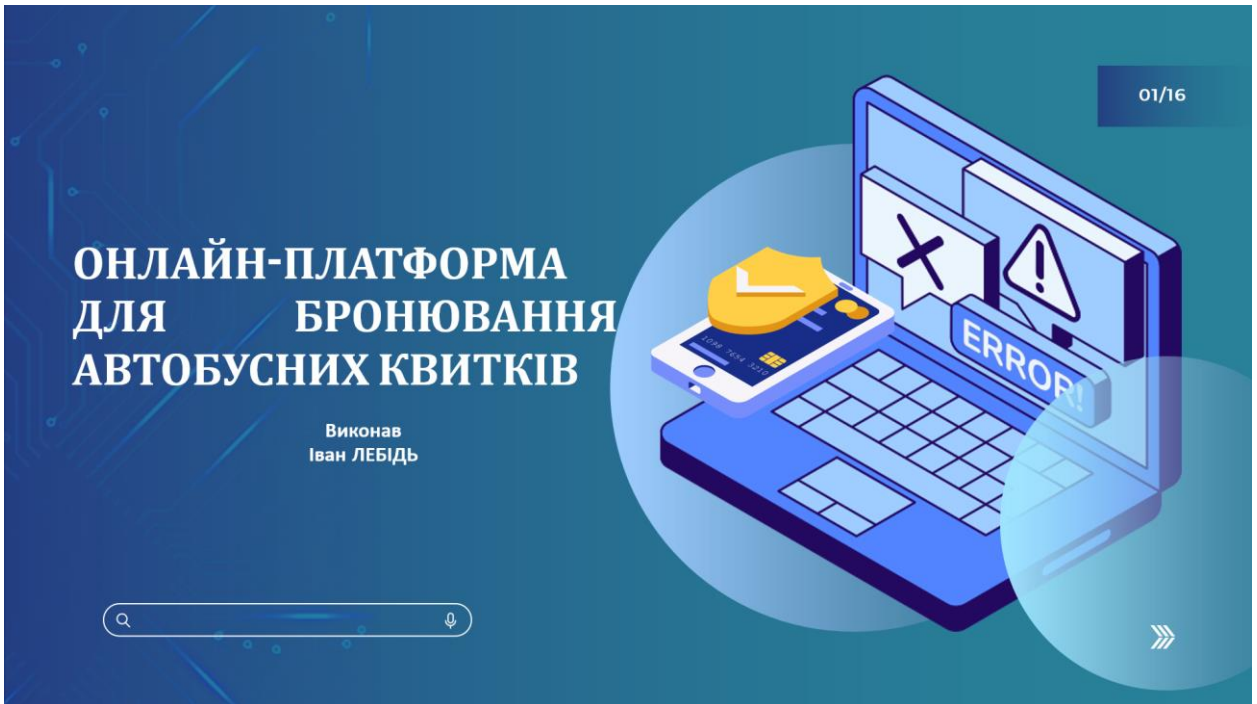


## ДОДАТОК А

Графічний матеріал кваліфікаційної роботи



02/16

## АКТУАЛЬНІСТЬ ТЕМИ

У багатьох регіонах України процес бронювання автобусних квитків досі відбувається телефоном або на автостанціях. Це створює незручності, збільшує ризик помилок і ускладнює доступ до актуальної інформації.

Тому виникає потреба у створенні сучасної, масштабованої веб-платформи для бронювання автобусних квитків з розширеним функціоналом.

Кращі авіаквитки за найнижчими цінами

Літак | Потяг | Автобус | Готель | Страхування

Звідки  
Харків, Україна

Куди

Туди | Назад

Пасажири, клас  
1 пасажир, будь-який

Знайти

Search bar with magnifying glass and share icon.

Navigation arrow (three chevrons).

03/16

# МЕТА ТА ЗАВДАННЯ

Метою є створення веб-застосунку для онлайн-бронювання автобусних квитків із зручним інтерфейсом, підтримкою пошуку рейсів, бронювання та інтеграції з платіжними сервісами.



## Аналіз ринку

Дослідження наявних рішень (Busfor, Tickets.ua) та виявлення їхніх обмежень



## Розробка клієнтської частини

Інтерфейс для користувача з формами пошуку, вибору квитка та підтвердження замовлення



## Розробка серверної логіки

Обробка запитів, збереження даних у MySQL, авторизація та сесії



## Демонстраційне розгортання у Tomcat

Запуск веб-застосунку у середовищі Apache Tomcat для перевірки працездатності



04/16

# АНАЛІЗ РИНКУ І ПРОБЛЕМАТИКА




СЕРВІС	АВТОМАТИЗАЦІЯ	ЗРУЧНІСТЬ ІНТЕРФЕЙСУ	ПІДТРИМКА ЛОКАЛЬНИХ МАРШРУТІВ
OLX	✘	⚠	⚠
BUSFOR	✓	✓	⚠
TICKETS.UA	⚠	✓	⚠
ВЛАСНА СИСТЕМА	✓	✓	✓




05/16

# ВИБІР ТЕХНОЛОГІЙ: FRONT-END



**Чому обрано JSP та JSTL?**

JSP та JSTL забезпечують безшовну інтеграцію з Java-бекендом. Вони дозволяють ефективно динамічно генерувати HTML-код, використовуючи дані з Java-сервісів



**Bootstrap для адаптивності**

Bootstrap – фреймворк для створення адаптивного і сучасного інтерфейсу. Завдяки йому користувацький інтерфейс коректно відображається на різних пристроях і розширеннях екранів

»»

06/16

# ВИБІР ТЕХНОЛОГІЙ: BACK-END

**01**

**Spring Boot**

Забезпечує швидку розробку завдяки модульній архітектурі, підтримує REST API та легко інтегрується з іншими технологіями Java.

