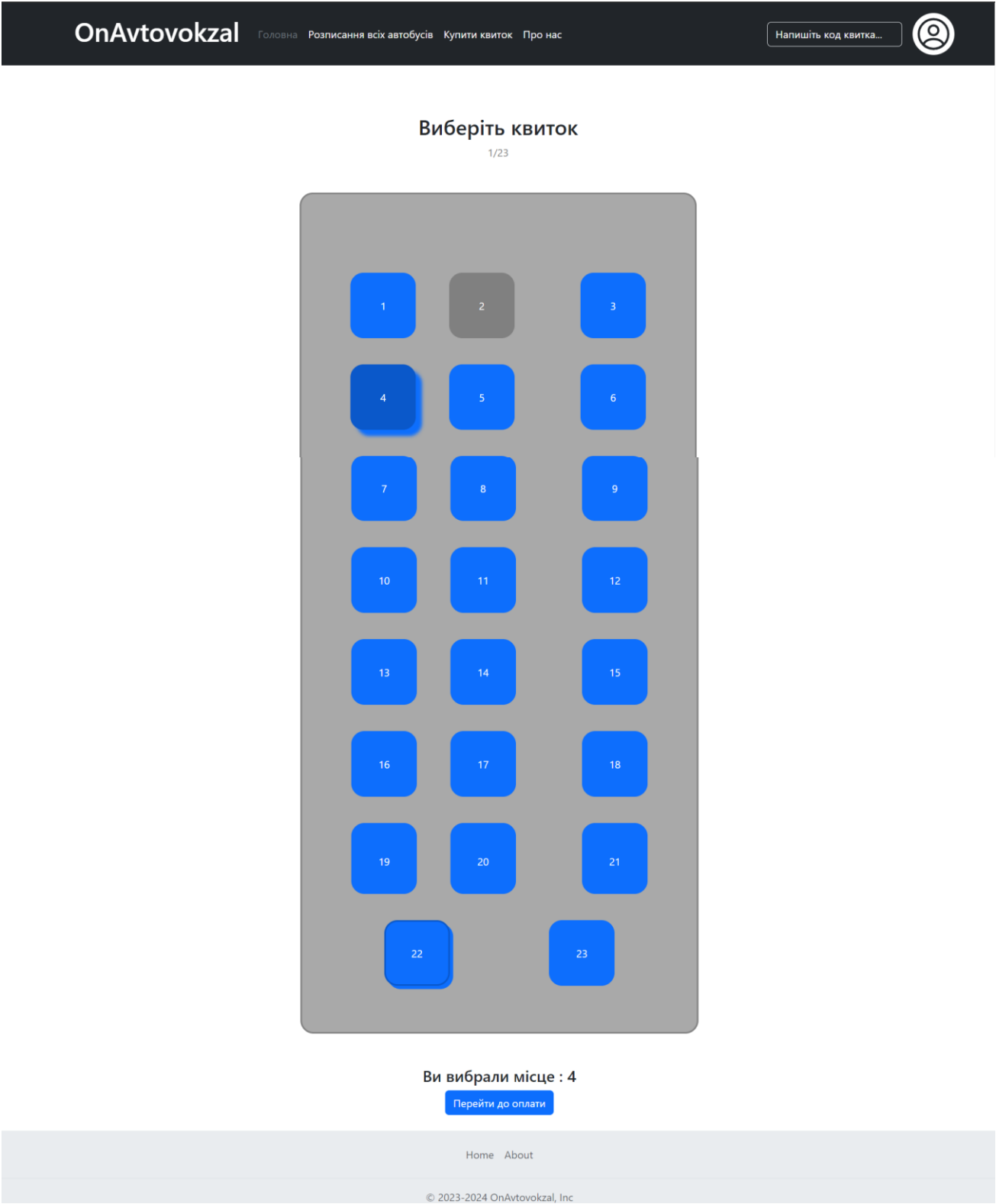


Рис.3.14 Сторінка вибору місця посадки

На рисунку 24 показано дію після вибору маршруту, ми вибираємо своє сидіння на якому бажаємо провести подорож, зверху ми можемо побачити

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижмотря О.В.				34
Змн.	Арк.	№ докум.	Підпис	Дата		

кількість вільних та зайнятих місць. Сірим колором показані зайняті місця , синім показані вільні , темно-синім ваше вибране місце. В наступному, рисунок 25 ,вікні людина перевіряє інформацію та підтверджує оплату.

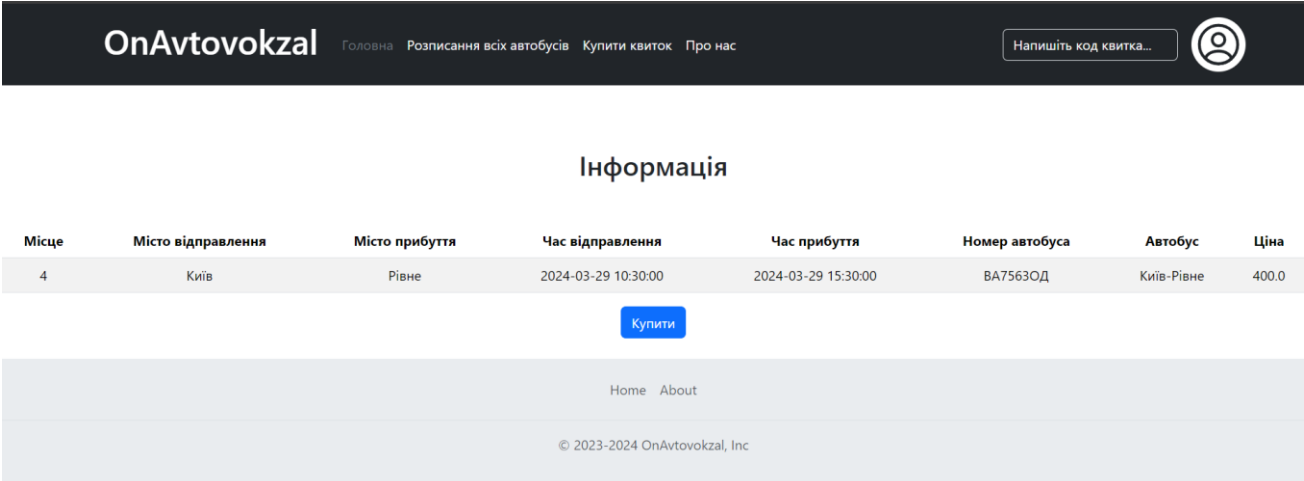


Рис.3.15. Підтвердження інформації

Після підтвердження проходить оплата і надається квиток із своїм унікальним кодом, та вся інформація про поїздки.

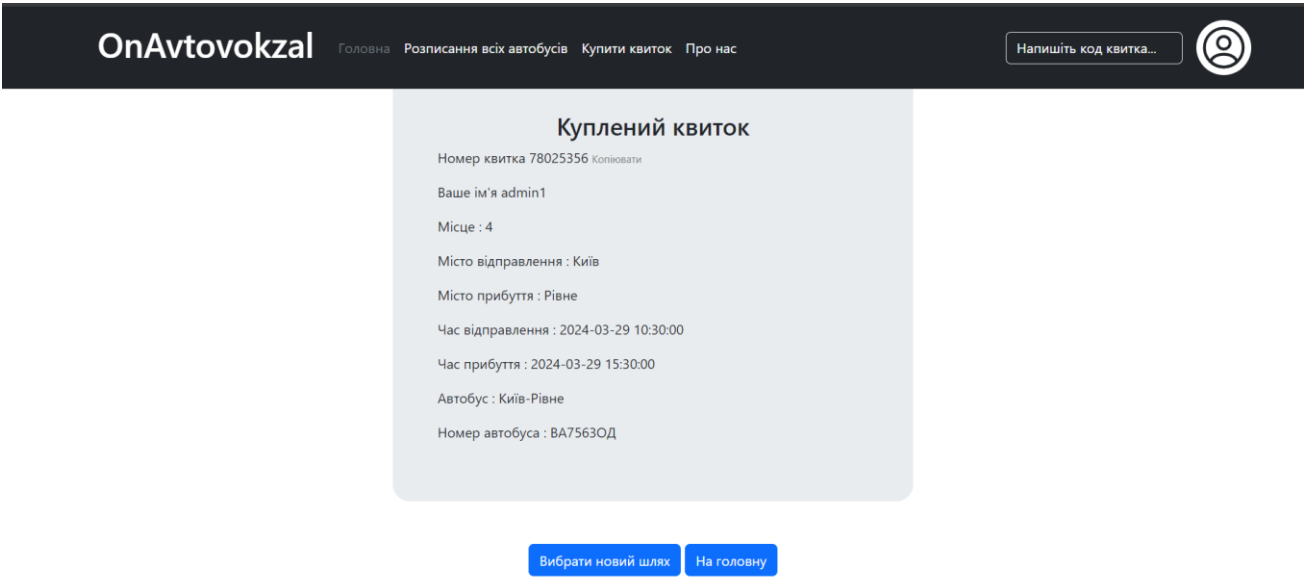


Рис.3.16. Отримання квитка та збереження його в базі даних

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижмотря О.В.				35
Змн.	Арк.	№ докум.	Підпис	Дата		

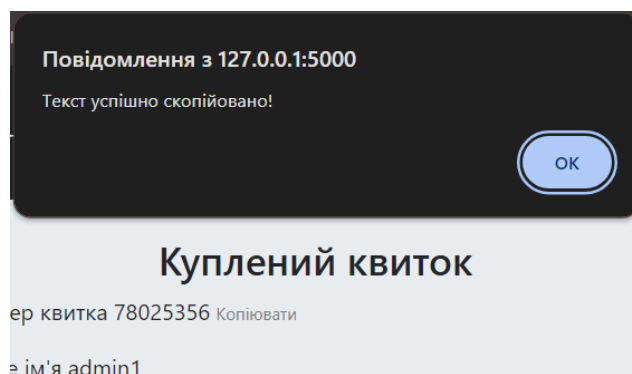


Рис.3.17.Копіювання коду квитка

На рисунку 28 показано особистий кабінет користувача тут ми можемо побачити телефон, email та ПІБ. Також є кнопка вийти яка видаляє користувача з сесії та кнопка мої квитки.

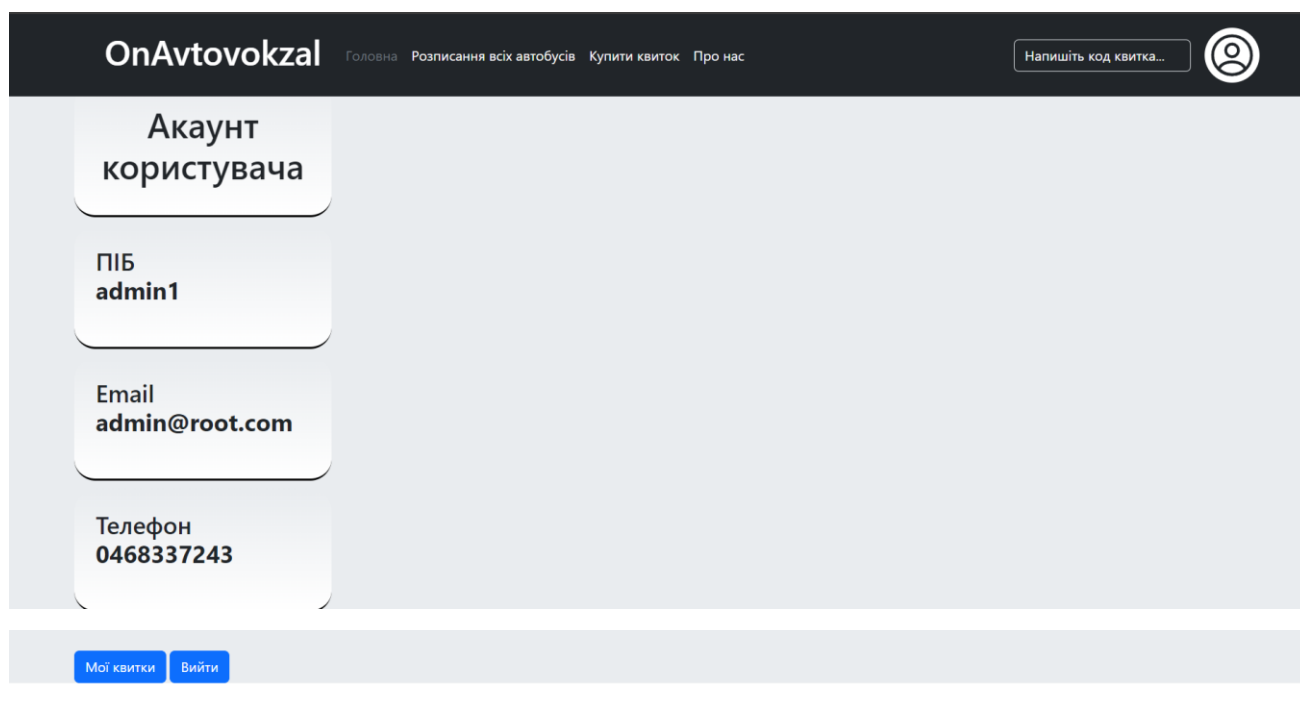


Рис.3.18. Перегляд інформації акаунту

Тепер ми можемо знайти цей квиток як за його кодом так і за допомогою форми для пошуку квитків за кодом.

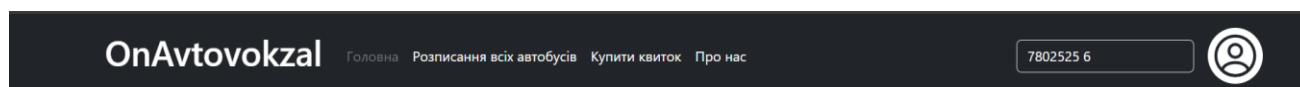


Рис.3.19. Пошук за номером квитка

Тільки ввівши всі цифри(без пробілів) квитка він розпочинає швидкий пошук та повертає саме цей квиток який нам потрібен.

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижмотря О.В.				36
Змн.	Арк.	№ докум.	Підпис	Дата		

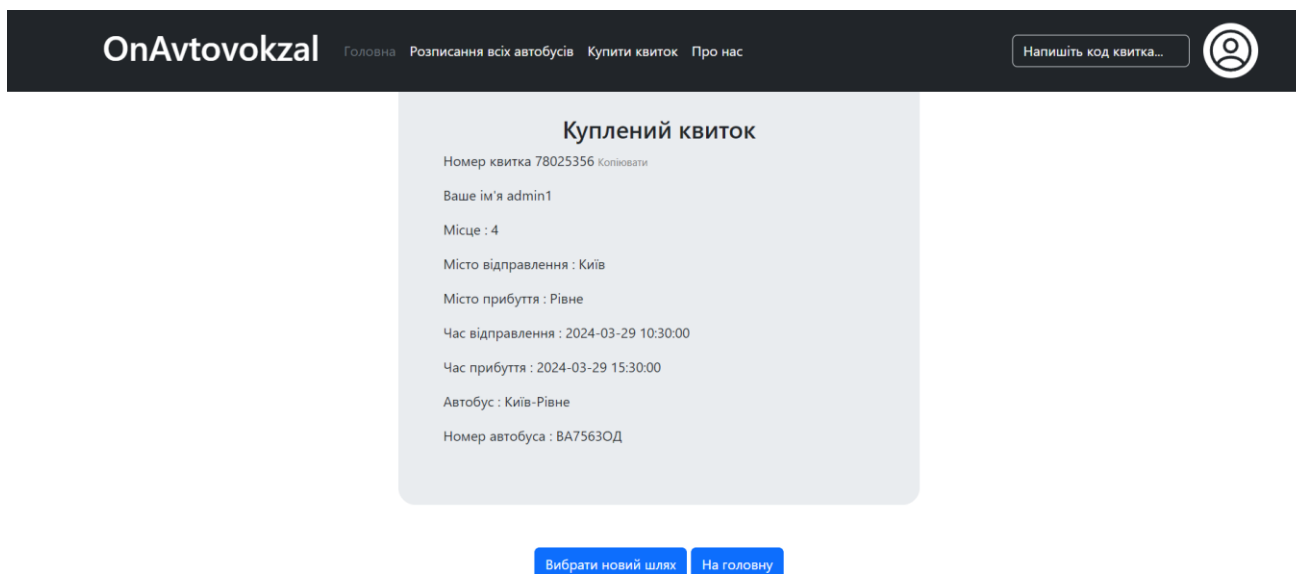


Рис.3.20. Результат пошуку

Зайшовши у власний кабінет та вибрати “Мої квитки”, та матимемо можливість переглянути квитки. Дія білетів вважається завершеною, коли подорож починається. Після клікання на кнопку "дивитись далі", користувач отримує можливість переглянути докладнішу інформацію про маршрут. Інформація про квитки доступна завдяки запитам за їхніми ідентифікаторами та ідентифікаторами відповідних даних.

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижмотря О.В.				37
Змн.	Арк.	№ докум.	Підпис	Дата		

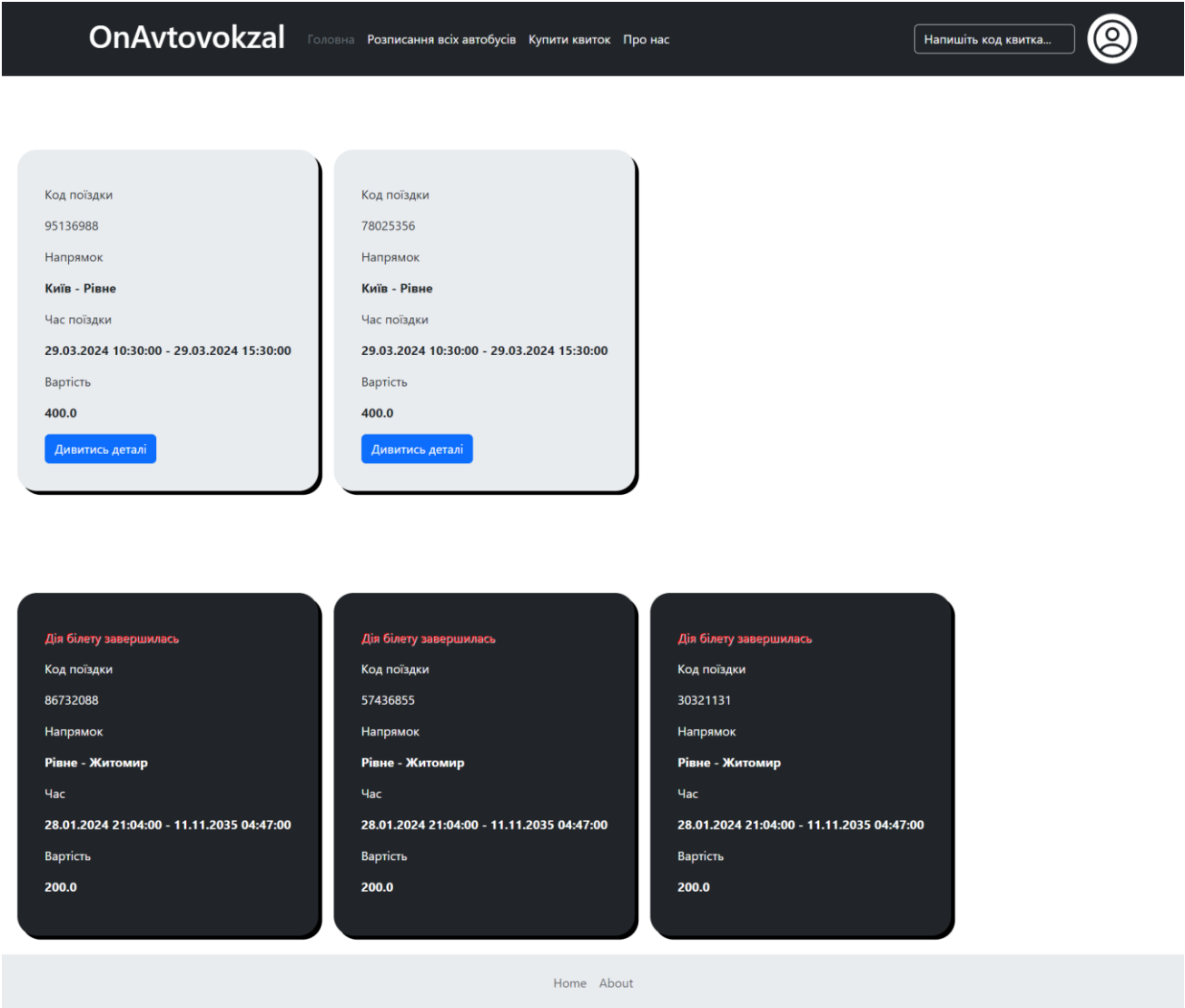


Рис.3.21. Сторінка перегляду квитків

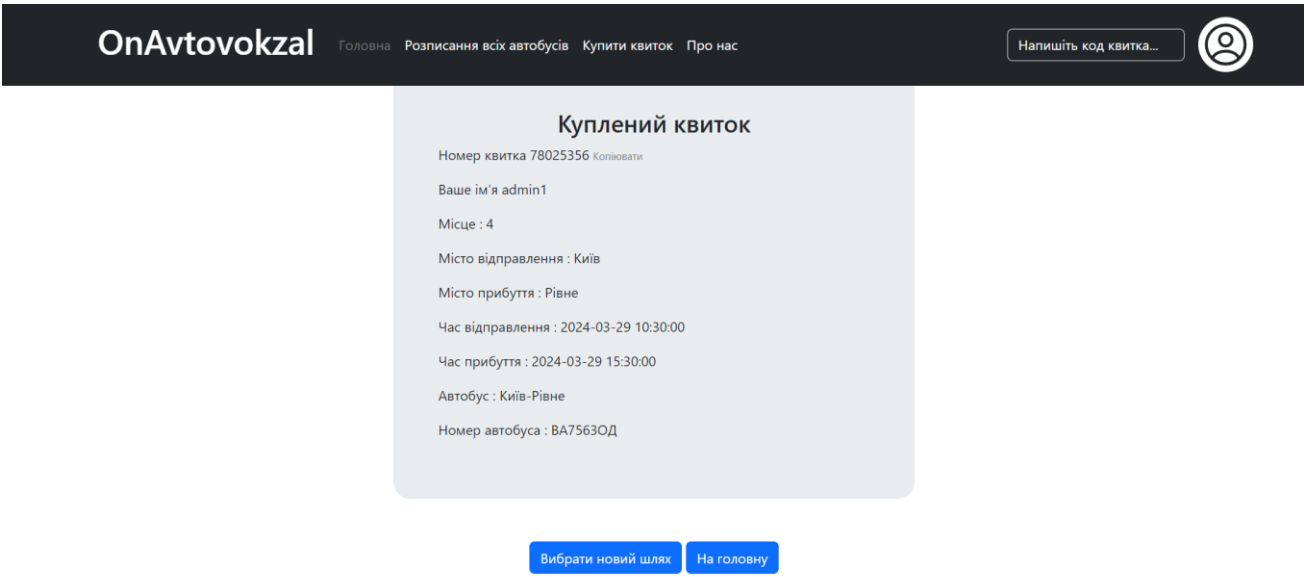


Рис.3.22. Результат перегляду

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижмотря О.В.				38
Змн.	Арк.	№ докум.	Підпис	Дата		

На сторінці “Про нас” можна знайти інформацію про компанію та її переваги.

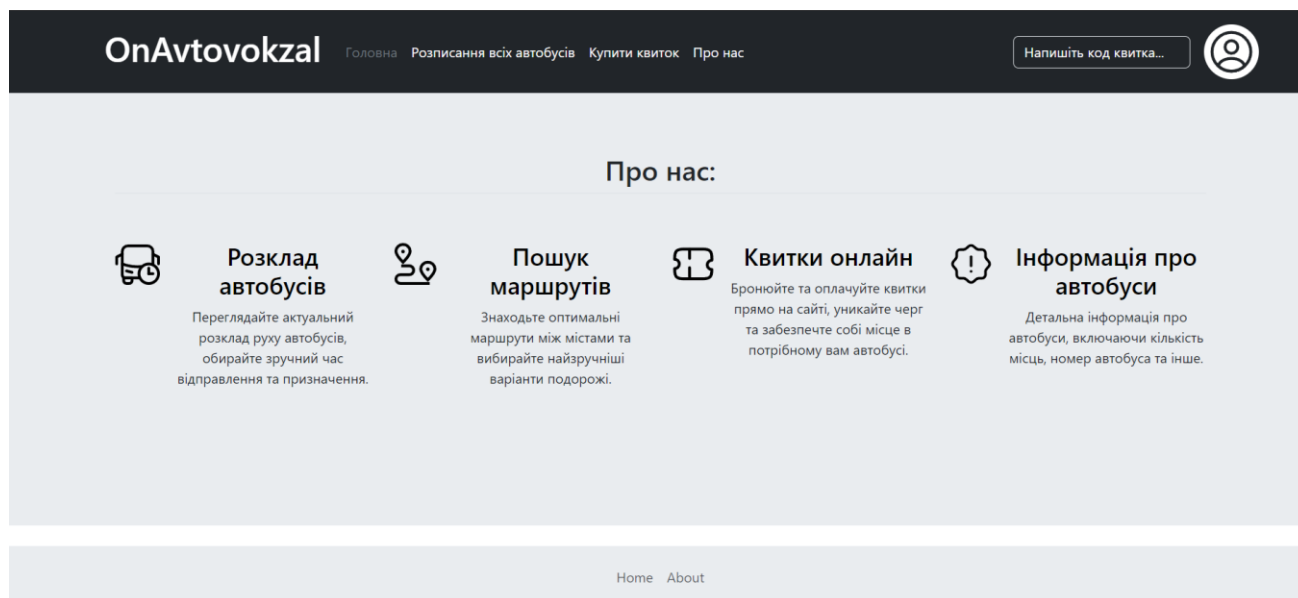


Рис.3.23. Сторінка “Про нас”

Адмінпанель:

Доступ до панелі адміністратора є тільки в адмінів, кожен користувач має свою роль, на данному етапі є тільки дві ролі такі як user та admin.

Щоб прейти в адмінпанель нам потрібно, авторизуватись з сторінки адміністратора та в полі пошуку прописати <http://127.0.0.1:5000/adminpanel>

На головній сторінці ми можемо побачити кнопки для додавання інформації, оновлення та видалення, таким чином ця сторінка реалізує посилання в совокупності на всі процеси CRUD – Create, Read, Update, Delete. Завдяки цій панелі ми можемо керувати інформацією комфортно та безпосередньо з сайту без застосування СУБД.

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижмотря О.В.				39
Змн.	Арк.	№ докум.	Підпис	Дата		

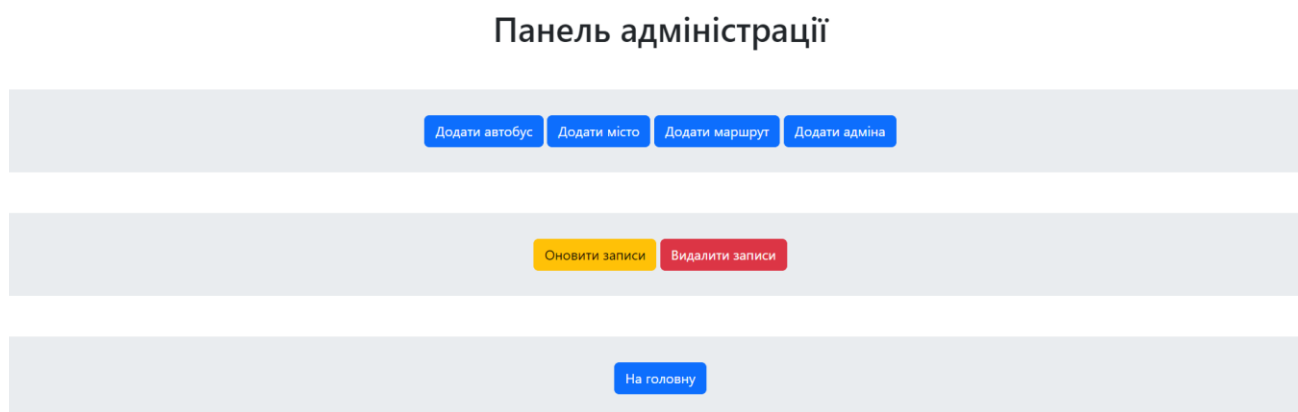


Рис.3.24. Головна сторінка адмін панелі

Операції Create:

На сторінці “Додати автобус” ми можемо помітити поля для надання назви, кількості мість та номеру нового автобуса. У випадку помилки показується аналогове вікно з текстом “Автобус не додано”.

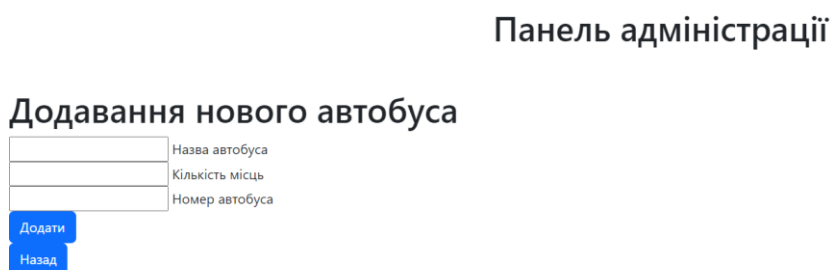


Рис.3.25. Сторінка яка реалізує додавання нового автобусу

На сторінці “Додати місто” ми можемо помітити поля для надання назви та країни. У випадку помилки показується аналогове вікно з текстом “Місто не додано”.

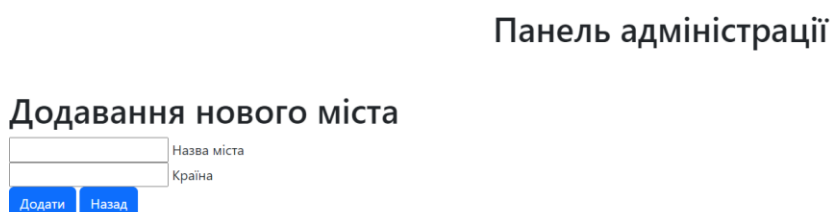


Рис.3.26. Сторінка яка реалізує додавання нового міста

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижмотря О.В.				40
Змн.	Арк.	№ докум.	Підпис	Дата		

На сторінці “Додати маршрут” ми можемо помітити поля для надання ід автобуса, міста відправлення та прибуття, плату за проїзд, час відправлення та прибуття і кількість вже зайнятих місць. У випадку помилки показується аналогове вікно з текстом “Маршрут не додано”. На цій сторінці є ще список міст та автобусів з їхніми ід для легкого додавання їх ідентифікаторів в новий запис.

Панель адміністрації

Додавання нової поїздки

☐ Час відправлення

☐ Час прибуття

#	Назва	Місця	Місто відправлення	Місто прибуття
1	щатцщущт	26	ФТ2345ОО	ФТ2345ОО
3	Новий автобус	17	ПР8976РП	ПР8976РП
4	Рівне-Житомир	30	АТ3842РО	АТ3842РО
5	Житомир-Київ	37	АП8765ІВ	АП8765ІВ
6	Київ-Рівне	23	ПР3648ІО	ПР3648ІО
7	Київ-Житомир	45	ПР8909ПР	ПР8909ПР
1007	Житомир-Київ	30	АА6758РП	АА6758РП
1008	Житомир-Київ	30	АА6758РП	АА6758РП
1009	Житомир-Львів	30	ВА6746ІВ	ВА6746ІВ
1010	Київ-Рівне	20	ВА7563ОД	ВА7563ОД

#	ІД	Назва	Країна
4	Рівне	Україна	
5	Житомир	Україна	
6	Київ	Україна	
8	Запоріжжя	Україна	
9	Кіровоград	Україна	
10	Ужгород	Україна	
13	Луцьк	Україна	
14	Чернігів	Україна	
15	Львів	Україна	

Рис.3.27. Сторінка яка реалізує додавання нового маршруту

На сторінці “Додати адміна” ми можемо помітити поля для надання імені нового адміністратора, email, телефону та паролю . У випадку помилки показується аналогове вікно з текстом “Адміністратора не додано”.

Панель адміністрації

Додавання нового адміністратора

<input type="text"/>	Ім'я Адміністратора
<input type="text"/>	Email
<input type="text"/>	Phone
<input type="password"/>	password
<input type="button" value="Додати"/>	
<input type="button" value="Назад"/>	

Рис.3.28. Сторінка яка реалізує додавання нового адміністратора сайту

Операції Update :

Після кліку по кнопці “Оновити записи” нас посилає на аналогову сторінку з вибором що ми хочемо оновити.

Панель адміністрації

Виберіть що хочете оновити

<input type="text" value="Автобус"/>	<input type="button" value="Оновити"/> <input type="button" value="Назад"/>
<input type="text" value="Подорож"/>	
<input type="text" value="Місто"/>	
<input type="text" value="Квиток"/>	

Рис.3.29. Сторінка для вибору оновлення певного об'єкту

Панель адміністрації

Що оновлюємо ?

<input type="text" value="id_city"/>
<input type="text" value="name"/>
<input type="text" value="country"/>
<input type="button" value="Оновити це"/>
<input type="button" value="Назад"/>

Рис.3.30. Сторінка для вибору оновлення певного поля об'єкту

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижмотря О.В.				42
Змн.	Арк.	№ докум.	Підпис	Дата		

Панель адміністрації

Оновлення запису

Значення поля Введіть id запису

Рис.3.31. Інтерфейс для оновлення певного поля об'єкту якого ми вибрали раніше

Після вводу даних надсилається запит оновлення в БД таблиці, поля, відповідного id . Коли виникає проблема сайт відображає повідомлення з повідомленням “Не оновлено”.

Операції Delete :

Панель адміністрації

Виберіть що хочете видалити

Автобус	<input type="button" value="Видалити"/> <input type="button" value="Назад"/>
Подорож	
Місто	
Квиток	
Клієнт	

Рис.3.32. Вибір таблиці об'єкта для видалення

Панель адміністрації

За яким полем будемо видаляти? ?

id_bus
name
seats
bus_number
<input type="button" value="Видалити за цим"/>
<input type="button" value="Назад"/>

Рис.3.33. Вибір за яким полем ми будемо видаляти

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижмотря О.В.				43
Змн.	Арк.	№ докум.	Підпис	Дата		

Панель адміністрації

Видалення запису

Значення поля

Рис.3.34. Вибір за яким значенням поля ми будемо видаляти запис Операції Read :

Операції читання вже реалізовані як на деяких сторінках адмін панелі так і в користувацькому інтерфейсі.

3.2. Реалізації операцій обробки даних в БД

Функція перевірки користувача:

```
def CreateUser(password, item, value):  
    if db_context.get_user(password, item, value) == "Пароль не співпадає":  
        return "Пароль не співпадає"  
    elif db_context.get_user(password, item, value):  
        user = db_context.get_user(password, item, value)  
        AddToDict(user)  
        return True  
    else:  
        return False
```

Ця функція призначена для перевірки користувача в базі даних за допомогою пароля та іншої інформації (наприклад, електронної пошти або номера телефону). Ось її опис:

- Перевіряється, чи співпадає пароль користувача з паролем, що введений.
- Якщо пароль не співпадає, повертається відповідне повідомлення.
- Якщо користувач існує в базі даних і пароль співпадає, користувача додається до сесії або контексту додатка, і повертається True.
- Якщо користувача не знайдено в базі даних, повертається False.

Функція отримання користувача з бази даних:

```
def get_user(self, password, item, value):  
    query = f"SELECT * FROM clients WHERE {str(item)} = ?"  
    self.cursor.execute(query, (str(value),))  
    row = self.cursor.fetchone()  
    if row:  
        user = Clients()  
        attributes = list(user.__dict__.keys())  
        for attribute, row_value in zip(attributes, row):  
            setattr(user, attribute, row_value if row_value is not None else "")
```

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижморя О.В.				44
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        print(str(user.id_client) + " " + user.name + " " + user.phone + " " +
user.password + " " + user.email)
        if user.password == password:
            print("User authorized")
            return user
        else:
            print("Incorrect password")
            return "Пароль не співпадає"
print("None account")
return None

```

Ця функція виконує запит до бази даних для отримання користувача з вказаним паролем та іншим атрибутом (електронна пошта або номер телефону).

Ось її опис:

- Формується SQL-запит з використанням формат строки f-string. Запит виглядає як `SELECT * FROM clients WHERE {str(item)} = ?`, де `{str(item)}` замінюється на ім'я поля (електронна пошта або телефон), передане як параметр `item`.
- Виконується SQL-запит з параметром `value`, який визначається у вигляді кортежу (`str(value),`).
- Отриманий результат перевіряється за допомогою методу `fetchone()`. Якщо рядок існує (тобто користувач з такою електронною поштою або номером телефону існує у базі даних), то відбувається наступне:
 - Створюється об'єкт класу `Clients`.
 - Дані з рядка записуються у відповідні атрибути об'єкта.
 - Порівнюється пароль, який був переданий функції, з паролем користувача. Якщо вони співпадають, функція повертає об'єкт користувача `user`.
 - Якщо пароль не співпадає, функція повертає рядок "Пароль не співпадає".
 - Якщо рядок не знайдено у базі даних, функція повертає `None`.

Функція додавання об'єкта в базу даних:

```

def add_item(self, obj):
    db_context = DbContext()
    db_context.select_last_index(obj)

    array_items = list(obj.__dict__.keys())
    non_id_items = [item for item in array_items if not item.startswith("id")]

```

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижмотря О.В.				45
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        values = ', '.join(
            [f'"{obj.__dict__[item]}"' if obj.__dict__[item] is not None else "NULL"
            for item in non_id_items])

        query = f"INSERT INTO {obj.__class__.__name__} ({', '.join(non_id_items)})
VALUES ({values})"
        print(query)

    try:
        self.cursor.execute(query)
        self.conn.commit()

        print(f"Item added successfully to {obj.__class__.__name__} table.")
        return True
    except Exception as e:
        print(f"Error adding item to {obj.__class__.__name__} table: {str(e)}")
        return False

```

Ця функція призначена для додавання об'єкта в базу даних, вона додає всі об'єкти які за своєю структурою схожі з таблицями бд. Ось опис цієї функції:

- Викликається метод `select_last_index`, який встановлює останній індекс для нового об'єкта, якщо це потрібно.
- Створюється список `non_id_items`, який містить атрибути об'єкта, за винятком ідентифікатора.
- Формується рядок `values`, який представляє значення для вставки в SQL-запит. Для кожного атрибута об'єкта перевіряється, чи він не є `None`, і якщо ні, то він включається у рядок значень.
- Створюється SQL-запит для вставки нового рядка в таблицю бази даних.
- Виконується SQL-запит за допомогою курсора бази даних.
- Зафіксовується зміна у базі даних за допомогою команди `commit`.
- Виводиться повідомлення про успішне додавання об'єкта до таблиці.
- Якщо сталася помилка під час додавання об'єкта до бази даних, виводиться повідомлення про помилку, і функція повертає `False`.

Взагалом ця функція додає всі можливі об'єкти які наявні в БД, але розглянемо реєстрацію нового користувача

Додавання нового користувача в базу даних:

```

if db_context.add_item(user):
    AddToDict(user)

```

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижмотря О.В.				46
Змн.	Арк.	№ докум.	Підпис	Дата		

Ці рядки коду викликають функцію `add_item` з екземпляром `DbContext` для додавання нового користувача `user` до бази даних. Якщо додавання пройшло успішно (функція `add_item` повернула `True`), то користувач також додається до сесії за допомогою функції `AddToDict`.

Функція вибірки одного запису по `id`:

```
def get_item_by_id(self, other_class_instance, item_id):
    array_items = list(other_class_instance.__dict__.keys())
    table = other_class_instance.__class__.__name__

    for item in array_items:
        if item.startswith("id"):
            id_class_name = item
            break

    query = f"SELECT {'', ' '.join(array_items)} FROM {table} WHERE {id_class_name} = ?"

    self.cursor.execute(query, (item_id,))

    row = self.cursor.fetchone()
    instance = None
    if row:
        instance = other_class_instance.__class__()
        for idx, item in enumerate(array_items):
            value = row[idx]
            setattr(instance, item, value if value is not None else None)

    return instance
```

Ця функція `get_item_by_id` призначена для отримання об'єкта з бази даних за його унікальним ідентифікатором (ID). Ось її пояснення:

- Отримання назви таблиці та списку атрибутів: Функція отримує назву таблиці та список атрибутів об'єкта, для якого вона шукає об'єкт, через параметр `other_class_instance`.
- Визначення поля ID: Потім визначається поле, яке є ідентифікатором, тобто має назву, що починається з "id".
- Створення SQL-запиту: Формується SQL-запит для отримання об'єкта з бази даних за його ID.
- Виконання запиту та створення об'єкта: Виконується запит до бази даних, і якщо об'єкт знайдено, то на його основі створюється новий об'єкт класу

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижморя О.В.				47
Змн.	Арк.	№ докум.	Підпис	Дата		

`other_class_instance.__class__`, а дані з бази даних присвоюються відповідним атрибутам цього об'єкта.

- Повернення об'єкта: Повертається знайдений об'єкт з бази даних або None, якщо об'єкт не знайдено.

Ця функція дуже корисна для отримання даних з бази даних за їхнім унікальним ідентифікатором, що дозволяє ефективно працювати з об'єктами в системі. Вона забезпечує можливість отримувати об'єкти з бази даних для подальшої обробки чи відображення.

Ця функція також використовується напротязі всього життя проекту наприклад :

```
journey = Timetable()  
journey = db_context.get_item_by_id(journey, id_journey)
```

Ця строка коду бере з бази даних один запис маршруту і перетворює його в об'єкт `Timetable`.

Функція вибору останнього індексу з певної таблиці:

```
def select_last_index(self,obj):  
    array_items = list(obj.__dict__.keys())  
    for item in array_items:  
        if item.startswith("id"):  
            id_class_name = item  
            break  
  
    query = f"SELECT MAX({str(id_class_name)}) FROM {obj.__class__.__name__}"  
  
    try:  
        self.cursor.execute(query)  
        result = self.cursor.fetchone()[0]  
        return result if result is not None else 0  
    except Exception as e:  
        print(f"Error getting last index from {obj.__class__.__name__}:  
{str(e)}")  
        return 0
```

Ця функція `select_last_index` призначена для отримання останнього індексу (або ID) з таблиці бази даних. Ось її пояснення:

- Отримання списку атрибутів об'єкта: Функція отримує список атрибутів об'єкта через його метод `__dict__`.

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижморя О.В.				48
Змн.	Арк.	№ докум.	Підпис	Дата		

- Визначення поля ID: Потім визначається поле, яке є ідентифікатором, тобто має назву, що починається з "id".
- Створення SQL-запиту: Формується SQL-запит для отримання максимального значення ID з таблиці.
- Виконання запиту та отримання результату: Виконується запит до бази даних, і якщо результат отримано успішно, то повертається максимальне значення ID. Якщо результату немає або сталася помилка, повертається 0.
- Повернення результату: Повертається максимальне значення ID або 0 як значення за замовчуванням.

Ця функція дозволяє отримати останній індекс з таблиці, що може бути корисно для створення нових записів в базі даних з унікальним ідентифікатором. Вона забезпечує можливість отримувати доступ до останнього доступного індексу для подальших операцій з даними.

Отримання елемента шукаючи за іншим значенням відмінним від id:

```
def get_item_by_other_value(self, other_class_instance, item, value):
    array_items = list(other_class_instance.__dict__.keys())
    table = other_class_instance.__class__.__name__

    query = f"SELECT * FROM {table} WHERE {item} = ?"
    self.cursor.execute(query, (value,))

    row = self.cursor.fetchone()

    if row:
        new_instance = type(other_class_instance)()
        for idx, attribute in enumerate(array_items):
            setattr(new_instance, attribute, row[idx] if row[idx] is not None else
        "")

    return new_instance

    return None
```

Ця функція `get_item_by_other_value` призначена для отримання об'єкта з бази даних за значенням певного поля, яке не є ідентифікатором. Ось її пояснення:

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижморя О.В.				49
Змн.	Арк.	№ докум.	Підпис	Дата		

- Отримання назви таблиці та списку атрибутів: Функція отримує назву таблиці та список атрибутів об'єкта, для якого вона шукає об'єкт, через параметр `other_class_instance`.
- Створення SQL-запиту: Формується SQL-запит для отримання об'єкта з бази даних за значенням певного поля.
- Виконання запиту та створення об'єкта: Виконується запит до бази даних, і якщо об'єкт знайдено, то на його основі створюється новий об'єкт того ж класу, що і вихідний, а дані з бази даних присвоюються відповідним атрибутам цього об'єкта.
- Повернення об'єкта: Повертається знайдений об'єкт з бази даних або `None`, якщо об'єкт не знайдено.

Ця функція дозволяє отримувати об'єкти з бази даних за значенням певного поля, що може бути корисно для пошуку об'єктів за різними критеріями. Вона забезпечує можливість отримувати доступ до об'єктів за їхніми властивостями для подальшого використання в програмі.

Отримання списку елементів шукаючи за значенням відмінним від `id`:

```
def get_items_by_other_value(self, other_class_instance, item, value):
    try:
        array_items = list(other_class_instance.__dict__.keys())
        table = other_class_instance.__class__.__name__

        query = f"SELECT * FROM {table} WHERE {item} = ?"
        self.cursor.execute(query, (value,))

        rows = self.cursor.fetchall()

        result = []
        if rows:
            for row in rows:
                instance = other_class_instance.__class__()
                for idx, item_name in enumerate(array_items):
                    item_value = row[idx]
                    print(item_value)
                    setattr(instance, item_name, item_value if item_value is not
None else "")
                result.append(instance)
            print("Items retrieved successfully")
```

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижморя О.В.				50
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        return result
    else:
        return None
except Exception as e:
    print(f"Error executing SQL query: {e}")
    return None

```

Ця функція `get_items_by_other_value` призначена для отримання списку об'єктів з бази даних за значенням певного поля, яке не є ідентифікатором. Ось її пояснення:

- Отримання назви таблиці та списку атрибутів: Функція отримує назву таблиці та список атрибутів об'єкта, для якого вона шукає об'єкти, через параметр `other_class_instance`.
- Створення SQL-запиту та виконання: Формується SQL-запит для отримання об'єктів з бази даних за значенням певного поля. Запит виконується із значенням параметра, яке передається через `value`.
- Обробка результату запиту: Після виконання запиту отримані рядки з бази даних обробляються. Якщо рядки знайдено, для кожного рядка створюється новий об'єкт того ж класу, що і вихідний, а дані з бази даних присвоюються відповідним атрибутам цього об'єкта. Створені об'єкти додаються до списку `result`.
- Повернення списку об'єктів: Повертається список знайдених об'єктів з бази даних або `None`, якщо об'єкти не знайдено.

Ця функція дозволяє отримувати списки об'єктів з бази даних за значенням певного поля, що може бути корисно для виконання різноманітних завдань пошуку та фільтрації даних. Вона забезпечує можливість отримувати доступ до групи об'єктів за їхніми властивостями для подальшого використання в програмі.

Функція оновлення запису :

```

def update_item(self, id, item, value, obj):
    nametable = obj.__class__.__name__

    if nametable=="Buses":
        name_id='id_bus'
    elif nametable=="Clients":
        name_id='id_client'
    elif nametable=="Timetable":

```

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижмотря О.В.				51
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        name_id='id_journey'
    elif nametable == "Cities":
        name_id = 'id_city'
    elif nametable == "Non_authorized_users":
        name_id = 'id_user'
    elif nametable == "Non_authorized_users":
        name_id = 'id_user'
    elif nametable == "Orders":
        name_id = 'id_order'
    elif nametable == "Tickets":
        name_id = 'id_ticket'
    else:
        print("Не підтримуваний клас")
        print("err on DBContext at 158 line")
        return False

    query = f"UPDATE {nametable} SET {item}=? WHERE {name_id}=?"
    values = (value, id)

    try:
        with self.conn:
            with self.cursor.execute(query, values):
                print(f"Update successful for {item} with value {value} for record
with {name_id}={id} in table {nametable}.")
                return True
    except Exception as e:
        print(f"Error updating record: {e}")
        return False

```

Ця функція update_item призначена для оновлення запису в базі даних. Ось її пояснення:

- Визначення назви поля для ідентифікатора: Перевіряється назва класу obj, для якого виконується оновлення, і відповідно до цього визначається назва поля ідентифікатора.
- Створення SQL-запиту: Формується SQL-запит для оновлення запису в таблиці. Запит виконується для вказаного поля item та значення value, де умовою є ідентифікатор рядка, що дорівнює id.
- Виконання запиту: Виконується SQL-запит для оновлення запису в таблиці з використанням значень item та id.
- Повернення результату оновлення: Повертається True, якщо оновлення пройшло успішно, або False, якщо сталася помилка під час виконання запиту.