**02**

**Hibernate / JPA**

Спрощує роботу з базами даних, автоматизуючи збереження, оновлення та видалення об'єктів через ORM (Object-Relational Mapping).

**03**

**MySQL**

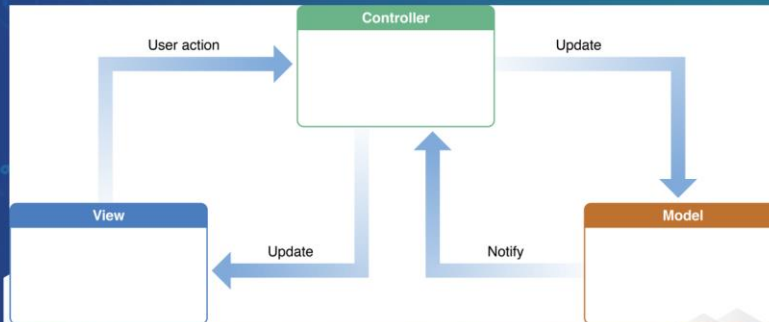
Надійна та безкоштовна реляційна СУБД, яка добре масштабується та підходить як для малих, так і для великих проєктів.

»»

⏪ ⏩ 🔍 🔄 🏠

# АРХІТЕКТУРНА СИСТЕМИ (MVC)

07/16



## Model

- зберігає й обробляє дані: рейси, користувачі, замовлення
- взаємодіє з MySQL через JPA