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижмотря О.В.				52
Змн.	Арк.	№ докум.	Підпис	Дата		

Ця функція дозволяє оновлювати дані у вказаному полі для запису в таблиці з використанням ідентифікатора цього запису. Вона може бути використана для зміни певних даних в базі даних, наприклад, оновлення даних про автобус, клієнта або розклад подорожі.

Функція видалення записів з БД:

```
def delete_item(self, obj, item, value):
    nametable = obj.__class__.__name__
    query = f"DELETE FROM {nametable} WHERE {item} = ?"
    values = (value,)

    try:
        with self.conn:
            with self.cursor.execute(query, values):
                print(f"Delete successful from {nametable} where {item} = {value}.")
                return True
    except Exception as e:
        print(f"Error deleting record: {e}")
        return False
```

Ця функція `delete_item` призначена для видалення записів з бази даних за певною умовою. Ось її пояснення:

- **Визначення назви таблиці:** Визначається назва таблиці, з якої будуть видалятися записи. Для цього використовується назва класу `obj`.
- **Створення SQL-запиту:** Формується SQL-запит на видалення записів з таблиці, де умовою для видалення є співпадіння значення поля `item` з вказаним значенням `value`.
- **Виконання запиту:** Виконується SQL-запит для видалення записів з таблиці, які відповідають умові.
- **Повернення результату видалення:** Повертається `True`, якщо видалення пройшло успішно, або `False`, якщо сталася помилка під час виконання запиту.

Ця функція дозволяє видаляти записи з таблиці бази даних за певною умовою, яка вказується через параметри `item` та `value`. Наприклад, вона може бути

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижмотря О.В.				53
Змн.	Арк.	№ докум.	Підпис	Дата		

використана для видалення певного клієнта або автобуса з бази даних за його ідентифікатором або іншими атрибутами.

Клас DbContext реалізує доступ до бази даних та виконує операції з нею за допомогою бібліотеки pyodbc, яка надає засоби для з'єднання з базами даних за допомогою ODBC. Ось коротка інформація про бібліотеку :

- pyodbc - це бібліотека Python для доступу до баз даних за допомогою інтерфейсу ODBC (Open Database Connectivity).
- Вона дозволяє взаємодіяти з різними типами баз даних, такими як SQL Server, MySQL, PostgreSQL тощо, використовуючи стандартний інтерфейс ODBC.

Висновок до третього розділу :

У цьому розділі було описано розробку функцій взаємодії з базою даних у веб-додатку за допомогою класу DbContext та бібліотеки pyodbc. pyodbc - це бібліотека для з'єднання з базами даних за допомогою ODBC (Open Database Connectivity) у Python. Вона дозволяє взаємодіяти з різними системами управління базами даних (СУБД), такими як SQL Server, MySQL, PostgreSQL та іншими.

Клас DbContext містить реалізацію функцій для виконання операцій з базою даних, таких як додавання, оновлення, видалення та отримання даних. Ці функції дозволяють взаємодіяти з об'єктами бази даних, такими як користувачі, автобуси, розклади, міста тощо, шляхом виконання SQL-запитів.

У результаті було розроблено повний набір функцій для взаємодії користувача з базою даних у веб-додатку. Ці функції дозволяють додавати, оновлювати, видаляти та отримувати записи з бази даних, що забезпечує повний функціонал для управління даними у додатку.

Також був розроблений комфортний користувацький інтерфейс веб-сайту для, що включає сторінки для реєстрації, авторизації, купівлі квитків, перегляду всіх маршрутів та їх фільтрування, а також перегляд збережених квитків.

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижмотря О.В.				54
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 4. АДМІНІСТРУВАННЯ БАЗ ДАНИХ

4.1. Розробка заходів захисту інформації в БД

Отже, розглянемо категорії користувачів веб-сайту автовокзалу "OnAvtovokzal": адміністратор, авторизовані користувачі, неавторизовані користувачі. Адміністратор має найбільше доступних опцій та функцій, що необхідно для підтримання роботи сайту. Вони можуть редагувати інформацію, більш детально про роль надано ролі "Адміністратор" наведено в табл. 8

Таблиця 4.1.

Матриця доступу для ролі "Адміністратор"

	Адміністратор
Створити новий маршрут	+
Редагувати інформацію про маршрути	+
Створити новий автобус	+
Редагувати інформацію про маршрути	+
Створити новий маршрут	+
Редагувати інформацію про маршрути	+
Створити нове місто	+

Редагувати інформацію про міста	+
Додати адміністратора	+
Видалити користувача	+
Видалити квиток	+
Видалити маршрут	+
Видалити автобус	+
Видалити місто	+
Купити квиток	+
Переглянути інформацію про акаунт	+
Переглянути інформацію про квитки які зв'язані з цим клієнтом	+
Переглянути розписання маршрутів	+
Купівля квитка на сторінці	+

“Розписання всіх автобусів”	
Переглянути квиток за кодом	+

Основну масу користувачів системи складають клієнти, які реєструються на сайті та мають свої особисті кабінети, тому, що вони можуть переглядати всі квитки в кабінеті, які вони купували на відміну від не авторизованих користувачів. Більш детально про роль "Авторизований користувач" наведено в табл. 9

Таблиця 4.2.

Матриця доступу для ролі "Авторизований користувач"

	Авторизований користувач
Створити новий маршрут	-
Редагувати інформацію про маршрути	-
Створити новий автобус	-
Редагувати інформацію про маршрути	-
Створити новий маршрут	-
Редагувати інформацію про маршрути	-

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижмотря О.В.				57
Змн.	Арк.	№ докум.	Підпис	Дата		

Створити нове місто	-
Редагувати інформацію про міста	-
Додати адміністратора	-
Видалити користувача	-
Видалити квиток	-
Видалити маршрут	-
Видалити автобус	-
Видалити місто	-
Купити квиток	+
Переглянути інформацію про акаунт	+
Переглянути інформацію про квитки які зв'язані з цим клієнтом	+
Переглянути розписання маршрутів	+

Купівля квитка на сторінці “Розписання всіх автобусів”	+
Переглянути квиток за кодом	+

Також розглянемо функціонал для не авторизованого користувача:

Таблиця 4.3.

Матриця доступу для ролі "Не авторизований користувач"

	Авторизований користувач
Створити новий маршрут	-
Редагувати інформацію про маршрути	-
Створити новий автобус	-
Редагувати інформацію про маршрути	-
Створити новий маршрут	-
Редагувати інформацію про маршрути	-

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижмотря О.В.				59
Змн.	Арк.	№ докум.	Підпис	Дата		

Створити нове місто	-
Редагувати інформацію про міста	-
Додати адміністратора	-
Видалити користувача	-
Видалити квиток	-
Видалити маршрут	-
Видалити автобус	-
Видалити місто	-
Купити квиток	+
Переглянути інформацію про акаунт	-
Переглянути інформацію про квитки які зв'язані з цим клієнтом	-
Переглянути розписання маршрутів	+

Купівля квитка на сторінці “Розписання всіх автобусів”	-
Переглянути квиток за кодом	+

4.2. Налаштування параметрів роботи Sql Server.

Для того, щоб підключити Sql Server в проект потрібно імпортувати та встановити бібліотеку `pyodbc` за допомогою команди `pip install`.

Бібліотека `pyodbc` - це інструмент для забезпечення зв'язку між Python та базами даних, особливо з реляційними СУБД, такими як Microsoft SQL Server. Вона дозволяє виконувати SQL-запити до бази даних, отримувати результати та взаємодіяти з ними безпосередньо з Python-коду. Ця бібліотека надає доступ до драйверів ODBC (Open Database Connectivity), які є стандартом для з'єднання з різними типами баз даних. Завдяки цьому, вона підтримує роботу з багатьма типами баз даних, включаючи Microsoft SQL Server, MySQL, PostgreSQL та інші.

За допомогою `pyodbc` можна створювати з'єднання з базою даних, виконувати SQL-запити, вставляти, оновлювати та видаляти дані, а також отримувати результати запитів для подальшого аналізу та використання в програмах на Python. Вона дозволяє розробникам легко і ефективно взаємодіяти з базами даних у своїх проектах.

Функціонал цієї бібліотеки було реалізовано в класі `DbContext`. Цей клас містить реалізацію функціоналу для роботи з базою даних, такий як виконання SQL-запитів, вставка, оновлення та видалення даних, а також отримання результатів для подальшого використання на сервері Flask. Для початку ми під'єднуємо клас до БД за допомогою строки під'єднання, та створюємо курсор який буде виконувати пошук по БД :

```
def __init__(self):
```

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижмотря О.В.				61
Змн.	Арк.	№ докум.	Підпис	Дата		

```
self.conn = pyodbc.connect("DRIVER={ODBC Driver 17 for SQL
Server};SERVER=DESKTOP-TSMTSMN;DATABASE=avtovokzal;Trusted_Connection=yes")
self.cursor = self.conn.cursor()
```

Для адміністрування БД за допомогою СУБД потрібно зайти в програму SSMS та встановлюємо server name у відповідності з назвою комп'ютера, або відповідно з даними які надав хост.

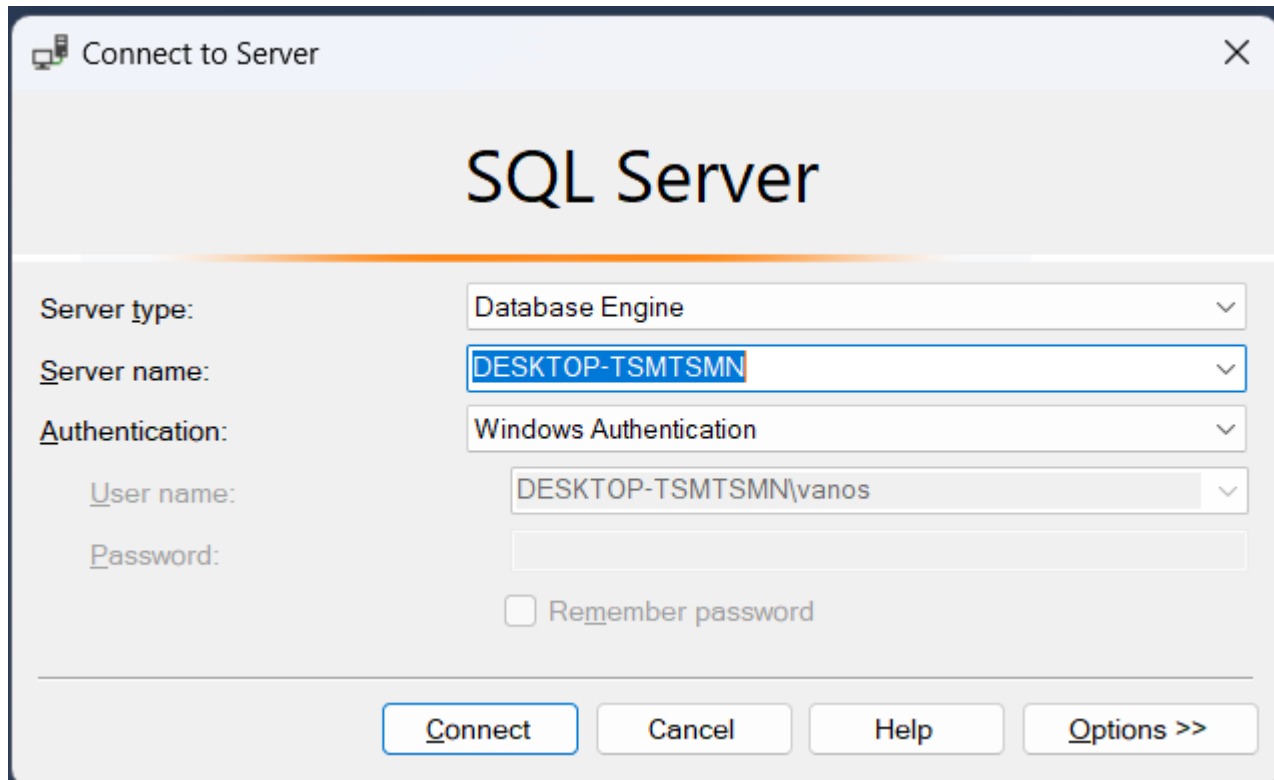


Рис.4.1. Підключення до серверу бази даних в СУБД

Висновки до четвертого розділу

У ході виконання даного розділу проведено налаштування прав доступу користувачів до бази даних. Було розглянуто різні категорії користувачів веб-сайту автовокзалу "OnAvtovokzal" та встановлено їхні права доступу до функціоналу сайту.

Адміністратор має найбільші можливості управління сайтом, включаючи створення та редагування інформації про маршрути, автобуси, міста, а також можливість додавати та видаляти користувачів та квитки. Авторизований користувач має обмежені можливості порівняно з адміністратором, але все ж може купувати квитки, переглядати інформацію про свій акаунт та квитки, а

		Середя І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижморя О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		62

також переглядати розписання маршрутів. Неавторизований користувач може тільки переглядати розписання маршрутів та купувати квитки.

Для роботи з базою даних SQL Server у проекті використовується бібліотека pyodbc, яка забезпечує зв'язок між Python та базою даних. Бібліотека надає можливість виконувати SQL-запити, вставляти, оновлювати та видаляти дані, а також отримувати результати для подальшого використання. Для адміністрування БД використовується SQL Server Management Studio (SSMS), де налаштовуються параметри роботи сервера баз даних.

У результаті проведених дій забезпечено ефективне управління та захист інформації в БД для забезпечення безпеки та функціональності веб-сайту "OnAvtovokzal".

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижмотря О.В.				63
Змн.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВКИ

У результаті виконання даного проекту було спроектовано та розроблено сайт новинного блогу.

Під час виконання курсового проекту було проаналізовано головні, можливості та особливості різних БД, що допомогло визначити основні плюси та мінуси їх та вибрати ту, за допомогою якої можна якісно та швидко виконати дану роботу.

Після того, як було вибрано та пояснено причину та логіку вибору БД було виконано проектування схеми бази. У результаті виконання цього етапу, було розроблено коректну загальну схему бази, що дозволило отримати розуміння того, що саме має відбуватись в коді та як це потрібно реалізувати.

Після проектування загальної структури бази, було розпочато проектування інтерфейсу додатку. Під час цього етапу було реалізовано підключення до БД, створені різні методи для коректної взаємодії з базою. Наприклад, було написано методи для вставок даних, для оновлен даних, для їх видалення й пошуку.

При виконання загальної структури коду, було виконано налаштування деяких додаткових сервісів для коректної роботи додатку, а саме для збереження медіа-файлів та можливості валідного входу та реєстрації користувача в додатку.

Було проведено тестування у ручному режимі, після виявлення помилок, їх було виправлено та пере-протестовано додаток, що дає можливість визнати, що додаток є стабільним. Таким чином була розроблена програма з використанням мови PHP, SQL, HTML, CSS, JavaScript використаний Bootstrap та phpMyAdmin.

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижмотря О.В.				64
Змн.	Арк.	№ докум.	Підпис	Дата		

Список використаних джерел

1. SQL Server Management Studio (SSMS) - TutorialsTeacher.com, [Електронний ресурс] - 2024, Режим доступу: <https://www.tutorialsteacher.com/sqlserver/sql-server-management-studio>
2. SQL Server Management Studio (SSMS) - learn.microsoft.com , [Електронний ресурс] - 2023, Режим доступу: <https://learn.microsoft.com/en-us/sql/ssms/sql-server-management-studio-ssms?view=sql-server-ver16s>
3. Python Tutorial - W3Schools [Електронний ресурс] -
Режим доступу до ресурсу : <https://www.w3schools.com/python/>
4. Learn Python - Free Interactive Python Tutorial - learnpython.org [електронний ресурс] - 2006. Режим доступу: <https://www.learnpython.org/>
5. Learn Python - Free Interactive Python Tutorial - learnpython.org [електронний ресурс] – 2022. Режим доступу: <https://docs.python.org/3/tutorial/index.html>
6. Welcome to Flask — Flask Documentation (3.0.x) - flask.palletsprojects.com, Режим доступу : <https://flask.palletsprojects.com/en/3.0.x/>
7. Створення SQL запитів [електронний ресурс] - 2016. Режим доступу: https://www.politerm.com/zuludoc/spatial_query_sql.htm#_sql_overview
8. Форум з питаннями та відповідями [Електронний ресурс] – Режим доступу до ресурсу: <https://stackoverflow.com/> .
9. SQL [електронний ресурс] - 20 16. Режим доступу: <https://uk.wikibooks.org/wiki/SQL>
10. AJAX Response from Asynchronous JavaScript Call [електронний ресурс] - 2017. Режим доступу: <https://stackify.com/return-ajax-response-asynchronous-javascript-call/> .

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижморя О.В.				65
Змн.	Арк.	№ докум.	Підпис	Дата		

ДОДАТКИ

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижмотря О.В.				66
Змн.	Арк.	№ докум.	Підпис	Дата		

Технічне завдання

1. Загальне положення

1.1. Найменування програмного засобу

Повне найменування програмної системи: “База даних каси автовокзалу” (надалі "програма"). Коротка назва програмної системи - “OnAvtovokzal”

1.2. Призначення розробки та область застосування

Програмна система “База даних автовокзалу” призначена для збору, обробки та відображення даних про рейси автобусів, квитки та користувачів. В програмі передбачено використання механізму пошуку рейсів додатку. Система може бути впроваджена у роботу перевезень з кількістю рейсів до 100 штук та базою користувачів до 100000 записів.

1.3. Мета

Програмна система “База даних автовокзалу” дозволить отримати інформацію про наявні перевезення, купівлю квитків на доступні рейси. Підвищити прибуток перевезень за рахунок продажу в інтернеті.

1.4. Найменування розробника та замовника.

Розробник даного продукту - студент групи ВТ-22-1 Середя Іван Романович (надалі "розробник").

Замовник програмного продукту – кафедра інженерія програмного забезпечення Житомирського державного технологічного університету в межах виконання курсової з дисципліни «Бази даних» Чижмотря Олексій Володимирович, Сугоняк Інна Іванівна (надалі замовник).

2. Підстава для розробки

2.1. Документ на підставі якого ведеться розробка

Робота ведеться на підставі навчального плану за напрямом 121 «Інженерія програмного забезпечення».

3. Вимоги до програми

3.1. Вимоги до функціональних характеристик.

		Середя І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижмотря О.В.				67
Змн.	Арк.	№ докум.	Підпис	Дата		

3.1.1. Загальні вимоги

Веб-додаток має забезпечувати:

- можливість дистанційної роботи з робочих станцій локальної та глобальної мережі підприємства;
- постійний доступ користувачів веб-додатку;
- організацію управління сайтом;
- можливість доступ до бази даних;
- постійний доступ користувачів до БД;
- оптимальне збереження даних (за обсягом та структурою);
- аутентифікацію користувачів та захист інформації від несанкціонованого доступу;
- надійне збереження даних та можливість відновлення даних у випадку непередбачуваних збоїв системи;
- можливість модернізації системи через зміну функціональних потреб користувача

3.1.2. Склад виконуваних функцій

Розробити базу даних, що використовується у веб-додатку, який підтримує виконання наступних операцій:

1. Реєстрація користувачів
2. Вхід в обліковий запис
3. Перегляд, редагування, створення, видалення користувачів, квитків, рейсів, сервісів, автобусів та компаній.
4. Редагування прав доступу кожного користувача.
5. Купівля квитків на рейси та скасування їх за потреби.
6. Резервне копіювання через веб-додаток.
7. Відображення всієї необхідної інформації про рейс.

3.1.3. Організація введення та виведення даних

Для ефективного введення та виведення даних система буде використовувати інформацію про рейси, квитки, автобуси та компанії. Вхідні та

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижмотря О.В.				68
Змн.	Арк.	№ докум.	Підпис	Дата		

вихідні дані повинні відповідати інформаційній структурі операцій, які виконуються.

Введення оперативних даних буде здійснюватися через діалогові екранні форми, що побудовані на базі візуальних компонентів. Введення даних буде здійснюватися за допомогою відповідних форм, доступних у додатку.

Це забезпечить зручний та ефективний процес введення та виведення необхідної інформації, що відповідає вимогам операцій та сприятиме оптимізації робочих процесів.

3.1.4. Часові характеристики і розмір пам'яті, необхідної для роботи програми.

Час реакції програми на взаємодію користувача (введення даних) не перевищує 0,25 секунди.

Час виконання команд меню не перевищує 1 секунди.

Відображення масивів даних за запитом не більше 3 хвилин.

Доступність бази даних - 90%, забезпечена цілодобово.

Операції з'єднання з базою даних займають не більше 1 хвилини.

Обсяг оперативної пам'яті, необхідний для роботи програми, складає не менше 1 ГБ.

Дисковий простір, необхідний для збереження програми та файлів даних, не перевищує 300 МБ для робочої станції та 20 ГБ.

Розмір інсталяційного пакету програми, який включає в себе базу даних, не перевищує 100 МБ.

3.2. Вимоги до надійності.

3.2.1. Вимоги до надійного функціонування

Програма повинна безперебійно функціонувати при стабільній роботі ПК. Доступність бази даних має бути на рівні 90% при одночасному доступі 30 користувачів. У випадку апаратних збоїв відновлення нормальної роботи програми повинно бути наступним чином:

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижморя О.В.				69
Змн.	Арк.	№ докум.	Підпис	Дата		

а) При апаратних збоях сервера - виконується перезавантаження операційної системи сервера та запуск сервера бази даних. Якщо передбачено, може бути використано резервний сервер або технології RAID для збереження даних.

б) При апаратних збоях робочої станції - виконується перезавантаження операційної системи ПК та запуск виконуваного файлу програми.

в) База даних повинна бути відновлена до найближчого несуперечного стану, для чого слід передбачити точки відновлення.

У випадку програмних збоїв:

а) Система повинна забезпечувати можливість відновлення даних та фіксацію та "відкат" транзакцій.

б) В системі має бути реалізована коректна обробка виняткових ситуацій.

3.2.2.Контроль вхідної і вихідної інформації

Для контролю коректності вхідної інформації та захисту від помилок оператора система передбачає наступні заходи:

- Перевірка відповідності даних доменам інформаційних атрибутів, щоб упевнитися у відповідності введених даних до очікуваних типів та форматів.
- Використання механізму автоматичного заповнення та вибору з переліку для пов'язаних даних, що дозволяє операторам зручно та швидко вводити дані без помилок.
- Захист від помилок оператора, таких як залипання клавіш або випадкового введення символів, шляхом використання відповідних інтерфейсних рішень.

Некоректні дії оператора супроводжуються повідомленнями про помилку та блокуванням операцій оновлення даних. При цьому система запобігає загальному блокуванню для забезпечення неперервності роботи.

Для вихідної інформації система передбачає:

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижмотря О.В.				70
Змн.	Арк.	№ докум.	Підпис	Дата		

- Відсутність блокування даних через багато користувальницький доступ, забезпечуючи при цьому необхідний рівень доступності даних для всіх користувачів.
- Постійне оновлення даних у відображених на екрані звітних формах для актуальності та точності інформації.
- Повідомлення про використання режиму монопольного доступу супервайзерами бази даних, щоб уникнути конфліктів та забезпечити безперебійну роботу системи.

3.2.3. Час відновлення після відмови.

Час відновлення після відмови, не пов'язаною з роботою програми, повинен складатися із: часу перезавантаження операційної системи; часу запуску сервера БД (підключення до сервера) запуску виконуваного файлу, часу повторного введення або зчитування даних.

3.3. Умови експлуатації і збереження

Програма призначена для використання у багатокористувацькому середовищі. Регламентовані операції проводяться відповідно до наступного графіка:

- Створення резервних копій даних: проводиться один раз на місяць з метою забезпечення безпеки даних.
- Оновлення системи ідентифікації та аутентифікації користувачів: виконується один раз на місяць, а також за необхідності з урахуванням поточних вимог безпеки. Збереження архівних копій здійснюється на визначених носіях, а резервних копій - на визначеному диску без перезапису.
- Знищення копій проводиться відповідно до регламенту з метою забезпечення конфіденційності та безпеки даних.

3.4. Вимоги до інформаційної і програмної сумісності.

3.4.1. Вимоги до інформаційних структур на вході і виході

Формат відображення даних має дозволяти виконання операцій з

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижморя О.В.				71
Змн.	Арк.	№ докум.	Підпис	Дата		

даними в браузері.

3.4.2. Вимоги до методів рішення і мов програмування

Вибір методів рішення здійснюється розробникам без узгодження з замовником. СУБД обирається у відповідності до характеристик визначених в п.3. З замовником погоджується вибір варіанту за вартісною ознакою.

3.4.3. Вимоги до системи програмних засобів.

Вимоги до програмного забезпечення сервера: ОС – серверна версія ОС Windows не нижче Windows 2000 Server, IIS (або інший сервер), СУБД визначається встановлюється та налаштовується розробником (ліцензування СУБД виконується замовником).

Вимоги до програмного забезпечення робочої станції:

ОС - Windows не нижче Windows XP.

3.5. Вимоги до складу і параметрів технічних засобів.

Вимоги до складу технічних засобів:

- Сервер повинен мати базову конфігурацію з підтримкою технології RAID, можливість для замінного зберігання даних та можливість підключення до локальної мережі.
- Робоча станція має бути оснащена процесором Intel або AMD зі стандартним набором комплектації, а також засобами для підключення до локальної мережі. Мінімальні технічні характеристики включають процесор Pentium III з тактовою частотою 750 МГц, оперативну пам'ять об'ємом 128 Мбайт та дисків з пам'яттю 20 Гбайт.

Вимоги до програмної документації:

Програмна документація повинна містити наступні дані:

- "Інструкція по інсталяції програми", яка складається з опису процесу встановлення програми, переліку кроків та послідовності їх виконання, опису встановлених програмних компонентів та їх робочих режимів після встановлення. Ця інструкція зазвичай розміщується у четвертому розділі пояснювальної записки до курсової роботи.

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижмотря О.В.				72
Змн.	Арк.	№ докум.	Підпис	Дата		

- "Керівництво користувача", яке включає опис процесу запуску програми, основних режимів роботи, опис екранних форм, перелік виняткових ситуацій та рекомендації щодо їх вирішення, а також порядок виконання завдань у системі. Ця інструкція зазвичай розміщується у третьому розділі пояснювальної записки до курсової роботи.

- "Керівництво адміністратора баз даних", яке включає опис структури таблиць бази даних та правил доступу до них, інструкції щодо обслуговування бази даних (наприклад, архівація, створення резервних копій), налаштування сервера та клієнтських додатків. Ця інструкція зазвичай розміщується у другому, третьому та четвертому розділах пояснювальної записки до курсової роботи відповідно до плану.

Додаток Б

Основний клас dbcontext.db

```
import pyodbc
from ClassLib.clients import Clients

class DbContext:
    def __init__(self):
        self.conn = pyodbc.connect("DRIVER={ODBC Driver 17 for SQL
Server};SERVER=DESKTOP-TSMTSMN;DATABASE=avtovokzal;Trusted_Connection=yes")
        self.cursor = self.conn.cursor()

    def get_items(self, other_class_instance, newquery=""):
        array_items = list(other_class_instance.__dict__.keys())
        table = other_class_instance.__class__.__name__
        query = f"SELECT {'', '.join(array_items)} FROM {table}"

        if newquery != "":
            query = newquery
        print(query)
        try:
            self.cursor.execute(query)
            rows = self.cursor.fetchall()
            result = []

            if rows:
                for row in rows:
                    instance = other_class_instance.__class__()
                    for idx, item in enumerate(array_items):
                        value = row[idx]
                        setattr(instance, item, value if value is not None else "")
```

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижморя О.В.				73
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        result.append(instance)
        print("Item Get succes")
        return result
    else:
        return None
except Exception as e:
    print(f"Error executing SQL query: {e}")
    return None

def get_item_by_id(self, other_class_instance, item_id):
    array_items = list(other_class_instance.__dict__.keys())
    table = other_class_instance.__class__.__name__

    for item in array_items:
        if item.startswith("id"):
            id_class_name = item
            break

    query = f"SELECT {'', '.join(array_items)} FROM {table} WHERE {id_class_name} = ?"

    self.cursor.execute(query, (item_id,))

    row = self.cursor.fetchone()
    instance = None
    if row:
        instance = other_class_instance.__class__()
        for idx, item in enumerate(array_items):
            value = row[idx]
            setattr(instance, item, value if value is not None else None)

    return instance

def add_item(self, obj):
    db_context = DbContext()
    db_context.select_last_index(obj)

    array_items = list(obj.__dict__.keys())
    non_id_items = [item for item in array_items if not item.startswith("id")]

    values = ', '.join(
        [f"'{obj.__dict__[item]}'" if obj.__dict__[item] != None else "NULL" for
        item in non_id_items])

    query = f"INSERT INTO {obj.__class__.__name__} ({', '.join(non_id_items)})
VALUES ({values})"
    print(query)

    try:
        self.cursor.execute(query)

```

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижмоторя О.В.				74
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        self.conn.commit()

        print(f"Item added successfully to {obj.__class__.__name__} table.")
        return True
    except Exception as e:
        print(f"Error adding item to {obj.__class__.__name__} table: {str(e)}")
        return False

def select_last_index(self, obj):
    array_items = list(obj.__dict__.keys())
    for item in array_items:
        if item.startswith("id"):
            id_class_name = item
            break

    query = f"SELECT MAX({str(id_class_name)}) FROM {obj.__class__.__name__}"

    try:
        self.cursor.execute(query)
        result = self.cursor.fetchone()[0]
        return result if result is not None else 0
    except Exception as e:
        print(f"Error getting last index from {obj.__class__.__name__}: {str(e)}")
        return 0

def get_item_by_other_value(self, other_class_instance, item, value):
    array_items = list(other_class_instance.__dict__.keys())
    table = other_class_instance.__class__.__name__

    query = f"SELECT * FROM {table} WHERE {item} = ?"
    self.cursor.execute(query, (value,))

    row = self.cursor.fetchone()

    if row:
        new_instance = type(other_class_instance)()
        for idx, attribute in enumerate(array_items):
            setattr(new_instance, attribute, row[idx] if row[idx] is not None
else "")

        return new_instance

    return None

def get_items_by_other_value(self, other_class_instance, item, value):
    try:
        array_items = list(other_class_instance.__dict__.keys())
        table = other_class_instance.__class__.__name__

```

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижемотря О.В.				75
Змн.	Арк.	№ докум.	Підпис	Дата		

```

query = f"SELECT * FROM {table} WHERE {item} = ?"
self.cursor.execute(query, (value,))

rows = self.cursor.fetchall()

result = []
if rows:
    for row in rows:
        instance = other_class_instance.__class__()
        for idx, item_name in enumerate(array_items):
            item_value = row[idx]
            print(item_value)
            setattr(instance, item_name, item_value if item_value is not
None else "")
        result.append(instance)
    print("Items retrieved successfully")
    return result
else:
    return None
except Exception as e:
    print(f"Error executing SQL query: {e}")
    return None

def get_user(self, password, item, value):
    query = f"SELECT * FROM clients WHERE {str(item)} = ?"
    self.cursor.execute(query, (str(value),))
    row = self.cursor.fetchone()
    if row:
        user = Clients()
        attributes = list(user.__dict__.keys())
        for attribute, row_value in zip(attributes, row):
            setattr(user, attribute, row_value if row_value is not None else "")
        print(str(user.id_client)+" "+user.name+" "+user.phone+"
"+user.password+" "+user.email)
        if user.password == password:
            print("User authorized")
            return user
        else :
            print("incorect password")
            return "Пароль не співпадає"

    print("None account")
    return None

def update_item(self, id, item, value, obj):
    nametable = obj.__class__.__name__

    if nametable=="Buses":
        name_id='id_bus'
    elif nametable=="Clients":

```

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижемотря О.В.				76
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        name_id='id_client'
    elif nametable=="Timetable":
        name_id='id_journey'
    elif nametable == "Cities":
        name_id = 'id_city'
    elif nametable == "Non_authorized_users":
        name_id = 'id_user'
    elif nametable == "Non_authorized_users":
        name_id = 'id_user'
    elif nametable == "Orders":
        name_id = 'id_order'
    elif nametable == "Tickets":
        name_id = 'id_ticket'
    else:
        print("Не підтримуємий клас")
        print("err on DBContext at 158 line")
        return False

    query = f"UPDATE {nametable} SET {item}=? WHERE {name_id}=?"
    values = (value, id)

    try:
        with self.conn:
            with self.cursor.execute(query, values):
                print(f"Update successful for {item} with value {value} for
record with {name_id}={id} in table {nametable}.")
                return True
    except Exception as e:
        print(f"Error updating record: {e}")
        return False

def delete_item(self, obj, item, value):
    nametable = obj.__class__.__name__
    query = f"DELETE FROM {nametable} WHERE {item} = ?"
    values = (value,)

    try:
        with self.conn:
            with self.cursor.execute(query, values):
                print(f"Delete successful from {nametable} where {item} =
{value}.")
                return True
    except Exception as e:
        print(f"Error deleting record: {e}")
        return False

```

Вихідний код проекту

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижмотря О.В.				77
Змн.	Арк.	№ докум.	Підпис	Дата		

Program.py

```
import os
from flask import Flask, render_template, request, session, redirect, abort, url_for
from dbcontext import DbContext
from ClassLib.buses import Buses
from ClassLib.cities import Cities
from ClassLib.clients import Clients
from ClassLib.tickets import Tickets
from ClassLib.timetable import Timetable
from ClassLib.orders import Orders
from ClassLib.journeystable import JourneysTable
from ClassLib.non_authorized_users import Non_authorized_users
from ClassLib.cardticket import CardTicket
from datetime import datetime
import random
import time

app = Flask(__name__)
app.secret_key = os.urandom(24)

db_context = DbContext()

@app.route('/buyticket/', methods=['GET', 'POST'])
def buyticket():
    if request.method == "POST":
        return redirect("/home")
    return redirect("/formbuyticket")

@app.route('/formbuyticket', methods=['GET', 'POST'])
def formbuyticket():
    if request.method == "GET":
        if (request.args.get('citystart') and request.args.get('cityfinish')
            and request.args.get('starttime') and
            request.args.get('finishtime')):
            query = (
                f"SELECT timetable.id_journey, buses.name AS bus_name, start_city.name AS "
                f"start_city, finish_city.name AS finish_city, timetable.time_start, "
                f"timetable.time_finish, timetable.cost FROM timetable "
                f"JOIN cities AS start_city ON timetable.city_start_id = "
                f"start_city.id_city "
                f"JOIN cities AS finish_city ON timetable.city_finish_id = "
                f"finish_city.id_city "
                f"JOIN buses ON timetable.bus_id = buses.id_bus ")
            if request.args.get('citystart') == "Відправляємося з":
                pass
            elif request.args.get('citystart'):
                citystart = request.args.get('citystart')
```

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижморя О.В.				78
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        query += f" WHERE start_city.name LIKE '{citystart}'"
    if request.args.get('cityfinish') == "Їдем в":
        pass
    elif request.args.get('cityfinish'):
        cityfinish = request.args.get('cityfinish')
        query += f" AND finish_city.name LIKE '{cityfinish}'"
    if request.args.get('starttime') and request.args.get('finishtime'):
        starttime = request.args.get('starttime')
        finishtime = request.args.get('finishtime')
        starttime = datetime.strptime(starttime, '%Y-%m-%dT%H:%M')
        finishtime = datetime.strptime(finishtime, '%Y-%m-%dT%H:%M')
        query += f" AND timetable.time_start BETWEEN '{starttime}' AND
'{finishtime}'"
    table = db_context.get_items(JourneysTable(), newquery=query)
    if table is not None:
        if request.args.get('name') and request.args.get('phone'):
            non_auth_user=Non_authorized_users()
            non_auth_user.name=request.args.get('name')
            non_auth_user.phone=request.args.get('phone')
            return render_template("tablejourneys.html", user=GetFromDict(),
table=table,non_auth_user=non_auth_user)
        else:
            return render_template("tablejourneys.html", user=GetFromDict(),
table=table,non_auth_user=None)
    else:
        cities = db_context.get_items(Cities())
        return render_template("formbuyticket.html", user=GetFromDict(),
cities=cities)

cities = db_context.get_items(Cities())
return render_template("formbuyticket.html", user=GetFromDict(), cities=cities)

@app.route("/tablejourneys",methods=['GET','POST'])
def tablejourneys():
    pass

@app.route('/selectplace/', methods=['POST'])
def selectplace():
    if request.method == "POST":
        print('POST')
        non_auth_user = None
        if request.form.get('phone') and request.form.get('name'):
            non_auth_user = Non_authorized_users()
            non_auth_user.phone = request.form.get('phone')
            non_auth_user.name = request.form.get('name')
        id_journey = request.form.get('id_journey')
        bus_name = request.form.get('bus_name').strip()

```

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижемотря О.В.				79
Змн.	Арк.	№ докум.	Підпис	Дата		

```

start_city = request.form.get('start_city')
finish_city = request.form.get('finish_city')
print("name" + bus_name)
selected_bus = Buses() # Змінилася назва змінної bus на selected_bus
selected_bus = db_context.get_item_by_other_value(selected_bus, 'name',
bus_name)
print("seat"+str(selected_bus.seats))
print("name"+str(selected_bus.name))
print("bus_number"+str(selected_bus.bus_number))

journey = Timetable()
journey = db_context.get_item_by_id(journey, id_journey)
tickets = None
query = f"SELECT * FROM tickets WHERE tickets.journey_id Like {id_journey}"
if db_context.get_items(Tickets(), newquery=query) is not None:
    tickets = db_context.get_items(Tickets(), newquery=query)
countstickets = 0
greyseats = None
if tickets is not None:
    countstickets = len(tickets)
    greyseats = []
    for ticket in tickets:
        for seat in range(selected_bus.seats): # Використовується
selected_bus замість bus
            if ticket.seat == seat:
                greyseats.append(seat)

    if non_auth_user is not None:
        return render_template("selectplace.html", user=GetFromDict(),
non_auth_user=non_auth_user,
                                bus=selected_bus,
                                journey=journey,
                                start_city=start_city, finish_city=finish_city,
tickets=tickets,
                                countstickets=countstickets, greyseats=greyseats)
    else:
        return render_template("selectplace.html", user=GetFromDict(),
non_auth_user=None, bus=selected_bus,
                                journey=journey,
                                start_city=start_city, finish_city=finish_city,
tickets=tickets,
                                countstickets=countstickets, greyseats=greyseats)

return redirect('/formbuyticket')

@app.route('/pay/', methods=['POST', 'GET'])
def pay():
    if request.method == "POST":
        id_journey = request.form.get('id_journey')

```

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижмоторя О.В.				80
Змн.	Арк.	№ докум.	Підпис	Дата		


```

seat = request.form.get('seat')
non_auth_user = None
if request.form.get('phone') and request.form.get('name'):
    non_auth_user = Non_authorized_users()
    non_auth_user.phone = request.form.get('phone')
    non_auth_user.name = request.form.get('name')
else:
    user = GetFromDict()

journey = Timetable()
journey = db_context.get_item_by_id(journey, id_journey)

bus = Buses()
bus = db_context.get_item_by_id(bus, journey.bus_id)

start_city = Cities()
finish_city = Cities()
start_city = db_context.get_item_by_id(start_city, journey.city_start_id)
finish_city = db_context.get_item_by_id(finish_city, journey.city_finish_id)

return render_template('pay.html',user=GetFromDict(), journey=journey,
bus=bus, seat=seat, start_city=start_city,
                                finish_city=finish_city, non_auth_user=non_auth_user)
else:
    return redirect('/formbuyticket')

@app.route('/succesorder/', methods=['POST'])
def succesorder():
    if request.method == 'POST':
        non_auth_user = None
        if request.form.get('phone') and request.form.get('name'):
            non_auth_user = Non_authorized_users()
            non_auth_user.phone = request.form.get('phone')
            non_auth_user.name = request.form.get('name')
            if db_context.add_item(non_auth_user):
                non_auth_user.id_user=db_context.select_last_index(non_auth_user)

        id_journey = request.form.get('id_journey')
        seat = request.form.get('seat')

        journey = Timetable()
        journey = db_context.get_item_by_id(journey, id_journey)
        #створення нового ордеру
        neworder=Orders()
        #заповнення даних
        if non_auth_user is None:
            user = GetFromDict()
            user=db_context.get_item_by_other_value(user,'email',user.email)

```

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижморя О.В.				81
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        print(user.id_client)
        neworder.client_id = user.id_client
    else:
        neworder.non_authorized_users_id = non_auth_user.id_user
        neworder.date_buying = neworder.date_buying =
datetime.strptime(str(datetime.utcnow().replace(microsecond=0)), '%Y-%m-%d
%H:%M:%S')
        neworder.cost = journey.cost

        db_context.add_item(neworder)
#дістаєм id останнього ордеру
        indexLastOrder = db_context.select_last_index(neworder)
#створення нового екземпляру ордеру та діставання даних з БД
        order = Orders()
        order = db_context.get_item_by_id(order, indexLastOrder)
#перевірка ордерів між собою
        if (order.client_id == neworder.client_id and \
            order.non_authorized_users_id == neworder.non_authorized_users_id
and \
            str(order.date_buying) == str(neworder.date_buying) and \
            order.cost == neworder.cost):
#зміна статусу оплати
            whetherpaid = db_context.update_item(order.id_order, 'whether_paid', 1,
order)

            if whetherpaid:
                # заповнення даних квитка
                newticket = Tickets()
                newticket.journey_id = journey.id_journey
                newticket.bus_id = journey.bus_id
                newticket.order_id = order.id_order
                newticket.client_id = order.client_id
                newticket.date_buying =
datetime.strptime(str(datetime.utcnow().replace(microsecond=0)), '%Y-%m-%d
%H:%M:%S')

                newticket.code=GenerateRandomNumber()
                if non_auth_user is None:
                    newticket.client_id = user.id_client
                else:
                    newticket.non_authorized_users_id = non_auth_user.id_user
                newticket.seat=seat
                iffer = db_context.add_item(newticket)
                if iffer:
                    bus = Buses()
                    bus = db_context.get_item_by_id(bus, journey.bus_id)
                    start_city = Cities()
                    finish_city = Cities()
                    start_city = db_context.get_item_by_id(start_city,
journey.city_start_id)
                    finish_city = db_context.get_item_by_id(finish_city,
journey.city_finish_id)

```

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижморя О.В.				82
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        if iffer is None:
            return render_template('succesorder.html', title="Succes",
user=GetFromDict(), non_auth_user=non_auth_user,

start_city=start_city,finish_city=finish_city,journey=journey, bus=bus, seat=seat,
ticket=newticket)
        else:
            return render_template('succesorder.html',
title="Succes",user=GetFromDict(), non_auth_user=non_auth_user,

start_city=start_city,finish_city=finish_city,journey=journey, bus=bus, seat=seat,
ticket=newticket)
        else:
            return render_template('succesorder.html',
title=None,user=GetFromDict())
        else:
            pass
    else:
        return redirect('/formbuyticket')
    return render_template('succesorder.html', title=None)

@app.route("/getticket",methods=["POST"])
def getticket():
    if request.form.get('ticket_id'):
        ticket_id=request.form.get('ticket_id')
        ticket=Tickets()
        if db_context.get_item_by_other_value(ticket,'code',ticket_id):
            ticket=db_context.get_item_by_other_value(ticket,'code',ticket_id)
            journey=Timetable()
            journey=db_context.get_item_by_id(journey,ticket.journey_id)
            start_city = Cities()
            finish_city = Cities()
            start_city = db_context.get_item_by_id(start_city,
journey.city_start_id)
            finish_city = db_context.get_item_by_id(finish_city,
journey.city_finish_id)
            bus=Buses()
            bus=db_context.get_item_by_id(bus,ticket.bus_id)
            if ticket.client_id is not None:
                user=Clients()
                user=db_context.get_item_by_id(user,ticket.client_id)
                nonuser=None
            else:
                nonuser = Non_authorized_users()
                nonuser = db_context.get_item_by_id(nonuser,
ticket.non_authorized_users_id)
                user=None
            return render_template('succesorder.html', title="Succes", user=user,
non_auth_user=nonuser,

```

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижмотря О.В.				83
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        start_city=start_city, finish_city=finish_city,
journey=journey, bus=bus, seat=ticket.seat,
        ticket=ticket)

    else:
        return redirect('/home')

@app.route('/journeys', methods=['GET','POST'])
def journeys():
    for req in request.args:
        print(request.args[req])
        print('DA')

    try:
        if ((request.args.get('item_sort') and request.args.get('direction') and
request.args.get('citystart')) or
            (request.args.get('citystart') and request.args.get('cityfinish'))
or
            (request.args.get('item_sort') and request.args.get('direction') and
request.args.get('citystart')and request.args.get('cityfinish'))):
            item_sort=""
            direction=""
            if request.args.get('item_sort'):
                item_sort = request.args.get('item_sort')
            if request.args.get('direction'):
                direction = request.args.get('direction')
            query = (
                f"SELECT timetable.id_journey,buses.name AS bus_name,
start_city.name AS start_city, finish_city.name AS finish_city,
timetable.time_start, timetable.time_finish,timetable.cost FROM timetable "
                f"JOIN cities AS start_city ON timetable.city_start_id =
start_city.id_city "
                f"JOIN cities AS finish_city ON timetable.city_finish_id =
finish_city.id_city "
                f"JOIN buses ON timetable.bus_id = buses.id_bus ")
            if request.args.get('citystart') == "Відправляємося з":
                pass
            elif request.args.get('citystart'):
                citystart = request.args.get('citystart')
                query += f" WHERE start_city.name LIKE '{citystart}'"
            if request.args.get('cityfinish')== "Їдемо в":
                pass
            elif request.args.get('cityfinish'):
                cityfinish = request.args.get('cityfinish')
                query += f" AND finish_city.name LIKE '{cityfinish}'"
            if request.args.get('starttime') and request.args.get('finishtime'):
                starttime = request.args.get('starttime')
                finishtime = request.args.get('finishtime')

                starttime = datetime.strptime(starttime, '%Y-%m-%dT%H:%M')
                finishtime = datetime.strptime(finishtime, '%Y-%m-%dT%H:%M')

```

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижморя О.В.				84
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        query += f" AND timetable.time_start BETWEEN '{starttime}' AND
'{finishtime}'"

    if item_sort == "Назва автобусу":
        query += " ORDER BY bus_name"
    elif item_sort == "За іменем міста відправлення":
        query += " ORDER BY start_city"
    elif item_sort == "За іменем міста прибуття":
        query += " ORDER BY finish_city"
    elif item_sort == "За часом відправлення":
        query += " ORDER BY time_start"
    elif item_sort == "За часом прибуття":
        query += " ORDER BY time_finish"
    else:
        return render_template('Error', 500)

    if direction == "По спаданню":
        query += " DESC"
    elif direction == "По зростанню":
        query += " ASC"

    table = db_context.get_items(JourneysTable(), newquery=query)
    cities = db_context.get_items(Cities())
    return render_template("journeys.html", user=GetFromDict(), table=table,
cities=cities)

except Exception as e:
    print(f"An error occurred: {str(e)}")

buses = db_context.get_items(Buses())
journeys = db_context.get_items(Timetable())
cities = db_context.get_items(Cities())
return render_template("journeys.html", user=GetFromDict(), buses=buses,
journeys=journeys, cities=cities,
                        table=None)

@app.route('/')
def firstroute():
    return render_template("home.html",user=GetFromDict())

@app.route('/home')
def home():
    return render_template("home.html", user=GetFromDict())

@app.route('/About')
def about():
    return render_template("about.html",user=GetFromDict())

```

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижмотря О.В.				85
Змн.	Арк.	№ докум.	Підпис	Дата		

```

@app.route('/loginselect')
def loginselect():
    return render_template("loginselect.html",user=None)

@app.route('/login/<data>',methods=['GET','POST'])
def login(data):
    if request.method == "POST":
        item=str(data)
        password= request.form['floatingPassword']
        if data == "email":
            value = request.form['floatingInputEmail']
            if CreateUser(password, item, value)=="Пароль не співпадає":
                return render_template('login.html', data=data, user=None, info="Не вірний пароль")
            elif CreateUser(password, item, value):
                return redirect("/home")
            else:
                return render_template('login.html', data=data, user=None, info="")
        elif data == "phone":
            value = request.form['floatingInputPhone']
            if CreateUser(password,item, value):
                return redirect("/home")
            else:
                return render_template('login.html', data=data, user=None,info="")
    if request.method == "GET":
        return render_template('login.html', data=data, user=None,info="")
    else:
        return redirect("/home")

@app.route('/register' ,methods=['POST','GET'])
def register():
    if request.method == "POST":
        if GetFromDict() is None:
            name = str(request.form['floatingInputName'])
            email=str(request.form['floatingInputEmail'])
            phone=str(request.form['floatingInputPhone'])
            password=str(request.form['floatingInputPassTwo'])
            role_client='user'
            user=Clients()
            if name:
                user.name = name
            if email:
                user.email = email
            if phone:
                user.phone = phone
            if password:
                user.password = password

```

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижмотря О.В.				86
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        if role_client:
            user.role_client=role_client
        if(db_context.add_item(user)):
            AddToDict(user)
            return redirect("/home")
        else:
            return render_template("register.html", user=None,info="Account not
register")
    else:
        return redirect("/home")
    if request.method == "GET":
        return render_template("register.html",user=None,info=None)

@app.route('/accountview')
def accountview():
    user = GetFromDict()
    if user:
        return render_template("accountview.html", user=user)
    else:
        return redirect("/home")

@app.route('/exit')
def exit():
    DeleteFromDict()
    return redirect('/home');

@app.route('/usertickets')
def usertickets():
    user = GetFromDict()
    if user:
        #потрібно дізнатись id user
        user = db_context.get_item_by_other_value(user, 'email', user.email)
        tickets=None
        if db_context.get_items_by_other_value(Tickets(), 'client_id',
user.id_client):
            tickets = db_context.get_items_by_other_value(Tickets(), 'client_id',
user.id_client)
            if tickets is not None:
                for ticket in tickets:
                    print(ticket.id_ticket)
            oldtickets = []
            recenttickets = []

            if tickets is not None:
                for ticket in tickets:
                    card=CardTicket()

                    journeyy = Timetable()

```

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижмотря О.В.				87
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        journeyy = db_context.get_item_by_id(journeyy, ticket.journey_id)
        journeys = []
        journeys.append(journeyy)

        card.id_ticket=ticket.id_ticket
        card.code=ticket.code
        card.cost=journeyy.cost

        time_start = str(journeyy.time_start)
        time_finish = str(journeyy.time_finish)

        time_start_obj = datetime.strptime(time_start, "%Y-%m-%d %H:%M:%S")
        time_finish_obj = datetime.strptime(time_finish, "%Y-%m-%d
%H:%M:%S")

        formatted_time_start = time_start_obj.strftime("%d.%m.%Y %H:%M:%S")
        formatted_time_finish = time_finish_obj.strftime("%d.%m.%Y
%H:%M:%S")

        str_time = f"{formatted_time_start} - {formatted_time_finish}"

        card.time = str_time

        for journey in journeys:
            city_start=Cities()
            city_finish=Cities()

city_start=db_context.get_item_by_id(city_start,journey.city_start_id)

city_finish=db_context.get_item_by_id(city_finish,journey.city_finish_id)
            str_city=str(city_start.name)+" - " + str(city_finish.name)
            card.name=str_city

            current_time = datetime.utcnow().replace(second=0)
            if ticket.journey_id==journey.id_journey:
                if journey.time_start < current_time:
                    oldtickets.append(card)
                else:
                    recenttickets.append(card)

        return render_template("usertickets.html", user=user,
oldtickets=oldtickets, recenttickets=recenttickets)
    else:
        return render_template("usertickets.html", user=user, oldtickets=None,
recenttickets=None)
    else:
        return redirect("/home")

@app.route('/addbus', methods=['POST','GET'])

```

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижморя О.В.				88
Змн.	Арк.	№ докум.	Підпис	Дата		


```

def addbus():
    if request.method=="GET":
        if GetFromDict() is not None:
            return render_template("addbus.html", info=None)
        else:
            return redirect("/home")
    if request.method == "POST":
        bus=Buses()
        bus.name=str(request.form['nameBus'])
        bus.seats=str(request.form['seats'])
        bus.bus_number=str(request.form['busNumber'])
        if db_context.add_item(bus):
            return render_template("addbus.html", info=None)
        else:
            return render_template("addbus.html", info="Не додано")

@app.route('/addcity', methods=['POST','GET'])
def addcity():
    if request.method=="GET":
        if GetFromDict() is not None:
            return render_template("addcity.html", info=None)
        else:
            return redirect("/home")
    if request.method=="POST":
        city=Cities()
        city.name=str(request.form['nameCity'])
        city.country=str(request.form['country'])
        if db_context.add_item(city):
            return render_template("addcity.html", info=None)
        else:
            return render_template("addcity.html", info="Не додано")

@app.route('/addjourney', methods=['POST', 'GET'])
def addjourney():
    if request.method == "GET":
        if GetFromDict() is not None:
            cities = db_context.get_items(Cities())
            buses = db_context.get_items(Buses())
            return render_template("addjourney.html", info=None, buses=buses,
cities=cities)
        else:
            return redirect("/home")

    if request.method == "POST":
        journey = Timetable()
        journey.bus_id = int(request.form['idbus'])
        journey.city_start_id = int(request.form['idcitystart'])
        journey.city_finish_id = int(request.form['idcityfinish'])

```

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижмотря О.В.				89
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        journey.cost = float(request.form['cost'])
        journey.time_start = datetime.strptime(request.form['timestart'], '%Y-%m-%dT%H:%M')
        journey.time_finish = datetime.strptime(request.form['timefinish'], '%Y-%m-%dT%H:%M')
        journey.seats_occupied=int(request.form['seats_occupied'])
        cities = db_context.get_items(Cities())
        buses = db_context.get_items(Buses())
        if db_context.add_item(journey):
            return render_template("addjourney.html", info=None, buses=buses,
cities=cities)
        else:
            return render_template("addjourney.html", info="Не додано", buses=buses,
cities=cities)

@app.route('/addadmin', methods=['POST', 'GET'])
def addadmin():
    if request.method == "GET":
        if GetFromDict() is not None:
            return render_template("addadmin.html", info=None)
        else:
            return redirect("/home")

    if request.method == "POST":
        user = Clients()
        user.name=str(request.form['name'])
        user.email=str(request.form['email'])
        user.phone=str(request.form['phone'])
        user.password=str(request.form['password'])
        user.role_client='admin'
        if db_context.add_item(user):
            return render_template("addadmin.html", info=None)
        else:
            return render_template("addadmin.html", info="Не додано")

@app.route('/delete', methods=['POST','GET'])
def delete():
    if request.method == "GET":
        if GetFromDict():
            if GetFromDict().role_client == "admin":
                return render_template("delete.html")
            else:
                return redirect("/home")
    if request.method == "POST":
        selected_radio = request.form.get('listGroupCheckableRadios')

        if selected_radio == "bus":

```

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижемотря О.В.				90
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        bus = Buses()
        fieldlist = GetClassFields(bus)
        return render_template("deletewhatrecord.html", fieldlist=fieldlist,
clas=bus)
    elif selected_radio == "journey":
        journey = Timetable()
        fieldlist = GetClassFields(journey)
        return render_template("deletewhatrecord.html", fieldlist=fieldlist,
clas=journey)
    elif selected_radio == "city":
        city = Cities()
        fieldlist = GetClassFields(city)
        return render_template("deletewhatrecord.html", fieldlist=fieldlist,
clas=city)
    elif selected_radio == "ticket":
        tick = Tickets()
        fieldlist = GetClassFields(tick)
        return render_template("deletewhatrecord.html", fieldlist=fieldlist,
clas=tick)
    elif selected_radio == "client":
        client = Clients()
        fieldlist = GetClassFields(client)
        return render_template("deletewhatrecord.html", fieldlist=fieldlist,
clas=client)

    return redirect("/adminpanel")

@app.route('/deletepost/<selectedfield>/<clas>', methods=['POST', 'GET'])
def deletepost(selectedfield, clas):
    if request.method == "GET":
        if GetFromDict() is not None:
            if GetFromDict().role_client == "admin":
                ss = selectedfield
                return render_template("deletepost.html", selectedfield=ss,
clas=clas, info=None)
            else:
                return redirect("/home")

    if request.method == "POST":
        item = selectedfield
        value = request.form['value']
        type=request.form['type']

        try:
            obj_class = globals()[clas]
            obj = obj_class()
        except KeyError:
            return render_template("deletepost.html", selectedfield=selectedfield,
clas=clas,
                                info="Не вдалося знайти клас")

```

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижморя О.В.				91
Змн.	Арк.	№ докум.	Підпис	Дата		

```

try:
    if type == "datetime":
        return render_template("deletepost.html",
selectedfield=selectedfield, clas=clas,
                                info="За цим полем не можна видалити")
    except ValueError:
        return render_template("deletepost.html", selectedfield=selectedfield,
clas=clas,
                                info="Помилка конвертації у формат дати/часу")
try:
    if type == "int":
        value = int(value)
    except ValueError:
        return render_template("deletepost.html", selectedfield=selectedfield,
clas=clas,
                                info="Помилка конвертації у формат int")

    if db_context.delete_item( obj, item, value):
        return render_template("adminpanel.html")
    else:
        return render_template("deletepost.html", selectedfield=selectedfield,
clas=clas, info="Не видалено")

@app.route('/update', methods=['POST','GET'])
def update():
    if request.method=="GET":
        if GetFromDict():
            if GetFromDict().role_client == "admin":
                return render_template("update.html")
            else:
                return redirect("/home")
    if request.method=="POST":
        selected_radio = request.form.get('listGroupCheckableRadios')

        if selected_radio == "bus":
            bus=Buses()
            fieldlist=GetClassFields(bus)
            return render_template("updatewhatrecord.html",
fieldlist=fieldlist,clas=bus)
        elif selected_radio == "journey":
            journey=Timetable()
            fieldlist = GetClassFields(journey)
            return render_template("updatewhatrecord.html",
fieldlist=fieldlist,clas=journey)
        elif selected_radio == "city":
            city=Cities()
            fieldlist = GetClassFields(city)
            return render_template("updatewhatrecord.html",
fieldlist=fieldlist,clas=city)

```

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижморя О.В.				92
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        elif selected_radio == "ticket":
            tick=Tickets()
            fieldlist = GetClassFields(tick)
            return render_template("updatewhatrecord.html",
fieldlist=fieldlist,clas=tick)

        return redirect("/adminpanel")

@app.route('/updatewhatrecord', methods=['POST', 'GET'])
def updatewhatrecord():
    if request.method == "GET":
        if GetFromDict() is not None:
            if GetFromDict().role_client == "admin":
                return render_template("updatewhatrecord.html")
            else:
                return redirect("/home")
    elif request.method == "POST":
        clas = request.args.get('clas')
        selected_radio = request.form.get('listGroupCheckableRadios')

        return render_template("updatepost.html" ,dict=dict)

@app.route('/updatepost/<selectedfield>/<clas>', methods=['POST', 'GET'])
def updatepost(selectedfield, clas):
    if request.method == "GET":
        if GetFromDict() is not None:
            if GetFromDict().role_client == "admin":
                ss = selectedfield
                return render_template("updatepost.html", selectedfield=ss,
clas=clas, info=None)
            else:
                return redirect("/home")

    if request.method == "POST":
        item = selectedfield
        value = request.form['value']
        type=request.form['type']
        id = request.form['id']

        try:
            obj_class = globals()[clas]
            obj = obj_class()
        except KeyError:
            return render_template("updatepost.html", selectedfield=selectedfield,
clas=clas,
                                info="Не вдалося знайти клас")

        try:
            if type == "datetime":

```

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижморя О.В.				93
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        value = datetime.strptime(value, '%Y-%m-%dT%H:%M')
    except ValueError:
        return render_template("updatepost.html", selectedfield=selectedfield,
                                clas=clas,
                                info="Помилка конвертації у формат дати/часу")
    try:
        if type == "int":
            value = int(value)
    except ValueError:
        return render_template("updatepost.html", selectedfield=selectedfield,
                                clas=clas,
                                info="Помилка конвертації у формат int")

    if db_context.update_item(id, item, value, obj):
        return render_template("adminpanel.html")
    else:
        return render_template("updatepost.html", selectedfield=selectedfield,
                                clas=clas, info="Не оновлено")

@app.route('/adminpanel')
def adminpanel():
    if GetFromDict():
        user = GetFromDict()
        if user.role_client == "admin":
            return render_template("adminpanel.html", user=user)
    return redirect("/home")

def CreateUser(password,item, value):
    if db_context.get_user(password, item, value)=="Пароль не співпадає":
        return "Пароль не співпадає"
    elif db_context.get_user(password,item,value):
        user = db_context.get_user(password,item,value)
        AddToDict(user)
        return True
    else:
        return False

def GetClassFields(obj):
    return list(obj.__dict__.keys())

def GenerateRandomNumber():
    return ''.join(random.choices('0123456789', k=8))

def GetFromDict():
    if session.get('user', {}):
        user_dict = session.get('user', {})
        user = Clients.from_dict(user_dict)

```

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижмоторя О.В.				94
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        return user
    else:
        return None

def AddToDict(user=Clients()):
    user_dict = user.to_dict()
    session['user'] = user_dict

def DeleteFromDict():
    if 'user' in session:
        del session['user']

if __name__ == "__main__":
    app.run(debug=True)

```

buses.py

```

class Buses:
    def __init__(self):
        self.id_bus = None
        self.name = None
        self.seats = None
        self.bus_number = None

```

cardticket.py

```

class CardTicket:
    def __init__(self):
        self.id_ticket=None
        self.code = None
        self.name=None
        self.time=None
        self.cost=None

```

cities.py

```

class Cities:
    def __init__(self):
        self.id_city = None
        self.name = None
        self.country=None

```

clients.py

```

class Clients:
    def __init__(self):
        self.id_client = None
        self.name = None
        self.phone = None
        self.email = None
        self.password = None
        self.role_client = None

    def to_dict(self):
        return {

```

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижморя О.В.				95
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        'id': self.id_client,
        'name': self.name,
        'email': self.email,
        'phone': self.phone,
        'password': self.password,
        'role_client': self.role_client
    }

```

```

@classmethod
def from_dict(cls, user_dict):
    user = cls()
    user.id_client = user_dict.get('id_client', '')
    user.name = user_dict.get('name', '')
    user.email = user_dict.get('email', '')
    user.phone = user_dict.get('phone', '')
    user.password = user_dict.get('password', '')
    user.role_client = user_dict.get('role_client', '')
    return user

```

journeystable.py

```

class JourneysTable:
    def __init__(self):
        self.id_journey=None
        self.bus_name = None
        self.start_city = None
        self.finish_city = None
        self.time_start = None
        self.time_finish = None
        self.cost = None

```

```

    def getJourneyId(self):
        return self.id_journey

```

non_authorized_users.py

```

class Non_authorized_users:
    def __init__(self):
        self.id_user = None
        self.name = None
        self.phone = None

```

orders.py

```

class Orders:
    def __init__(self):
        self.id_order = None
        self.client_id = None
        self.non_authorized_users_id = None
        self.date_buying = None
        self.cost = None
        self.whether_paid = None

```

tickets.py

```

class Tickets:
    def __init__(self):

```

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижмоторя О.В.				96
Змн.	Арк.	№ докум.	Підпис	Дата		


```

self.id_ticket = None
self.bus_id = None
self.seat=None
self.date_buying = None
self.journey_id = None
self.non_authorized_users_id = None
self.client_id = None
self.order_id = None
self.code=None

```

timetable.py

```
import datetime
```

```

class Timetable:
    def __init__(self):
        self.id_journey = None
        self.bus_id = None
        self.city_start_id = None
        self.city_finish_id = None
        self.cost = None
        self.time_start = None
        self.time_finish = None
        self.seats_occupied=None
        self.is_active=None

```

```

def getBusId(self):
    return self.id_journey

```

about.html

```
{% extends 'base.html' %}
```

```
{% block title %}
```

Про нас

```
{% endblock %}
```

```
{% block body %}
```

```
<div class="b-example-divider"></div>
```

```
<div class="p-5 text-center bg-body-secondary">
```

```
<div class="container px-4 py-5 " id="hanging-icons">
```

```
<h2 class="pb-2 border-bottom">Про нас:</h2>
```

```
<div class="row g-4 py-5 row-cols-1 row-cols-lg-4">
```

```
<div class="col d-flex align-items-start">
```

```
<div style="" class="icon-square text-body-emphasis y d-inline-flex align-items-center justify-content-center fs-4 flex-shrink-0 me-3">
```

```

```

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижемотря О.В.				97
Змн.	Арк.	№ докум.	Підпис	Дата		

```

</div>
<div>
  <h3 class="fs-3 text-body-emphasis">Розклад автобусів</h3>
  <p>Переглядайте актуальний розклад руху автобусів, обирайте зручний час
відправлення та призначення.</p>
</div>
</div>
<div class="col d-flex align-items-start">
  <div class="icon-square text-body-emphasis d-inline-flex align-items-center
justify-content-center fs-4 flex-shrink-0 me-3">
    
  </div>
  <div>
    <h3 class="fs-3 text-body-emphasis">Пошук маршрутів</h3>
    <p>Знаходьте оптимальні маршрути між містами та вибирайте найзручніші
варіанти подорожі.</p>
  </div>
</div>
<div class="col d-flex align-items-start">
  <div class="icon-square text-body-emphasis d-inline-flex align-items-center
justify-content-center fs-4 flex-shrink-0 me-3">
    
  </div>
  <div>
    <h3 class="fs-3 text-body-emphasis">Квитки онлайн</h3>
    <p>Бронюйте та оплачуйте квитки прямо на сайті, уникайте черг та
забезпечте собі місце в потрібному вам автобусі.</p>
  </div>
</div>
<div class="col d-flex align-items-start">
  <div class="icon-square text-body-emphasis d-inline-flex align-items-center
justify-content-center fs-4 flex-shrink-0 me-3">
    
  </div>
  <div>
    <h3 class="fs-3 text-body-emphasis">Інформація про автобуси</h3>
    <p>Детальна інформація про автобуси, включаючи кількість місць, номер
автобуса та інше.</p>
  </div>
</div>
</div>
</div>
{% endblock %}

```

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижморя О.В.				98
Змн.	Арк.	№ докум.	Підпис	Дата		

accountview.html

```
{% extends 'base.html' %}

{% block title %}
Аккаунт користувача
{% endblock %}

{% block body %}
<style>
#acc{
padding-left: 5%;
}
#acc>div{
margin-top: 10px;
padding: 25px;
border-bottom: 3px solid black;
border-radius: 25px;
background: linear-gradient(to bottom, #E9ECEF,white);
display: block;
width: 300px;
}
#acc>a{
margin-top: 25px;
}
span{
font-size: 14px;
}
h3{
margin: 0;
}

</style>
<div id="acc" class="bg-body-secondary " >
<div class="text-center" style="display: inline-block">
<h1>Акаунт користувача</h1>
</div><br>
<div>
<h3>ПІБ</h3>
<h3><strong>{{ user.name }}</strong></h3><br>
</div>
<div>
<h3>Email</h3>
<h3><strong>{{ user.email }} </strong></h3><br>
</div>
<div>
<h3>Телефон</h3>
<h3><strong>{{ user.phone }}</strong></h3><br>
</div>
```

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижомотря О.В.				99
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        <a class="btn btn-primary" href="/usertickets">Мої квитки</a>
        <a class="btn btn-primary" href="/exit">Вийти</a>
    </div>

```

```
{% endblock %}
```

addadmin.html

```
{% extends 'baseadmin.html' %}
```

```
{% block title %}
```

Додати поїздки

```
{% endblock %}
```

```
{% block body %}
```

```
{% if info is not none %}
```

```
<script>
```

```
    alert("Адміністратора не додано")
```

```
</script>
```

```
{% endif %}
```

```
<h1>Додавання нового адміністратора</h1>
```

```
<form method="post" >
```

```
    <input type="text" name="name" id="name">
```

```
    <label for="name">Ім'я Адміністратора</label><br>
```

```
    <input type="text" name="email" id="email">
```

```
    <label for="email">Email</label><br>
```

```
    <input type="text" name="phone" id="phone">
```

```
    <label for="phone">Phone</label><br>
```

```
    <input type="text" name="password" id="password">
```

```
    <label for="password">password</label><br>
```

```
    <button type="submit" class="btn btn-primary">Додати</button>
```

```
</form>
```

```
    <a href="/adminpanel" class="btn btn-primary">Назад</a>
```

```
{% endblock %}
```

addbus.html

```
{% extends 'baseadmin.html' %}
```

```
{% block title %}
```

Додати автобус

```
{% endblock %}
```

```
{% block body %}
```

```
{% if info is not none %}
```

```
<script>
```

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижмоторя О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		100

```

        alert("Автобус не додано")
    </script>
{% endif %}
<h1>Додавання нового автобуса</h1>
<form method="post">
    <input type="text" name="nameBus" id="nameBus">
    <label for="nameBus">Назва автобуса</label><br>
    <input type="text" name="seats" id="seats">
    <label for="seats">Кількість місць</label><br>
    <input type="text" name="busNumber" id="seats">
    <label for="busNumber">Номер автобуса</label><br>
    <button type="submit" class="btn btn-primary">Додати</button>
</form>
<a href="/adminpanel" class="btn btn-primary">Назад</a>
{% endblock %}

```

addcity.html

```

{% extends 'baseadmin.html' %}

{% block title %}
Додати місто
{% endblock %}

{% block body %}
    {% if info is not none %}
        <script>
            alert("Місто не додано")
        </script>
    {% endif %}
    <h1>Додавання нового міста</h1>
    <form method="post">
        <input type="text" name="nameCity" id="nameCity">
        <label for="nameCity">Назва міста</label><br>
        <input type="text" name="country" id="country">
        <label for="country">Країна</label><br>
        <button type="submit" class="btn btn-primary">Додати</button>
        <a href="/adminpanel" class="btn btn-primary">Назад</a>
    </form>
{% endblock %}

```

addjourney.html

```

{% extends 'baseadmin.html' %}

{% block title %}
Додати поїздки
{% endblock %}

```

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижмотря О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		101

```

{% block body %}
    {% if info is not none %}
        <script>
            alert("Поїздки не додано")
        </script>
    {% endif %}

    <h1>Додавання нової поїздки</h1>
    <form method="post" >
        <input type="text" name="idbus" id="idbus">
        <label for="idbus">ID Автобуса</label><br>
        <input type="text" name="idcitystart" id="idcitystart">
        <label for="idcitystart">Місто відправлення</label><br>
        <input type="text" name="idcityfinish" id="idcityfinish">
        <label for="idcityfinish">Місто прибуття</label><br>
        <input type="text" name="cost" id="cost">
        <label for="cost">Плата за проїзд</label><br>
        <input type="datetime-local" name="timestart" id="timestart">
        <label for="timestart">Час відправлення</label><br>
        <input type="datetime-local" name="timefinish" id="timefinish">
        <label for="timefinish">Час прибуття</label><br>
        <input type="text" name="seats_occupied" id="seats">
        <label for="seats_occupied">Зайняті місця</label><br>
        <button type="submit" class="btn btn-primary">Додати</button>
    </form>
    <table class="table table-striped">
        <thead>
            <tr>
                <th scope="col">#</th>
                <th scope="col">Назва</th>
                <th scope="col">Місця</th>
                <th scope="col">Місто відправлення</th>
                <th scope="col">Місто прибуття</th>
            </tr>
        </thead>

        <tbody>
            {% for bus in buses %}
                <tr>
                    <th scope="row">{{ bus.id_bus }}</th>
                    <td>{{ bus.name }}</td>
                    <td>{{ bus.seats }}</td>
                    <td>{{ bus.bus_number }}</td>
                    <td>{{ bus.bus_number }}</td>
                </tr>
            {% endfor %}
        </tbody>

    </table>
    <table class="table table-striped">

```

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижмоторя О.В.				102
Змн.	Арк.	№ докум.	Підпис	Дата		

```

<thead>
<tr>
  <th scope="col">#</th>
  <th scope="col">ІД</th>
  <th scope="col">Назва</th>
  <th scope="col">Країна</th>
</tr>
</thead>

<tbody>
{% for city in cities %}
  <tr>
    <th scope="row">{{ city.id_city }}</th>
    <td>{{ city.name }}</td>
    <td>{{ city.country }}</td>
  </tr>
{% endfor %}
</tbody>

</table>

<a href="/adminpanel" class="btn btn-primary">Назад</a>
{% endblock %}

```

adminpanel.html

```

{% extends 'baseadmin.html' %}

{% block title %}
Адмінпанель
{% endblock %}

{% block body %}

<div class="my-5 ">
  <div class="bg-body-secondary my-5 text-center" style="padding: 30px">
    <a href="/addbus" class="btn btn-primary">Додати автобус</a>
    <a href="/addcity" class="btn btn-primary">Додати місто</a>
    <a href="/addjourney" class="btn btn-primary">Додати маршрут</a>
    <a href="/addadmin" class="btn btn-primary">Додати адміна</a>
  </div>
  <div class="bg-body-secondary my-5 text-center" style="padding: 30px">
    <a href="/update" class="btn btn-warning">Оновити записи</a>
    <a href="/delete" class="btn btn-danger">Видалити записи</a>
  </div>
</div>
<div class="bg-body-secondary my-5 text-center" style="padding: 30px">
  <a href="/home" class="btn btn-primary">На головну</a>
</div>

```

```
{% endblock %}
```

base.html

```
<!DOCTYPE html>
<html lang="en">
<script>
    function selectDiv(div) {
        var seatNumber = div.getAttribute('data-seat');

        var seatInput = document.querySelector('input[name="seat"]');
        seatInput.value = seatNumber;

        var busDivs = document.querySelectorAll('.bus > div');
        busDivs.forEach(function (element) {
            if (element.getAttribute('data-seat') === seatNumber) {
                element.style.boxShadow='10px 10px 5px #0D6EFD'
                element.style.backgroundColor = '#0a58ca';
                element.style.color = 'white';
                let labelseat =document.getElementById('labelseat')
                labelseat.innerHTML='Ви вибрали місце : ' + seatInput.value

            } else {
                element.style.backgroundColor = '#0D6EFD';
                element.style.color = 'white';
                element.style.boxShadow=''
            }
        });
    }
</script>
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="shortcut icon" href="{ { url_for('static',
filename='svgs/bus_schedule_icon.svg') }}" type="image/x-icon">
    <link rel="stylesheet" href="{ { url_for('static',filename='css/main.css') }}">
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
T3c6CoIi6uLrA9TneNEoa7RxnatzjcDSCmG1MXxSR1GAsXEV/Dwwykc2MPK8M2HN"
crossorigin="anonymous">
    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js"
integrity="sha384-C6RzsynM9kWDrMNeT87bh95OGNyZPhcTNxj1NW7RuBCsyN/o0jlpcV8Qyq46cDfL"
crossorigin="anonymous"></script>
    <title>{% block title %}{% endblock %}</title>
```

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижмоторя О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		104


```

</head>
<body>
<style>
  header {
    position: fixed;
    top: 0;
    left: 0;
    width: 100%;
    background-color: #343a40;
    z-index: 1000;
  }

  body {
    padding-top: 70px;
  }
  @media (max-width: 1000px) {
    body {
      padding-top: 150px;
    }
  }
  @media (max-width: 768px) {
    body {
      padding-top: 200px;
    }
  }
</style>
<header class="p-3 text-bg-dark" >
  <div class="container" >
    <div class="d-flex flex-wrap align-items-center justify-content-center justify-content-lg-start" >
      <a href="/home" class="d-flex align-items-center mb-2 mb-lg-0 text-white text-decoration-none">
        <h1 style="margin-right: 20px">OnAvtovokzal</h1>
      </a>

      <ul class="nav col-12 col-lg-auto me-lg-auto mb-2 justify-content-center mb-md-0">
        <li><a href="/home" class="nav-link px-2 text-secondary">Головна</a></li>
        <li><a href="/journeys" class="nav-link px-2 text-white">Розписання всіх автобусів</a></li>
        <li><a href="/buyticket/" class="nav-link px-2 text-white">Купити квиток</a></li>
        <li><a href="/About" class="nav-link px-2 text-white">Про нас</a></li>
      </ul>

      <form class="col-12 col-lg-auto mb-3 mb-lg-0 me-lg-3" role="search" method="POST" action="/getticket">
        <script>
          document.addEventListener('DOMContentLoaded', function() {

```

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижморя О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		105

```

        document.getElementById('search').addEventListener('focus',
function() {
            this.value = "";
        });
    });
    function checkTicketId() {
        var searchInput = document.getElementById('search');
        var ticketId = searchInput.value.trim();
        if (ticketId.length === 8 && !isNaN(ticketId)) {
            searchInput.closest('form').submit();
        }
    }
</script>
<input type="search" id="search" name="ticket_id" class="form-control
form-control-dark text-bg-dark" value="Напишіть код квитка..."
onkeyup="checkTicketId()">
</form>

{% if user is not none %}
<div class="text-end authorizationcompleted" style="margin-right:-50px ">
    <a href="/accountview"> </a>
</div>
{% endif %}
{% if user is none %}
<div class="text-end authorizationnotcompleted">
    <button type="button" class="btn btn-outline-light me-2"
onclick="redirectToLogin()"> Ввійти</button>
    <button type="button" class="btn btn-primary"
onclick="redirectToRegister()">Зареєструватися</button>
</div>
{% endif %}
<script>
    function redirectToLogin() {
        window.location.href = "/loginselect";
    }

    function redirectToRegister() {
        window.location.href = "/register";
    }
</script>
</div>
</div>
</header>
{% block body %}{% endblock %}

<footer class="py-3 my-4 bg-body-secondary">
<ul class="nav justify-content-center border-bottom pb-3 mb-3">
    <li class="nav-item"><a href="/home" class="nav-link px-2 text-body-
secondary">Home</a></li>

```

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижморя О.В.				106
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    <li class="nav-item"><a href="/about" class="nav-link px-2 text-body-
secondary">About</a></li>
  </ul>
  <p class="text-center text-body-secondary ">© 2023-2024 OnAvtovokzal, Inc</p>
</footer>
</body>
</html>

```

baseadmin.html

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="{ url_for('static',filename='css/main.css') }">
  <link
href="https://cdn.jsdelivrivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
T3c6CoIi6uLrA9TneNEoa7RxnatzjcDSCmG1MXxSR1GAsXEV/Dwwykc2MPK8M2HN"
crossorigin="anonymous">
  <script
src="https://cdn.jsdelivrivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js"
integrity="sha384-C6RzsynM9kWDrmNeT87bh950GNyZPhcTNXj1NW7RuBCsyN/o0jlpcV8Qyq46cDfL"
crossorigin="anonymous"></script>
  <title>{% block title %}{% endblock %}</title>

</head>
<body>
  <div class="text-center my-5">
    <h1>Панель адміністрації</h1>
  </div>
  {% block body %}{% endblock %}
</body>
</html>

```

buyticket.html

```

{% extends 'base.html' %}

{% block title %}
  Купити квитки
{% endblock %}

{% block body %}
  {% block form %}
  {% endblock %}
  {% block table %}
  {% endblock %}
  {% block pay %}
  {% endblock %}
{% endblock %}

```

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижморя О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		107

delete.html

```
{% extends 'baseadmin.html' %}
```

```
{% block title %}
```

```
Видалити
```

```
{% endblock %}
```

```
{% block body %}
```

```
<script>
```

```
    document.addEventListener('DOMContentLoaded', function () {
```

```
    let radioButtons =
```

```
document.querySelectorAll('input[name="listGroupCheckableRadios"]');
```

```
    let labels = document.querySelectorAll('.list-group-item');
```

```
    for (let i = 0; i < radioButtons.length; i++) {
```

```
        radioButtons[i].addEventListener('change', function () {
```

```
            for (let j = 0; j < radioButtons.length; j++) {
```

```
                if (radioButtons[j].checked) {
```

```
                    labels[j].classList.add("active");
```

```
                    labels[j].classList.remove("non-active");
```

```
                } else {
```

```
                    labels[j].classList.remove("active");
```

```
                    labels[j].classList.add("non-active");
```

```
            }
```

```
        }
```

```
    });
```

```
}
```

```
});
```

```
</script>
```

```
<form method="post">
```

```
<div class="d-flex flex-column flex-md-row p-4 gap-4 py-md-5 align-items-center justify-content-center">
```

```
    <div class="list-group list-group-checkable d-grid gap-2 border-0">
```

```
        <h2>Виберіть що хочете видалити</h2>
```

```
        <input style="display: none;" class="list-group-item-check pe-none " type="radio" name="listGroupCheckableRadios" id="listGroupCheckableRadios1" value="bus" checked="">
```

```
        <label class="list-group-item rounded-3 py-3 active" for="listGroupCheckableRadios1">
```

```
            Автобус
```

```
        </label>
```

```
        <input style="display: none;" class="list-group-item-check pe-none" type="radio" name="listGroupCheckableRadios" id="listGroupCheckableRadios2" value="journey">
```

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижморя О.В.				108
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        <label class="list-group-item rounded-3 py-3"
for="listGroupCheckableRadios2">
            Подорож
        </label>
        <input style="display: none;" class="list-group-item-check pe-none"
type="radio" name="listGroupCheckableRadios" id="listGroupCheckableRadios3"
value="city">
        <label class="list-group-item rounded-3 py-3"
for="listGroupCheckableRadios3">
            Місто
        </label>
        <input style="display: none;" class="list-group-item-check pe-none"
type="radio" name="listGroupCheckableRadios" id="listGroupCheckableRadios4"
value="ticket">
        <label class="list-group-item rounded-3 py-3"
for="listGroupCheckableRadios4">
            Квиток
        </label>
        <input style="display: none;" class="list-group-item-check pe-none"
type="radio" name="listGroupCheckableRadios" id="listGroupCheckableRadios5"
value="client">
        <label class="list-group-item rounded-3 py-3"
for="listGroupCheckableRadios5">
            Клієнт
        </label>
    </div>
    <button type="submit" class="btn btn-primary">Видалити</button>
    <a href="/adminpanel" class="btn btn-primary">Назад</a>
</div>
</form>

```

```
{% endblock %}
```

deletepost.html

```
{% extends 'baseadmin.html' %}
```

```
{% block title %}
```

Видалити запис

```
{% endblock %}
```

```
{% block body %}
```

```
<script>
```

```
document.addEventListener('DOMContentLoaded', function () {
```

```
const fields = {
```

```
    'id_bus': 'int',
```

```
    'name': 'str',
```

```
    'seats': 'int',
```

```
    'seats_occupied': 'int',
```

```
    'bus_number': 'str',
```

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижморя О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		109

```

        'id_city': 'int',
        'country': 'str',
        'id_client': 'int',
        'phone': 'str',
        'email': 'str',
        'password': 'str',
        'role_client': 'str',
        'id_ticket': 'int',
        'id_journey': 'int',
        'seat': 'int',
        'date_buying': 'datetime',
        'bus_id': 'int',
        'city_start_id': 'int',
        'city_finish_id': 'int',
        'cost': 'int',
        'time_start': 'datetime',
        'time_finish': 'datetime'
    };

    const selectedField = '{{ selectedfield }}';
    const form = document.getElementsByTagName('form')[0];
    console.log("field"+ selectedField);
    let fieldType = fields[selectedField.replace(/["']/g, "")];
    let newInput=document.createElement('input');
    if (fieldType==='int'){
        newInput.type='number';
    }
    else if (fieldType==='str'){
        newInput.type='text';
    }
    else if (fieldType==='datetime'){
        window.location.href="/home";
    }
    newInput.name = 'value';
    newInput.id = 'value';
    let newLabel = document.createElement('label');
    newLabel.for = 'value';
    newLabel.innerHTML = "Значення поля";
    let inp=document.createElement('input');
    inp.type='hidden';
    inp.id='type';
    inp.name='type';
    inp.value=fieldType;
    let h2Element = document.querySelector('h2');
    h2Element.insertAdjacentElement('afterend', newInput);
    h2Element.insertAdjacentElement('afterend', newLabel);
    h2Element.insertAdjacentElement('afterend', inp);

    });

    function redirectToRegister(url) {

```

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижмотря О.В.				110
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        window.location.href = url;
    }
</script>

{% if info is none %}
    <form method="post">
        <h2>Видалення запису</h2>
        <button type="submit">Видалити</button>
    </form>

{% else %}
    <script>
        alert('Не оновлено')
    </script>
{% endif %}
{% endblock %}

```

deletewhatrecord.html

```

{% extends 'baseadmin.html' %}

{% block title %}
    Видалити запис
{% endblock %}

{% block body %}
    <script>
        document.addEventListener('DOMContentLoaded', function () {
            let radioButtons =
document.querySelectorAll('input[name="listGroupCheckableRadios"]');
            let labels = document.querySelectorAll('.list-group-item');

            for (let i = 0; i < radioButtons.length; i++) {
                radioButtons[i].addEventListener('change', function () {
                    for (let j = 0; j < radioButtons.length; j++) {
                        if (radioButtons[j].checked) {
                            labels[j].classList.add("active");
                            labels[j].classList.remove("non-active");
                        } else {
                            labels[j].classList.remove("active");
                            labels[j].classList.add("non-active");
                        }
                    }
                });
            }
        });

        function redirectToNext(event) {
            event.preventDefault()

```

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижмоторя О.В.				111
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        let selectedOption =
document.querySelector('input[name="listGroupCheckableRadios"]:checked').value;
        let clas="{ { clas.__class__.__name__ } }"
        console.log(clas)
        window.location.href = "/deletepost/" + selectedOption+"/"+clas;
    }
</script>
<form method="post">
    <div class="d-flex flex-column flex-md-row p-4 gap-4 py-md-5 align-items-
center justify-content-center">
        <div class="list-group list-group-checkable d-grid gap-2 border-0">
            <h2>За яким полем будемо видаляти? ?</h2>
            {% for i in fieldlist %}
                <input class="list-group-item-check pe-none " type="radio"
name="listGroupCheckableRadios" id="{ { i } }" value="{ { i } }" style="display: none;">
                <label class="list-group-item rounded-3 py-3 active" for="{ { i
}}">{ { i } }</label><br>
            {% endfor %}
            <button type="submit" class="btn btn-primary"
onclick="redirectToNext(event)">Видалити за цим</button>
            <a href="/adminpanel" class="btn btn-primary">Назад</a>
        </div>
    </div>
</form>
{% endblock %}

```

formbuyticket.html

```

{% extends 'buyticket.html' %}
{% block form %}
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Form</title>
</head>
<body>
<script>

    document.addEventListener('DOMContentLoaded', function () {
        document.querySelector('input[name="phone"]').addEventListener('input',
function () {
            phoneValid()
        });
    });

    document.addEventListener('DOMContentLoaded', function () {

```

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижмоторя О.В.				112
Змн.	Арк.	№ докум.	Підпис	Дата		


```

document.querySelector('input[name="finishtime"]').addEventListener('change',
function () {
    timeValid()
})

function timeValid(){
    let timestart = document.querySelector('input[name="starttime"']')
    let timefinish = document.querySelector('input[name="finishtime"']')
    if ( timefinish.value > timestart.value) {
        timefinish.style.border = "solid 2px green"
    }
    else {
        timefinish.style.border = "solid 2px red";
    }
}

function phoneValid(event=""){
    let phone=document.querySelector('input[name="phone"']')
    let phoneNumber = phone.value

    if (phoneNumber.length === 10) {
        phone.style.border = "2px solid green";
        //redirectToSelect()
    }
    else {
        phone.style.border = "2px solid red";
        let labphone = document.getElementById('labphone');
        labphone.innerHTML = "Введіть коректний номер";
        event.preventDefault()
    }
    console.log(phoneNumber)
}
</script>
<script
src="https://maps.googleapis.com/maps/api/js?key=AIzaSyA9h0cqXCKQ9PCEY0wz02r-
n1_3aU_p8Do&libraries=places"></script>
<script>

    let citystart;
    let cityfinish;
    let citylist = [];

    {% for city in cities %}
        citylist.push({
            name: "{{ city.name }}",
            country: "{{ city.country }}"
        });
    {% endfor %}

```

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижморя О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		113

```

console.log(citylist);
document.addEventListener('DOMContentLoaded',function (){

document.querySelector('select[name="citystart"]').addEventListener('change',
function() {
    console.log("strrt")
    setCities()
});

document.querySelector('select[name="cityfinish"]').addEventListener('change',
function() {
    console.log("finish")
    setCities()
});

})

function setCities(){
    citystart=document.querySelector('select[name="citystart"]').value
    cityfinish=document.querySelector('select[name="cityfinish"]').value

    if (citystart!="Відправляємося з" && cityfinish!="Їдем в") {
        console.log("Обидва міста вибрані:", citystart, cityfinish);
        let googlemap = document.querySelector('div[id="googlemap"]')
        googlemap.style.display="block"
        initMap()
    }
}

function initMap() {
    let map = new google.maps.Map(document.getElementById('map'), {
        center: {lat: 37.7749, lng: -122.4194},
        zoom: 13
    });

    var directionsService = new google.maps.DirectionsService();
    var directionsRenderer = new google.maps.DirectionsRenderer({map: map});

    calculateAndDisplayRoute(directionsService, directionsRenderer);
}

function calculateAndDisplayRoute(directionsService, directionsRenderer) {

    let or = citystart+", "+ citylist[citystart]
    let dest = cityfinish+", "+ citylist[cityfinish]
    directionsService.route(
    {

```

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижморя О.В.				114
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        origin: or,
        destination: dest,
        travelMode: 'DRIVING'
    },
    function(response, status) {
        if (status === 'OK') {
            directionsRenderer.setDirections(response);

            var route = response.routes[0];
            var durationInSeconds = 0;

            for (var i = 0; i < route.legs.length; i++) {
                durationInSeconds += route.legs[i].duration.value;
            }

            var hours = Math.floor(durationInSeconds / 3600);
            var minutes = Math.floor((durationInSeconds % 3600) / 60);

            var travelTimeInfo = document.getElementById('travel-time');
            travelTimeInfo.textContent = 'Приблизний час подорожі : ' + hours +
' годин ' + minutes + ' хвилин';
        } else {
            window.alert('Не вдалося розрахувати маршрут через: ' + status);
        }
    }
    );
}
</script>
{% if user is none %}
    <div class="d-flex align-items-center justify-content-center py-4 bg-body-
tertiary">
        <form class="text-center" method="GET">
            <div class="form-floating">
                <input type="text" class="form-control" id="inpname" name="name"
required>
                <label for="inpname">Введіть ім'я клієнта</label>
            </div>
            <div class="form-floating">
                <input type="text" class="form-control" id="inpphone"
name="phone" required>
                <label id="labphone" for="inpname">Введіть телефон
клієнта</label>
            </div>
        </form>
    </div>
{% else %}
    <div class="d-flex align-items-center justify-content-center py-4 bg-body-
tertiary">
        <form class="text-center" method="GET">
            <div class="form-floating">
                <input type="text" class="form-control" id="inpname" name="name"
required>
                <label for="inpname">Введіть ім'я клієнта</label>
            </div>
            <div class="form-floating">
                <input type="text" class="form-control" id="inpphone"
name="phone" required>
                <label id="labphone" for="inpname">Введіть телефон
клієнта</label>
            </div>
        </form>
    </div>
{% endif %}

```

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижморя О.В.				115
Змн.	Арк.	№ докум.	Підпис	Дата		

```

<div class="d-flex align-items-center justify-content-center py-4 bg-
body-tertiary">
  <div class="row text-center">
    <h3>ЗАМОВИТИ КВИТОК</h3>
    <div class="col-md-12 themed-grid-col" style="padding: 5px">
      <div class="row justify-content-center">
        <div class="col-md-6 themed-grid-col">
          <select name="citystart" class="form-select"
required>
            <option>Відправляємося з</option>
            {% for city in cities %}
              <option>{{ city.name }}</option>
            {% endfor %}
          </select>
        </div>
        <div class="col-md-6 themed-grid-col">
          <select name="cityfinish" class="form-select"
required>
            <option>Ідемо в</option>
            {% for city in cities %}
              <option>{{ city.name }}</option>
            {% endfor %}
          </select>
        </div>
      </div>
      <div class="col-md-12 themed-grid-col" style="padding: 5px">
        <div class="row justify-content-center">
          <div class="col-md-2 themed-grid-col" style="margin-
top: 6px">Відправлення від</div>
          <div class="col-md-4 themed-grid-col">
            <input class="form-select" type="datetime-local"
name="starttime" id="startinput" required>
          </div>
          <div class="col-md-2 themed-grid-col" style="margin-
top: 6px">До</div>
          <div class="col-md-4 themed-grid-col">
            <input class="form-select" type="datetime-local"
name="finishtime" id="finishinput" required>
          </div>
        </div>
      </div>
      <div id="googlemap" class="align-items-cente"
style="display: none">
        <div id="map" style="height: 400px;"></div>
        <div id="travel-time"></div>
      </div>
    </div>
  </div>
</div>

```

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижмоторя О.В.				116
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        <button type="submit" class="btn btn-primary"
style="padding-left: 50px;padding-right: 50px"
onclick="phoneValid(event)">Знайти</button>
    </div>

</div>
</div>
</div>
</form>
</div>
</body>
</html>
{% endblock %}

```

home.html

```

{% extends 'base.html' %}

{% block title %}
Головна
{% endblock %}

{% block body %}
    <div class="b-example-divider"></div>
    <div class="b-example-divider"></div>

    <div class="p-5 text-center bg-body-secondary">
        <div class="container py-5">
            <h1 class="text-body-emphasis">Ласкаво просимо на "OnAvtovokzal" – ваш онлайн
автовокзал!</h1>
            <p class="col-lg-8 mx-auto lead">
                Вітаємо вас на найзручнішому та найсучаснішому сервісі для планування вашого
мандрівного дня. "OnAvtovokzal" - це онлайн платформа, яка дозволяє вам легко та
зручно організувати ваші подорожі автобусом.
            </p>
        </div>
    </div>
</div>

    <div class="b-example-divider"></div>
<div class="p-5 text-center bg-body-secondary">
<div class="container px-4 py-5 " id="hanging-icons">
    <h2 class="pb-2 border-bottom">Наші можливості:</h2>
    <div class="row g-4 py-5 row-cols-1 row-cols-lg-4">
        <div class="col d-flex align-items-start">
            <div style="" class="icon-square text-body-emphasis y d-inline-flex align-
items-center justify-content-center fs-4 flex-shrink-0 me-3">
                
            </div>

```

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижморя О.В.				117
Змн.	Арк.	№ докум.	Підпис	Дата		

```

<div>
  <h3 class="fs-3 text-body-emphasis">Розклад автобусів</h3>
  <p>Переглядайте актуальний розклад руху автобусів, обирайте зручний час
відправлення та призначення.</p>
  <a href="/journeys" class="btn btn-primary">
    Переглянути розклад
  </a>
</div>
</div>
<div class="col d-flex align-items-start">
  <div class="icon-square text-body-emphasis d-inline-flex align-items-center
justify-content-center fs-4 flex-shrink-0 me-3">
    
  </div>
  <div>
    <h3 class="fs-3 text-body-emphasis">Пошук маршрутів</h3>
    <p>Знаходьте оптимальні маршрути між містами та вибирайте найзручніші
варіанти подорожі.</p>
    <a href="/formbuyticket" class="btn btn-primary">
      Знайти
    </a>
  </div>
</div>
<div class="col d-flex align-items-start">
  <div class="icon-square text-body-emphasis d-inline-flex align-items-center
justify-content-center fs-4 flex-shrink-0 me-3">
    
  </div>
  <div>
    <h3 class="fs-3 text-body-emphasis">Квитки онлайн</h3>
    <p>Бронюйте та оплачуйте квитки прямо на сайті, уникайте черг та
забезпечте собі місце в потрібному вам автобусі.</p>
    <a href="/formbuyticket" class="btn btn-primary">
      Забронювати
    </a>
  </div>
</div>
<div class="col d-flex align-items-start">
  <div class="icon-square text-body-emphasis d-inline-flex align-items-center
justify-content-center fs-4 flex-shrink-0 me-3">
    
  </div>
  <div>
    <h3 class="fs-3 text-body-emphasis">Інформація про автобуси</h3>
    <p>Детальна інформація про автобуси, включаючи кількість місць, номер
автобуса та інше.</p>
    <a href="/journeys" class="btn btn-primary" style="">

```

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижморя О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		118

```

        Дізнатись інформацію
    </a>
</div>
</div>
</div>
</div>
</div>
{% endblock %}

```

index.html

```

{% extends 'base.html' %}

{% block title %}
Головна
{% endblock %}

{% block body %}

<h1>Головна сторінка</h1>
{% endblock %}

```

journeys.html

```

{% extends 'base.html' %}

{% block title %}
Розклад автобусів
{% endblock %}

{% block body %}
<style>
td{
    justify-content: center;
}
</style>
<!--<script>
document.addEventListener('DOMContentLoaded', function() {
    // Отримуємо всі форми на сторінці
    var forms = document.querySelectorAll('form');

    // Додаємо обробник подій для кожної форми
    forms.forEach(function(form) {
        form.addEventListener('submit', function(event) {
            event.preventDefault();

            var formData = new FormData(form);

            fetch('/selectplace/', {
                method: 'POST',
                body: formData
            })
        });
    });
});

```

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижмоторя О.В.				119
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    })
    .then(response => {
        if (response.ok) {
            return response.text();
        } else {
            console.error('Помилка: ' + response.status);
        }
    })
    .then(html => {
        // Оновлюємо вміст поточної сторінки з отриманим HTML
        document.body.innerHTML = html;
    })
    .catch(error => {
        console.error('Помилка виконання запиту:', error);
    });
});
});
</script>-->
<script>
function submitForm(index) {
    let nameForm="buyTicketForm"+ index.toString()
    //if starttime
    document.getElementById('buyTicketForm').submit();
}

async function redirectToSelect() {
    const itemSort = document.querySelector('select[name="item_sort"]').value;
    const direction = document.querySelector('select[name="direction"]').value;
    const citystart = document.querySelector('select[name="citystart"]').value;
    if (document.querySelector('input[name="cityfinish"]').value){
        const cityfinish =
document.querySelector('select[name="cityfinish"]').value;
    }
    if (document.querySelector('input[name="starttime"]').value){
        const starttime =
document.querySelector('select[name="starttime"]').value;
        let hiddinptime=document.querySelector('input[name="starttime"]')
        hiddinptime.value=starttime
    }
    if (document.querySelector('input[name="finishtime"]').value){
        const finishtime =
document.querySelector('select[name="finishtime"]').value;
        let hiddinptime=document.querySelector('input[name="finishtime"]')
        hiddinptime.value=finishtime
    }
    //alert("POST")
    const formData = new FormData();
    formData.append('item_sort', itemSort);
    formData.append('direction', direction);

```

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижморя О.В.				120
Змн.	Арк.	№ докум.	Підпис	Дата		


```

formData.append('citystart', citystart);
if (document.querySelector('input[name="cityfinish"]').value){
    formData.append('cityfinish', cityfinish);
}
if (document.querySelector('input[name="starttime"]').value){
    formData.append('starttime', starttime);
}
if (document.querySelector('input[name="finishtime"]').value){
    formData.append('finishtime', finishtime);
}
try {
    const response = await fetch('/journeys', {
        method: 'POST',
        body: formData
    });

    if (response.ok) {
        window.location.href='/journeys';
        window.location.reload();
    } else {
        // Обробка помилок
        console.error('Помилка при відправці POST-запиту');
    }
} catch (error) {
    console.error('Помилка при виконанні fetch:', error);
}
}

```

</script>

<div class=" text-center bg-body-secondary" style="margin-top: 25px !important;">

<div class="container py-3">

<h1>Розклад всіх автобусів</h1>

</div>

</div>

<div class="b-example-divider"></div>

<div class="my-5 ">

<form class="form-control bg-body-secondary">

<div class="row text-center">

<h3>Фільтр</h3>

<div class="col-md-11 themed-grid-col" style="padding: 5px">

<div class="row justify-content-center">

<div class="col-md-1 themed-grid-col">

<select name="citystart" class="form-select" required>

<option>Відправляємося з</option>

{% for city in cities %}

<option>{{ city.name }}</option>

{% endfor %}

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижмоторя О.В.				121
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        </select>
    </div>
    <div class="col-md-1 themed-grid-col">
        <select name="cityfinish" class="form-select" required>
            <option>Їдем в</option>
            {% for city in cities %}
                <option>{{ city.name }}</option>
            {% endfor %}
        </select>
    </div>
    <div class="col-md-2 themed-grid-col">
        <select name="item_sort" class="form-select"
required>
            <option>Назва автобусу</option>
            <option>За іменем міста відправлення</option>
            <option>За іменем міста прибуття</option>
            <option>За часом відправлення</option>
            <option>За часом прибуття</option>
        </select>
    </div>
    <div class="col-md-2 themed-grid-col">
        <select name="direction" class="form-select" required>
            <option>По спаданню</option>
            <option>По зростанню</option>
        </select>
    </div>
    <div class="col-md-1 themed-grid-col" >Відправлення
від</div>
    <div class="col-md-2 themed-grid-col">
        <input class="form-select" type="datetime-local"
name="starttime" id="startinput">
    </div>
    <div class="col-md-1 themed-grid-col" style="margin-top:
6px">До</div>
    <div class="col-md-2 themed-grid-col">
        <input class="form-select" type="datetime-local"
name="finishtime" id="finishinput">
    </div>
    </div>
    <div class="col-md-1 themed-grid-col" style="padding: 5px">
        <button class="btn btn-primary" type="submit" id="btn_sort"
onclick="redirectToSelect()">Фільтрувати</button>
    </div>
</div>
</form>
</div>
{% if table is none %}
    <script>
        //alert('1')

```

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижмотря О.В.				122
Змн.	Арк.	№ докум.	Підпис	Дата		

```

</script>
<div class="bd-example-snippet bd-code-snippet">
<div class="bd-example m-0 border-0">
  <table class="table table-striped">
    <thead>
      <tr>
        <th scope="col">#</th>
        <th scope="col">Автобус</th>
        <th scope="col">Місто відправлення</th>
        <th scope="col">Місто прибуття</th>
        <th scope="col">Час відправлення</th>
        <th scope="col">Час прибуття</th>
        <th scope="col">Ціна</th>
        <th scope="col">Замовити</th>
      </tr>
    </thead>

    <tbody>
      {% for journey in journeys %}
      <tr>
        <th scope="row">{{ loop.index }}</th>
        <td>
          {% for bus in buses %}
            {% if journey.bus_id == bus.id_bus %}
              {{ bus.name }}
            {% endif %}
          {% endfor %}
        </td>
        <td>
          {% for city in cities %}
            {% if journey.city_start_id == city.id_city %}
              {{ city.name }}
            {% endif %}
          {% endfor %}
        </td>
        <td>
          {% for city in cities %}
            {% if journey.city_finish_id == city.id_city %}
              {{ city.name }}
            {% endif %}
          {% endfor %}
        </td>
        <td>{{ journey.time_start }}</td>
        <td>{{ journey.time_finish }}</td>
        <td>{{ journey.cost }}</td>
        <td>
          {% if user is not none %}
          <form action="/selectplace/" method="post" >
            <input type="hidden" name="id_journey" value="{{
journey.id_journey }}">

```

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижмотря О.В.				123
Змн.	Арк.	№ докум.	Підпис	Дата		

```

bus in buses %}

        <input type="hidden" name="bus_name" value="{% for
        {% if bus.id_bus == journey.bus_id %}
            {{ bus.name }}

        {% endif %}
        {% endfor %}">
citystart }}">
        <input type="hidden" name="citystart" value="{{
cityfinish }}">
        <input type="hidden" name="cityfinish" value="{{
        <button type="submit" class="btn btn-primary"
>Купити</button>

        </form>

        {% endif %}
        {% if user is none %}
        <a class="btn btn-primary"
href="/loginselect">Ввійти</a>
        {% endif %}

```

```

        </td>
    </tr>
    {% endfor %}
</tbody>

```

```

</table>
</div>
</div>
{% endif %}
{% if table is not none %}

```

```

<div class="bd-example-snippet bd-code-snippet">
<div class="bd-example m-0 border-0">
    <table class="table table-striped">
        <thead>
            <tr>
                <th scope="col">#</th>
                <th scope="col">Автобус</th>
                <th scope="col">Місто відправлення</th>
                <th scope="col">Місто прибуття</th>
                <th scope="col">Час відправлення</th>
                <th scope="col">Час прибуття</th>
                <th scope="col">Ціна</th>
                <th scope="col">Замовити</th>
            </tr>
        </thead>

```

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижморя О.В.				124
Змн.	Арк.	№ докум.	Підпис	Дата		

```
<tbody>
  {% for tab in table %}
  <tr>
    <th scope="row">{{ loop.index }}</th>
    <td>
      {{ tab.bus_name }}
    </td>
    <td>
      {{ tab.start_city }}
    </td>
    <td>
      {{ tab.finish_city }}
    </td>
    <td>{{ tab.time_start }}</td>
    <td>{{ tab.time_finish }}</td>
    <td>{{ tab.cost }}</td>
    <td>
      {% if user is not none %}
      <form action="/selectplace/" method="post">
        <input type="hidden" name="id_journey" value="{{
tab.id_journey }}">
        <input type="hidden" name="bus_name" value="{{
tab.bus_name }}">
        <input type="hidden" name="start_city" value="{{
tab.start_city }}">
        <input type="hidden" name="finish_city"
value="{{ tab.finish_city }}">
        <button type="submit" class="btn btn-primary"
>Купити</button>
      </form>
      {% endif %}
      {% if user is none %}
      <a class="btn btn-primary"
href="/loginselect">Ввійти</a>
      {% endif %}
    </td>
  </tr>
  {% endfor %}
</tbody>

</table>
</div>
</div>
{% endif %}
{% endblock %}
```

login.html

```
{% extends 'base.html' %}

{% block title %}
Ввійти в акаунт
{% endblock %}

{% block body %}
    {% if info=="Не вірний пароль" %}
        <script>
            alert("Не вірний пароль")
        </script>
    {% endif %}
<style>
    div.form-floating>input{
        margin-bottom: 5px;

    }
    form.my>button {
        margin-top: 10px;
    }
</style>
<div class="d-flex align-items-center justify-content-center py-4 bg-body-tertiary">
    <form class="text-center my" method="post">
        <h1 class="h3 mb-3 fw-normal">Вхід</h1>
        {% if data == 'email' %}
        <div class="form-floating">
            <input type="email" class="form-control" id="floatingInputEmail"
name="floatingInputEmail" placeholder="name@example.com">
            <label for="floatingInput">Введіть email</label>
        </div>
        {% endif %}
        {% if data == 'phone' %}
        <div class="form-floating">
            <input type="text" class="form-control" id="floatingInputPhone"
name="floatingInputPhone" placeholder="Phone">
            <label for="floatingInputPhone">Введіть телефон</label>
        </div>
        {% endif %}
        <div class="form-floating">
            <input type="password" class="form-control" id="floatingPassword"
name="floatingPassword" placeholder="Password">
            <label id="passwordlabel" for="floatingPassword">Введіть пароль</label>
        </div>

        <button class="btn btn-primary w-100 py-2" type="submit">Ввійти</button>
        <p class="mt-5 mb-3 text-body-secondary">©2024</p>
    </form>
```

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижмотря О.В.				126
Змн.	Арк.	№ докум.	Підпис	Дата		

</div>

{% endblock %}

loginselect.html

{% extends 'base.html' %}

{% block title %}

Виберіть спосіб входу

{% endblock %}

{% block body %}

<script>

function redirectToLogin() {

let selectedOption =

document.querySelector('input[name="listGroupCheckableRadios"]:checked').value;

window.location.href = "/login/" + selectedOption;

}

document.addEventListener('DOMContentLoaded', function () {

let radioButtons =

document.querySelectorAll('input[name="listGroupCheckableRadios"]');

let labels = document.querySelectorAll('.list-group-item');

for (let i = 0; i < radioButtons.length; i++) {

radioButtons[i].addEventListener('change', function () {

for (let j = 0; j < radioButtons.length; j++) {

if (radioButtons[j].checked) {

labels[j].classList.add("active");

labels[j].classList.remove("non-active");

} else {

labels[j].classList.remove("active");

labels[j].classList.add("non-active");

}

}

});

}

});

</script>

<div class="d-flex flex-column flex-md-row p-4 gap-4 py-md-5 align-items-center justify-content-center">

<div class="list-group list-group-checkable d-grid gap-2 border-0">

<input style="display: none;" class="list-group-item-check pe-none " type="radio" name="listGroupCheckableRadios" id="listGroupCheckableRadios1" value="email" checked="">

<label class="list-group-item rounded-3 py-3 active" for="listGroupCheckableRadios1">

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижевотря О.В.				127
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        Через email
    </label>

    <input style="display: none;" class="list-group-item-check pe-none"
type="radio" name="listGroupCheckableRadios" id="listGroupCheckableRadios2"
value="phone">
    <label class="list-group-item rounded-3 py-3"
for="listGroupCheckableRadios2">
        Через телефон
    </label>
</div>

<button type="button" class="btn btn-primary"
onclick="redirectToLogin()">Ввійти</button>
</div>

{% endblock %}

```

pay.html

```

{% extends "base.html" %}
{% block title %}
    Оплата
{% endblock %}
{% block body %}
    <style>
        div.main{
            display: inline-block;
            width: 100%;
        }
    </style>
    <div class="main text-center" style="margin-top: 100px">
        <h2>Інформація</h2>
        <div class="bd-example-snippet bd-code-snippet">
            <div class="bd-example m-0 border-0" style="margin-top: 50px !important;">
                <table class="table table-striped">
                    <thead>
                        <tr>
                            <th scope="col">Місце</th>
                            <th scope="col">Місто відправлення</th>
                            <th scope="col">Місто прибуття</th>
                            <th scope="col">Час відправлення</th>
                            <th scope="col">Час прибуття</th>
                            <th scope="col">Номер автобуса</th>
                            <th scope="col">Автобус</th>
                            <th scope="col">Ціна</th>
                        </tr>
                    </thead>
                    <tbody>
                        <tr>

```

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижморя О.В.				128
Змн.	Арк.	№ докум.	Підпис	Дата		


```

        <td>{{ seat }}</td>
        <td>{{ start_city.name }}</td>
        <td>{{ finish_city.name }}</td>
        <td>{{ journey.time_start }}</td>
        <td>{{ journey.time_finish }}</td>
        <td>{{ bus.bus_number }}</td>
        <td>{{ bus.name }}</td>
        <td>{{ journey.cost }}</td>
    </tr>
</tbody>
</table>
<form method="post" action="/succesorder/">
    {% if non_auth_user is not none %}
        <input type="hidden" name="phone" value="{{ non_auth_user.phone
    }}">
        <input type="hidden" name="name" value="{{ non_auth_user.name
    }}">
        {% endif %}
        <input type="hidden" name="id_journey" value="{{
journey.id_journey }}">
        <input type="hidden" name="seat" value="{{ seat }}">
        <button type="submit" class="btn btn-primary"> Купити </button>
    </form>

</div>
</div>
</div>
{% endblock %}

```

register.html

```
{% extends 'base.html' %}
```

```

{% block title %}
Зареєструватись
{% endblock %}

```

```

{% block body %}
    {% if info is not none %}
        <script>
            let info = {{ info|tojson|safe }}
            alert(info)
        </script>
    {% endif %}
<style>
    div.form-floating>input{
        margin-bottom: 5px;
    }

```

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижмоторя О.В.				129
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    }
    form.my>button {
        margin-top: 10px;
    }
</style>
<div class="d-flex align-items-center justify-content-center py-4 bg-body-tertiary">
    <form class="text-center my" method="post">
        <h1 class="h3 mb-3 fw-normal">Реєстрація</h1>
        <div class="form-floating">
            <input type="text" class="form-control" id="floatingInputName"
name="floatingInputName" placeholder="Ім'я" required>
            <label for="floatingInputPhone">Ім'я та прізвище</label>
        </div>
        <div class="form-floating">
            <input type="email" class="form-control" id="floatingInputEmail"
name="floatingInputEmail" placeholder="name@example.com" required>
            <label for="floatingInput">Введіть email</label>
        </div>
        <div class="form-floating">
            <input type="text" class="form-control" id="floatingInputPhone"
name="floatingInputPhone" placeholder="Phone" required>
            <label id="labphone" for="floatingInputPhone">Введіть телефон</label>
        </div>
        <div class="form-floating">
            <input type="password" class="form-control" id="floatingInputPassOne"
name="floatingInputPassOne" placeholder="Password" required>
            <label for="floatingPassword">Пароль</label>
        </div>
        <div class="form-floating">
            <input type="password" class="form-control" id="floatingInputPassTwo"
name="floatingInputPassTwo" placeholder="Password" required>
            <label id="labtwo" for="floatingPassword">Повторіть пароль</label>
        </div>
<script>
    let username = "{{ user.name }}";
    console.log("NAME "+username)
    document.addEventListener('DOMContentLoaded', function () {
        document.getElementById('floatingInputPhone').addEventListener('blur', function
() {
            let labphone = document.getElementById('labphone');
            let phoneNumber = this.value.replace(/\D/g, '');

            if (phoneNumber.length === 10) {
                this.style.border = "2px solid green";
                labphone.innerHTML = "Номер телефону вірний";
                //this.value = "+" + phoneNumber;
            } else {
                this.style.border = "2px solid red";
                labphone.innerHTML = "Введіть вірний номер телефону";
            }
        });
    });

```

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижморя О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		130

```

    }
  });
});

document.addEventListener('DOMContentLoaded', function () {
  document.getElementById('floatingInputPhone').addEventListener('change',
function () {

    if (this.value.length<=11||this.value.match(/[a-zA-ZА-Яа-я]/)) {
      let labphone = document.getElementById('labphone');
      this.style.border = "2px solid red";
      labphone.innerHTML = "Введіть вірний номер";
    }
  });
});

function intersting(event) {
  let pass2 = document.getElementById('floatingInputPassTwo');
  let pass1 = document.getElementById('floatingInputPassOne');
  let labeltwo = document.getElementById('labtwo');
  if (pass1.value !== pass2.value && pass1.value !== "" && pass2.value !== "") {
    pass2.style.border = "2px solid red";
    labeltwo.innerHTML = "Введіть схожий пароль";
    event.preventDefault();
  }
  else{
    if (pass1.value !== "" && pass2.value !== "") {

    }

  }
}
</script>
<button class="btn btn-primary w-100 py-2" type="submit" id="buttonsumb"
onclick="intersting(event)">Зареєструватись</button>
<p class="mt-5 mb-3 text-body-secondary">©2024</p>

</form>
</div>

{% endblock %}

```

selectplace.html

```

{% extends "base.html" %}
{% block title %}
  Вибрати місце
{% endblock %}
{% block body %}
  <style>
    .maindiv {

```

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижемотря О.В.				131
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        display: inline-block;
        margin-top: 100px;
        width: 100%;
        background-color: white;
    }

    .bus {
        margin-top: 50px;
        margin-bottom: 50px;
        margin-left: 30%;
        margin-right: 30%;
        background-color: darkgray;
        border: 3px solid grey;
        border-radius: 20px;
        padding: 50px;
        padding-top: 100px;
        display: flex;
        flex-wrap: wrap;
        justify-content: space-around;
    }

    .bus>div {
        margin: 20px;
        width: 100px;
        height: 100px;
        background-color: #0D6EFD;
        color: white;
        padding: 20px;
        display: flex;
        justify-content: center;
        align-items: center;
        border-radius: 20px;
        cursor: pointer;
    }

    .bus>div:nth-child(3n) {
        margin-left: 70px;
    }

    .bus>div:hover {
        border: 3px solid #0a58ca;
        box-shadow: 5px 5px #0D6EFD;
        background-color: #0a58ca;
    }

    .doesnt-select {
        background-color: grey !important;
        color: white !important;
    }
}
</style>

```

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижмотря О.В.				132
Змн.	Арк.	№ докум.	Підпис	Дата		

```

<script>
function validation(event) {
var seatInput = document.querySelector('input[name="seat"]');
var seatValue = seatInput.value.trim();

if (seatValue === "") {
    event.preventDefault();
}
}
</script>
<div class="maindiv text-center">
    <h2>Вибери́ть квиток</h2>
    <div><span style="color: grey">{{ countstickets }}/{{ bus.seats
}}</span></div>
    <div class="bus">
        {% set indexticket = 0 %}
        {% for seat in range(bus.seats) %}
            {% set index = loop.index %}
            {% if greyseats is not none %}
                {% if index in greyseats %}
                    <div class="doesnt-select" data-seat="{{ index }}">{{ index
}}</div>

                    {% set indexticket = indexticket + 1 %}
                {% else %}
                    <div class="selectable-seat" data-seat="{{ index }}"
onclick="selectDiv(this)">{{ index }}</div>
                {% endif %}
            {% else %}
                <div class="selectable-seat" data-seat="{{ index }}"
onclick="selectDiv(this)">{{ index }}</div>
                {% endif %}
            {% endfor %}
        </div>
        <form method="POST" action="/pay/">
            {% if non_auth_user is not none %}
                <input type="hidden" name="phone" value="{{ non_auth_user.phone }}">
                <input type="hidden" name="name" value="{{ non_auth_user.name }}">
            {% endif %}
            <input type="hidden" name="id_journey" value="{{ journey.id_journey }}">
            <input type="hidden" name="seat" value="">
            <h4 id="labelseat">Ви вибрали місце : </h4>
            <button class="btn btn-primary" name="butsub"
onclick="validation(event)">Перейти до оплати</button>
        </form>
    </div>
{% endblock %}

```

succesorder.html

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижморя О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		133

```

{% extends "base.html" %}
{% block title %}
    {% if title is not none %}
        Succes
    {% else %}
        Failed
    {% endif %}
{% endblock %}
{% block body %}
    <style>
    #divcent {
        margin-top: -50px;
        margin-bottom: 50px;
        margin-left: 30%;
        margin-right: 30%;
        background-color: #E9ECEF;
        border: 3px solid #E9ECEF;
        border-radius: 20px;
        padding: 50px;
        padding-top: 100px;
        display: flex;
        flex-wrap: wrap;
        justify-content: space-around;
    }
    #divcent>p{
        width: 100%;
    }
    </style>
    <script>
    function copyText() {
        var text = document.getElementById("textToCopy").innerText;
        navigator.clipboard.writeText(text)
            .then(function() {
                alert("Текст успішно скопійовано!");
            })
            .catch(function(error) {
                console.error("Не вдалося скопіювати текст: ", error);
            });
    }
    </script>

    {% if title is not none %}
        <div id="divcent">
            <div class="text-center" ><h3>Куплений квиток</h3></div>
            <p >Номер квитка <span id="textToCopy" >{{ ticket.code
}}</span> <a onclick="copyText()"><span style="font-size: 12px; color:
grey;cursor:pointer ; ">Копіювати</span></a></p>
            <p>Ваше ім'я
                {% if non_auth_user is none %}
                    {{ user.name }}

```

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижморя О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		134

```

        {% else %}
            {{ non_auth_user.name }}
        {% endif %}
    </p>
    <p>Місце : {{ seat }}</p>
    <p>Місто відправлення : {{ start_city.name }}</p>
    <p>Місто прибуття : {{ finish_city.name }}</p>
    <p>Час відправлення : {{ journey.time_start }}</p>
    <p>Час прибуття : {{ journey.time_finish }}</p>
    <p>Автобус : {{ bus.name }}</p>
    <p>Номер автобуса : {{ bus.bus_number }}</p>
</div>
{% else %}
    <div class="text-center" style="margin-top: 100px">
        <h2>Failed payment</h2>

        {% endif %}
    <div class="text-center">
        <a class="btn btn-primary" href="/formbuyticket">Вибрати новий шлях</a>
        <a class="btn btn-primary" href="/home">На головну</a>
    </div>
</div>
{% endblock %}

```

tablejourneys.html

```

{% extends 'buyticket.html' %}

{% block form %}
<body>
    <script>
    document.addEventListener('DOMContentLoaded', function() {
        var forms = document.querySelectorAll('form');

        forms.forEach(function(form) {
            form.addEventListener('submit', function(event) {
                event.preventDefault();

                var formData = new FormData(form);

                fetch('/selectplace/', {
                    method: 'POST',
                    body: formData
                })
                .then(response => {
                    if (response.ok) {
                        return response.text();
                    } else {
                        console.error('Помилка: ' + response.status);
                    }
                })
            });
        });
    });
    </script>
</body>

```

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижмотря О.В.				135
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    })
    .then(html => {
        document.body.innerHTML = html;
    })
    .catch(error => {
        console.error('Помилка виконання запиту:', error);
    });
});
});
</script>
{% if table is not none %}
<div class="bd-example-snippet bd-code-snippet">
<div class="bd-example m-0 border-0" style="margin-top: 50px !important;">
<table class="table table-striped">
<thead>
<tr>
<th scope="col">#</th>
<th scope="col">Автобус</th>
<th scope="col">Місто відправлення</th>
<th scope="col">Місто прибуття</th>
<th scope="col">Час відправлення</th>
<th scope="col">Час прибуття</th>
<th scope="col">Ціна</th>
<th scope="col">Замовити</th>
</tr>
</thead>

<tbody>

{% for row in table %}
<tr>
<td>
<form method="post">
<th scope="row">{{ loop.index }}</th>
<td>
{{ row.bus_name }}
</td>
<td>
{{ row.start_city }}
</td>
<td>
{{ row.finish_city }}
</td>
<td>
{{ row.time_start }}
</td>
<td>
{{ row.time_finish }}
</td>
<td>
</td>

```



```

        {{ row.cost }}
    </td>
    <td>
        {% if non_auth_user is not none %}
        <input type="hidden" name="phone" value="{{
non_auth_user.phone }}">
        <input type="hidden" name="name" value="{{
non_auth_user.name }}">
        {% else %}
        <input type="hidden" name="phone" value="">
        <input type="hidden" name="name" value="">
        {% endif %}
        <input type="hidden" name="id_journey" value="{{
row.id_journey }}">
        <input type="hidden" name="bus_name" value="{{
row.bus_name }}">
        <input type="hidden" name="start_city" value="{{
row.start_city }}">
        <input type="hidden" name="finish_city" value="{{
row.finish_city }}">
        <input type="hidden" name="time_start" value="{{
row.time_start }}">
        <input type="hidden" name="time_finish" value="{{
row.time_finish }}">
        <input type="hidden" name="cost" value="{{ row.cost
}}">
        <button type="submit" class="btn btn-primary"
>Купити</button>
    </td>
</form>
</tr>
{% endfor %}

</tbody>
</table>
</div>
</div>
{% endif %}
{% endblock %}

```

update.html

```

{% extends 'baseadmin.html' %}

{% block title %}
Update
{% endblock %}

```

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижмотря О.В.				137
Змн.	Арк.	№ докум.	Підпис	Дата		

```
{% block body %}
```

```
<script>
    document.addEventListener('DOMContentLoaded', function () {
    let radioButtons =
document.querySelectorAll('input[name="listGroupCheckableRadios"]');
    let labels = document.querySelectorAll('.list-group-item');

    for (let i = 0; i < radioButtons.length; i++) {
        radioButtons[i].addEventListener('change', function () {
            for (let j = 0; j < radioButtons.length; j++) {
                if (radioButtons[j].checked) {
                    labels[j].classList.add("active");
                    labels[j].classList.remove("non-active");
                } else {
                    labels[j].classList.remove("active");
                    labels[j].classList.add("non-active");
                }
            }
        });
    }
});
</script>
```

```
<form method="post">
<div class="d-flex flex-column flex-md-row p-4 gap-4 py-md-5 align-items-center
justify-content-center">
    <div class="list-group list-group-checkable d-grid gap-2 border-0">
        <h2>Виберіть що хочете оновити</h2>
        <input style="display: none;" class="list-group-item-check pe-none "
type="radio" name="listGroupCheckableRadios" id="listGroupCheckableRadios1"
value="bus" checked="">
        <label class="list-group-item rounded-3 py-3 active"
for="listGroupCheckableRadios1">
            Автобус
        </label>
        <input style="display: none;" class="list-group-item-check pe-none"
type="radio" name="listGroupCheckableRadios" id="listGroupCheckableRadios2"
value="journey">
        <label class="list-group-item rounded-3 py-3"
for="listGroupCheckableRadios2">
            Подорож
        </label>
        <input style="display: none;" class="list-group-item-check pe-none"
type="radio" name="listGroupCheckableRadios" id="listGroupCheckableRadios3"
value="city">
        <label class="list-group-item rounded-3 py-3"
for="listGroupCheckableRadios3">
            Місто
        </label>
```

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижмоторя О.В.				138
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        <input style="display: none;" class="list-group-item-check pe-none"
type="radio" name="listGroupCheckableRadios" id="listGroupCheckableRadios4"
value="ticket">
        <label class="list-group-item rounded-3 py-3"
for="listGroupCheckableRadios4">
            Квиток
        </label>
    </div>
    <button type="submit" class="btn btn-primary">Оновити</button>
    <a href="/adminpanel" class="btn btn-primary">Назад</a>
</div>
</form>

```

```
{% endblock %}
```

updatepost.html

```
{% extends 'baseadmin.html' %}
```

```
{% block title %}
```

Оновити запис

```
{% endblock %}
```

```
{% block body %}
```

```
<script>
```

```
document.addEventListener('DOMContentLoaded', function () {
```

```
const fields = {
```

```
    'id_bus': 'int',
```

```
    'name': 'str',
```

```
    'seats': 'int',
```

```
    'seats_occupied': 'int',
```

```
    'bus_number': 'str',
```

```
    'id_city': 'int',
```

```
    'country': 'str',
```

```
    'id_client': 'int',
```

```
    'phone': 'str',
```

```
    'email': 'str',
```

```
    'password': 'str',
```

```
    'role_client': 'str',
```

```
    'id_ticket': 'int',
```

```
    'id_journey': 'int',
```

```
    'seat': 'int',
```

```
    'date_buying': 'datetime',
```

```
    'bus_id': 'int',
```

```
    'city_start_id': 'int',
```

```
    'city_finish_id': 'int',
```

```
    'cost': 'int',
```

```
    'time_start': 'datetime',
```

```
    'time_finish': 'datetime'
```

```
};
```

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижмотря О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		139

```

const selectedField = '{{ selectedfield }}';
const form = document.getElementsByTagName('form')[0];
console.log("field"+ selectedField);
let fieldType = fields[selectedField.replace(/['"]/g, "")];
let newInput=document.createElement('input');
if (fieldType==='int'){
    newInput.type='number';
}
else if (fieldType==='str'){
    newInput.type='text';
}
else if (fieldType==='datetime'){
    newInput.type='datetime-local';
}
newInput.name = 'value';
newInput.id = 'value';
let newLabel = document.createElement('label');
newLabel.for = 'value';
newLabel.innerHTML = "Значення поля";
let inp=document.createElement('input');
inp.type='hidden';
inp.id='type';
inp.name='type';
inp.value=fieldType;
let h2Element = document.querySelector('h2');
h2Element.insertAdjacentElement('afterend', newInput);
h2Element.insertAdjacentElement('afterend', newLabel);
h2Element.insertAdjacentElement('afterend', inp);

});

function redirectToRegister(url) {
    window.location.href = url;
}
</script>

{% if info is none %}
    <form method="post">
        <h2>Оновлення запису</h2>
        <label for="id">Введіть id запису</label>
        <input type="text" name="id" id="id" required>
        <button type="submit">Оновити</button>
    </form>

{% else %}
    <script>
        alert('Не оновлено')
    </script>
{% endif %}
{% endblock %}

```

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижморя О.В.				140
Змн.	Арк.	№ докум.	Підпис	Дата		

updatewhatrecord.html

```
{% extends 'baseadmin.html' %}

{% block title %}
    ОНОВИТИ ЗАПИС
{% endblock %}

{% block body %}
    <script>
        document.addEventListener('DOMContentLoaded', function () {
            let radioButtons =
document.querySelectorAll('input[name="listGroupCheckableRadios"]');
            let labels = document.querySelectorAll('.list-group-item');

            for (let i = 0; i < radioButtons.length; i++) {
                radioButtons[i].addEventListener('change', function () {
                    for (let j = 0; j < radioButtons.length; j++) {
                        if (radioButtons[j].checked) {
                            labels[j].classList.add("active");
                            labels[j].classList.remove("non-active");
                        } else {
                            labels[j].classList.remove("active");
                            labels[j].classList.add("non-active");
                        }
                    }
                });
            }
        });

        function redirectToNext(event) {
            event.preventDefault()
            let selectedOption =
document.querySelector('input[name="listGroupCheckableRadios"]:checked').value;
            let clas="{{ clas.__class__.__name__ }}"
            console.log(clas)
            window.location.href = "/updatepost/" + selectedOption+"/"+clas;
        }
    </script>
    <form method="post">
        <div class="d-flex flex-column flex-md-row p-4 gap-4 py-md-5 align-items-
center justify-content-center">
            <div class="list-group list-group-checkable d-grid gap-2 border-0">
                <h2>Що оновлюємо ?</h2>
                {% for i in fieldlist %}
                    <input class="list-group-item-check pe-none " type="radio"
name="listGroupCheckableRadios" id="{{ i }}" value="{{ i }}" style="display: none;">
```

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижевотря О.В.				141
Змн.	Арк.	№ докум.	Підпис	Дата		

```

                <label class="list-group-item rounded-3 py-3 active" for="{{ i
}}">{{ i }}</label><br>
                {% endfor %}
                <button type="submit" class="btn btn-primary"
onclick="redirectToNext(event)">Оновити це</button>
                <a href="/adminpanel" class="btn btn-primary">Назад</a>
            </div>
        </div>
    </form>
{% endblock %}

```

usertickets.html

```

{% extends 'base.html' %}

{% block title %}
Мої квитки
{% endblock %}

{% block body %}
<style>
    #flexmain {
        margin-top: 100px;
        display: flex;
        flex-wrap: wrap;
        justify-content: flex-start;
    }

    #flexmain>div {
        background-color: #E9ECEF;
        border-radius: 25px;
        box-shadow: 5px 5px black;
        padding: 35px;
        margin-left: 20px;
        margin-top: 20px;
    }

    #flexmain>div:hover {
        background-color: #cfd0d1;
        border-radius: 25px;
        box-shadow: 6px 5px #0D6EFD;
    }

    #flexmain>div>* {
        margin-top: 10px;
    }
</style>
{% if recenttickets is not none %}

```

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижмоторя О.В.				142
Змн.	Арк.	№ докум.	Підпис	Дата		

```

<div id="flexmain">
  {% for cardticket in recenttickets %}
    <div>
      <p>Код поїздки</p><p>{{ cardticket.code }}</p>
      <p>Напрямок</p><p><strong>{{ cardticket.name }}</strong></p>
      <p>Час поїздки</p><p><strong>{{ cardticket.time }}</strong></p>
      <p>Вартість</p><p><strong>{{ cardticket.cost }}</strong></p>

      <form action="/getticket" method="post">
        <input type="hidden" name="ticket_id" value="{{ cardticket.code
    }}}">

        <button type="submit" class="btn btn-primary">Дивитись
деталі</button>

      </form>
    </div>
  {% endfor %}
</div>
{% endif %}
{% if oldtickets is not none %}
<div id="flexmain">
  {% for cardticket in oldtickets %}
    <div style="background-color: #212529;color: white">
      <p style="color: red;text-shadow: 1px 1px 1px white;">Дія білету
завершилась</p>
      <p>Код поїздки</p><p>{{ cardticket.code }}</p>
      <p>Напрямок</p><p><strong>{{ cardticket.name }}</strong></p>
      <p>Час</p><p><strong>{{ cardticket.time }}</strong></p>
      <p>Вартість</p><p><strong>{{ cardticket.cost }}</strong></p>
    </div>
  {% endfor %}
</div>
  {% else %}
    <div style="margin-top: 100px" class="text-center">
      <h3>Квитків не знайдено</h3><br>
    </div>
  {% endif %}

{% endblock %}

```

db_avtovokzal_create_script.sql

```

--create database avtovokzalf;

use avtovokzalf;

create table buses (
  id_bus int not null identity(1,1) PRIMARY KEY,
  name nvarchar(50) not null,
  seats int not null,
  bus_number nvarchar(8) check (bus_number like '[A-Я][A-Я][0-9][0-9][0-9][0-9][A-
Я][A-Я]') unique

```

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижморя О.В.				143
Змн.	Арк.	№ докум.	Підпис	Дата		

```

);

create table cities (
    id_city int not null identity(1,1) PRIMARY KEY,
    name nvarchar(50) not null,
    country nvarchar(50) null
);

create table clients (
    id_client int not null identity(1,1) PRIMARY KEY,
    name nvarchar(50) not null,
    phone nvarchar(11) check (phone like '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]') not null unique default '',
    email nvarchar(50) not null unique default '' check (email like '%@%.%'),
    password nvarchar(50) not null,
    role_client nvarchar(50) not null default 'user'
);

CREATE TABLE non_authorized_users(
    id_user int PRIMARY KEY not null,
    name nvarchar(20) not null,
    phone nvarchar(11) not null unique check (phone like '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]') default ''
);

CREATE TABLE timetable (
    id_journey int not null identity(1,1) PRIMARY KEY,
    bus_id int null,
    city_start_id int not null,
    city_finish_id int not null,
    cost float not null,
    time_start datetime default GETDATE() not null,
    time_finish datetime default GETDATE() not null,
    seats_occupied int not null default 0,
    is_active bit null default 1,
    FOREIGN KEY (bus_id) REFERENCES buses(id_bus),
    FOREIGN KEY (city_start_id) REFERENCES cities(id_city),
    FOREIGN KEY (city_finish_id) REFERENCES cities(id_city)
);

create table orders (
    id_order INT PRIMARY KEY IDENTITY(1,1),
    client_id INT NULL,
    non_authorized_users_id INT NULL,
    date_buying DATETIME DEFAULT GETDATE(),
    cost float null,
    whether_paid bit null,
    FOREIGN KEY (client_id) REFERENCES clients(id_client),
    FOREIGN KEY (non_authorized_users_id) REFERENCES non_authorized_users(id_user)
);

```

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижмотря О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		144


```

create table tickets(
    id_ticket int not null identity(1,1) PRIMARY KEY,
    bus_id int not null,
    seat int not null,
    date_buying datetime default GETDATE() null,
    journey_id int null,
    non_authorized_users_id int null,
    client_id int not null,
    order_id int not null,
    code int null unique,
    FOREIGN KEY (bus_id) REFERENCES buses(id_bus),
    FOREIGN KEY (client_id) REFERENCES clients(id_client),
    FOREIGN KEY (journey_id) REFERENCES timetable(id_journey),
    FOREIGN KEY (order_id) REFERENCES orders(id_order),
    FOREIGN KEY (non_authorized_users_id) REFERENCES non_authorized_users(id_user)
);

```

Procedures.sql

```

create procedure deleteClient
    @id int
AS
BEGIN
    delete from clients
    where id_client=@id
END;

```

```

create procedure deleteBus
    @id int
AS
BEGIN
    delete from buses
    where id_bus=@id
END;

```

```

create procedure deleteCity
    @id int
AS
BEGIN
    delete from cities
    where id_city=@id
END

```

```

create procedure deleteNonAthUser
    @id int
AS
BEGIN
    delete from non_authorized_users
    where id_user=@id
END

```

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижмотря О.В.				145
Змн.	Арк.	№ докум.	Підпис	Дата		

```

create procedure deleteOrder
    @id int
AS
BEGIN
    delete from orders
    where id_order=@id
END

```

```

create procedure deleteTicket
    @id int
AS
BEGIN
    delete from tickets
    where id_ticket=@id
END

```

```

create procedure deleteTimetable
    @id int
AS
BEGIN
    delete from timetable
    where id_journey=@id
END

```

```
--DROP PROCEDURE IF EXISTS get_income_per_day
```

```

create procedure get_income_per_day
as
begin

```

```

    DECLARE @time_now datetime,
            @time_start_day datetime,
            @time_finish_day datetime;

```

```

    SET @time_now = GETDATE();
    SET @time_start_day = CONVERT(datetime, CONVERT(date, @time_now));
    SET @time_finish_day = DATEADD(second, -1, DATEADD(day, 1,
CONVERT(datetime, CONVERT(date, @time_now))));
    SELECT @time_now AS time_now, @time_start_day AS time_start_day,
@time_finish_day AS time_finish_day;

```

```

        select Sum(orders.cost) as income_per_day from orders where date_buying
between @time_start_day and @time_finish_day and orders.whether_paid=1
end

```

```
exec get_income_per_day
```

```
--DROP PROCEDURE IF EXISTS get_income_per_month
```

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижмотря О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		146

```

CREATE PROCEDURE get_income_per_month
AS
BEGIN
    DECLARE @time_now datetime,
            @time_start_month datetime,
            @time_finish_month datetime;

    SET @time_now = GETDATE();
    SET @time_start_month = DATEADD(month, DATEDIFF(month, 0, @time_now), 0);
    SET @time_finish_month = DATEADD(second, -1, DATEADD(month, 1,
@time_start_month));

    SELECT @time_now AS time_now, @time_start_month AS time_start_month,
@time_finish_month AS time_finish_month;

    SELECT SUM(orders.cost) as income_per_month
    FROM orders
    WHERE date_buying BETWEEN @time_start_month AND @time_finish_month
    AND orders.whether_paid = 1;
END;

```

```
exec get_income_per_month
```

```
--DROP PROCEDURE IF EXISTS get_income_per_year
```

```

CREATE PROCEDURE get_income_per_year
AS
BEGIN
    DECLARE @time_now datetime,
            @time_start_year datetime,
            @time_finish_year datetime;

    SET @time_now = GETDATE();
    SET @time_start_year = DATEFROMPARTS(YEAR(@time_now), 1, 1); -- Початок
поточного року
    SET @time_finish_year = DATEADD(second, -1, DATEADD(year, 1,
@time_start_year)); -- Кінець поточного року

    SELECT @time_now AS time_now, @time_start_year AS time_start_year,
@time_finish_year AS time_finish_year;

    SELECT SUM(orders.cost) as income_per_year
    FROM orders
    WHERE date_buying BETWEEN @time_start_year AND @time_finish_year
    AND orders.whether_paid = 1;
END;

```

```
exec get_income_per_year
```

```
--DROP PROCEDURE IF EXISTS get_all_tickets_per_day
```

```

CREATE PROCEDURE get_all_tickets_per_day
AS

```

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижмотря О.В.				147
Змн.	Арк.	№ докум.	Підпис	Дата		

```

BEGIN
    DECLARE @time_now datetime,
            @time_start_day datetime,
            @time_finish_day datetime;

    SET @time_now = GETDATE();
    SET @time_start_day = CONVERT(datetime, CONVERT(date, @time_now));
    SET @time_finish_day = DATEADD(second, -1, DATEADD(day, 1, CONVERT(datetime,
CONVERT(date, @time_now))));

    SELECT @time_now AS time_now, @time_start_day AS time_start_day,
@time_finish_day AS time_finish_day;

    SELECT COUNT(id_ticket) as total_tickets
    FROM tickets
    WHERE order_id IS NOT NULL
    AND order_id <> '';
END;

exec get_all_tickets_per_day

--DROP PROCEDURE IF EXISTS get_all_tickets_per_month;

CREATE PROCEDURE get_all_tickets_per_month
AS
BEGIN
    DECLARE @time_now datetime,
            @time_start_month datetime,
            @time_finish_month datetime;

    SET @time_now = GETDATE();
    SET @time_start_month = DATEFROMPARTS(YEAR(@time_now), MONTH(@time_now), 1);
    -- Початок поточного місяця
    SET @time_finish_month = DATEADD(second, -1, DATEADD(month, 1,
@time_start_month)); -- Кінець поточного місяця

    SELECT @time_now AS time_now, @time_start_month AS time_start_month,
@time_finish_month AS time_finish_month;

    SELECT COUNT(id_ticket) as total_tickets
    FROM tickets
    WHERE order_id IS NOT NULL
    AND order_id <> ''
    AND date_buying BETWEEN @time_start_month AND @time_finish_month;
END;

exec get_all_tickets_per_month

--DROP PROCEDURE IF EXISTS get_all_tickets_per_year;

CREATE PROCEDURE get_all_tickets_per_year
AS
BEGIN
    DECLARE @time_now datetime,

```

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижмотря О.В.				148
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        @time_start_year datetime,
        @time_finish_year datetime;

SET @time_now = GETDATE();
SET @time_start_year = DATEFROMPARTS(YEAR(@time_now), 1, 1); -- Початок
поточного року
SET @time_finish_year = DATEADD(second, -1, DATEADD(year, 1,
@time_start_year)); -- Кінець поточного року

SELECT @time_now AS time_now, @time_start_year AS time_start_year,
@time_finish_year AS time_finish_year;

SELECT COUNT(id_ticket) as total_tickets
FROM tickets
WHERE order_id IS NOT NULL
AND order_id <> ''
AND date_buying BETWEEN @time_start_year AND @time_finish_year;
END;

exec get_all_tickets_per_year

--DROP PROCEDURE IF EXISTS delete_all_non_active_journeys;

CREATE PROCEDURE get_all_tickets_per_year
AS
BEGIN
    DECLARE @time_now datetime,
            @time_start_year datetime,
            @time_finish_year datetime;

    SET @time_now = GETDATE();
    SET @time_start_year = DATEFROMPARTS(YEAR(@time_now), 1, 1); -- Початок
поточного року
    SET @time_finish_year = DATEADD(second, -1, DATEADD(year, 1,
@time_start_year)); -- Кінець поточного року

    SELECT @time_now AS time_now, @time_start_year AS time_start_year,
@time_finish_year AS time_finish_year;

    SELECT COUNT(id_ticket) as total_tickets
    FROM tickets
    WHERE order_id IS NOT NULL
    AND order_id <> ''
    AND date_buying BETWEEN @time_start_year AND @time_finish_year;
END;

exec get_all_tickets_per_year

```

when_delete_bus.sql

```
USE [avtovokzal]
```

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижмотря О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		149

```

GO
drop trigger[when_delete_bus]

go
create TRIGGER when_delete_bus
ON buses
INSTEAD OF DELETE
AS
BEGIN
    PRINT 'Trigger fired';
    DECLARE @id_bus INT;
    SELECT @id_bus = id_bus FROM deleted;
    EXEC drop_FK_timetable_buses @id_bus;
    delete from buses
    where id_bus=@id_bus
END;

--0->>

create PROCEDURE drop_FK_timetable_buses
    @id_bus INT
AS
BEGIN
    BEGIN TRANSACTION;
    BEGIN TRY
        UPDATE timetable
        SET bus_id = NULL
        WHERE bus_id = @id_bus;

        print ('EXEC delete_null_journey start')
        EXEC delete_null_journey;
        COMMIT TRANSACTION;
    END TRY

    BEGIN CATCH
        print ('rollback')
        ROLLBACK TRANSACTION;
    END CATCH;
END;

--0->>

create PROCEDURE delete_null_journey
AS
BEGIN

    DELETE FROM timetable
    WHERE bus_id IS NULL;

```

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижмотря О.В.				150
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        print ('EXEC delete_null_journey end')
END;

```

when_delete_journey.sql

```

USE [avtovokzal]
GO

```

```

IF EXISTS (SELECT * FROM sys.triggers WHERE name = 'when_delete_journey')
BEGIN
    DROP TRIGGER when_delete_journey;
END
GO

```

```

CREATE TRIGGER when_delete_journey
ON timetable
INSTEAD OF DELETE
AS
BEGIN
    PRINT 'Trigger fired';
    DECLARE @id_journey INT;
    SELECT @id_journey = id_journey FROM deleted;
    EXEC drop_FK_tickets_timetable @id_journey;
    DELETE FROM timetable
    WHERE id_journey = @id_journey;
END;
GO

```

```

IF EXISTS (SELECT * FROM sys.procedures WHERE name = 'drop_FK_tickets_timetable')
BEGIN
    DROP PROCEDURE drop_FK_tickets_timetable;
END
GO

```

```

CREATE PROCEDURE drop_FK_tickets_timetable
    @id_journey INT
AS
BEGIN
    BEGIN TRANSACTION;
    BEGIN TRY
        UPDATE tickets
        SET journey_id = NULL
        WHERE journey_id = @id_journey;

        PRINT 'EXEC delete_null_journey start';
        EXEC delete_null_tickets;

        COMMIT TRANSACTION;
    END TRY

```

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижмотря О.В.				151
Змн.	Арк.	№ докум.	Підпис	Дата		

```

BEGIN CATCH
    PRINT 'Rollback';
    ROLLBACK TRANSACTION;
END CATCH;
END;
GO

IF EXISTS (SELECT * FROM sys.procedures WHERE name = 'delete_null_tickets')
BEGIN
    DROP PROCEDURE delete_null_tickets;
END
GO

CREATE PROCEDURE delete_null_tickets
AS
BEGIN
    DELETE FROM tickets
    WHERE journey_id IS NULL;
    PRINT 'EXEC delete_null_journey end';
END;
GO

```

when_delete_ticket.sql

```

USE [avtovokzal]
GO

IF EXISTS (SELECT * FROM sys.triggers WHERE name = 'when_delete_ticket')
BEGIN
    DROP TRIGGER when_delete_ticket;
END
GO

create TRIGGER when_delete_ticket
ON tickets
INSTEAD OF DELETE
AS
BEGIN
    PRINT 'Trigger fired';
    DECLARE @id_ticket INT,@id_order int;
    SELECT @id_ticket = id_ticket FROM deleted;
    SELECT @id_order = order_id FROM deleted;

    DELETE FROM tickets
    WHERE id_ticket = @id_ticket;
    exec delete_order @id_order
END;
GO

IF EXISTS (SELECT * FROM sys.procedures WHERE name = 'delete_order')

```

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижмотря О.В.				152
Змн.	Арк.	№ докум.	Підпис	Дата		


```

BEGIN
    DROP PROCEDURE delete_order;
END
GO
create procedure delete_order
@id_order int
as
begin
    DELETE FROM orders
    WHERE id_order = @id_order;
end

```

		Середа І.Р.			ДУ «Житомирська політехніка».24.121.19.000 - ПЗ	Арк.
		Чижмотря О.В.				153
Змн.	Арк.	№ докум.	Підпис	Дата		