## View

- JSP-шаблони формують HTML-сторінки
- показує користувачу результати запитів

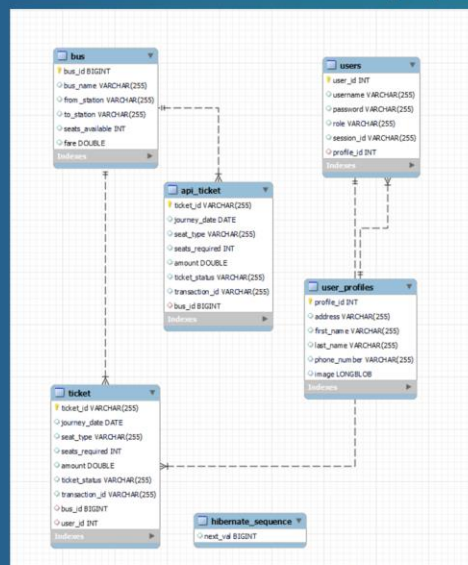
## Controller

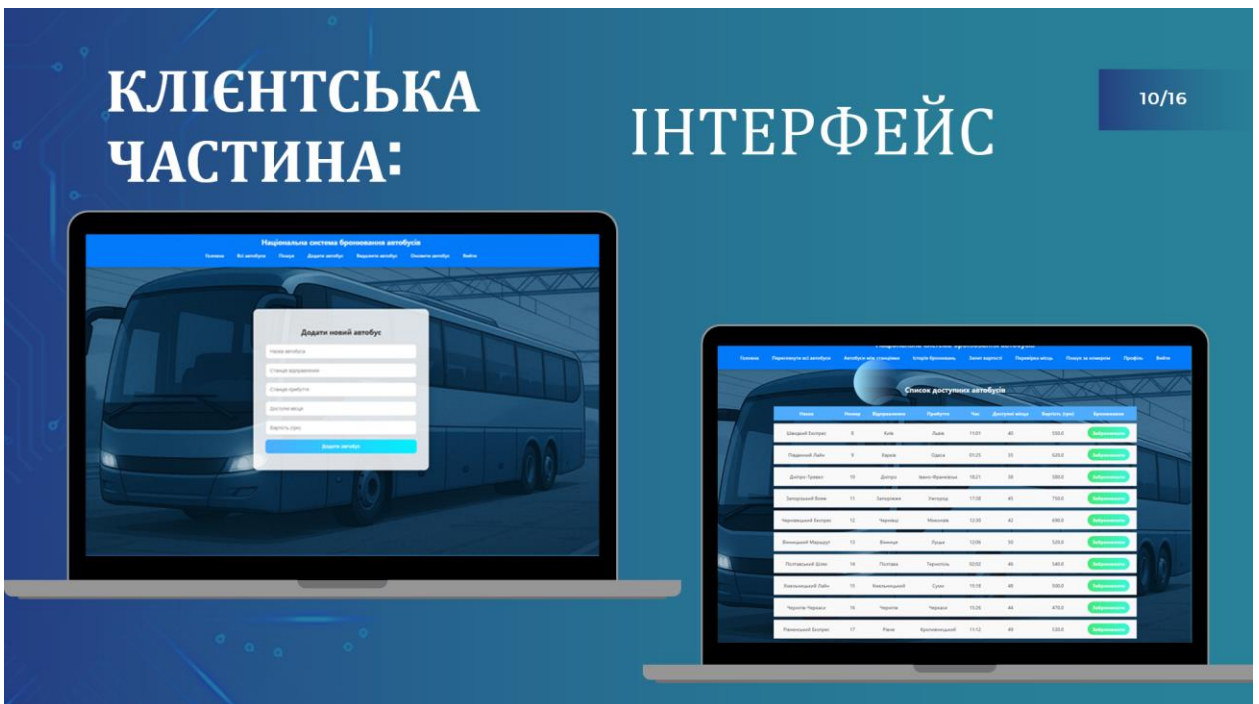
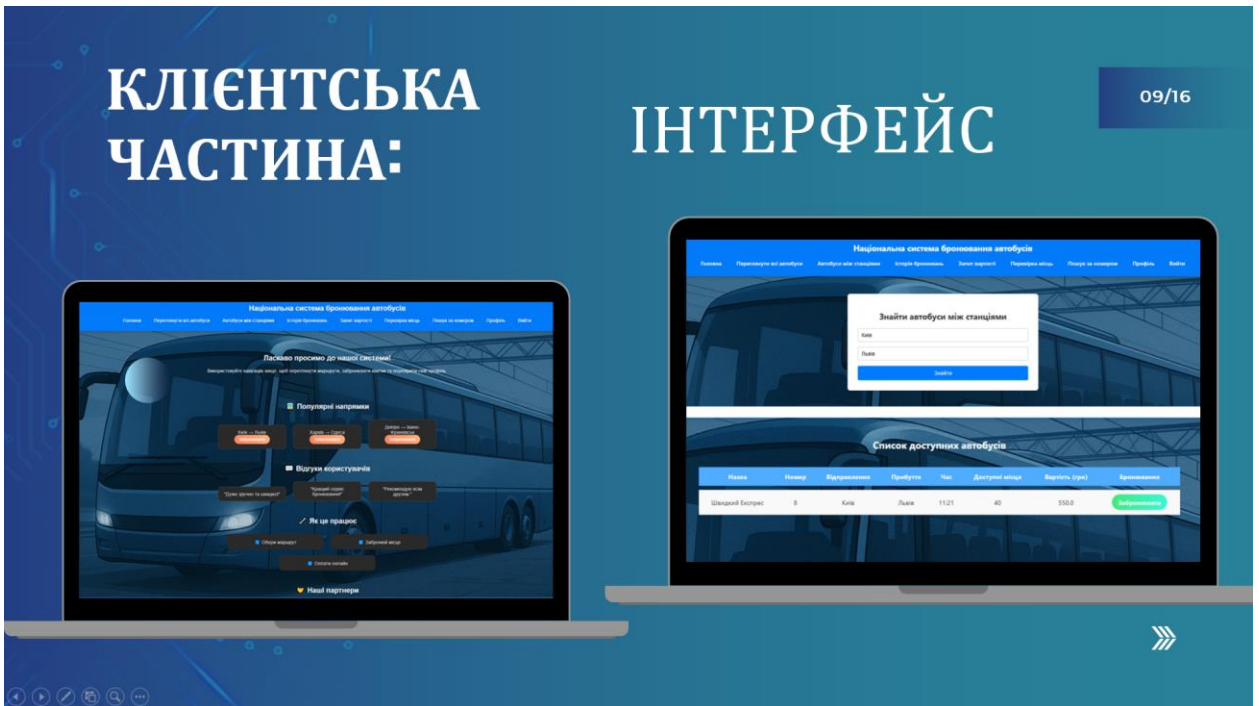
- приймає HTTP-запити
- викликає бізнес-логіку
- передає дані до View



# ER-ДІАГРАМА БАЗИ ДАНИХ

08/16





# КЛІЄНТСЬКА ЧАСТИНА:

# ІНТЕРФЕЙС

11/16

### Форма оплати

Номер картки

Термін дії

Код CVV/CVC

Ім'я власника картки

**Підтвердити оплату**

### Попереднє бронювання

Ім'я користувача: ivan.lebid@gmail.com

Номер автобуса: 7

Станція відправлення: Харків

Станція прибуття: Київ

Дата поїздки: 29.05.2025

Кількість місць: 1

Клас місця: Спальний (SL)

Перевага місця: Без переваги

**Оплатити та забронювати**



# КЛІЄНТСЬКА ЧАСТИНА:

# ІНТЕРФЕЙС

12/16

### Історія бронювань квитків

ID квитка	Назва автобуса	Відправлення	Прибуття	Дата	Місця	Сума (грн)	Статус	Клас
L-1833405987053109249	Швидкий Експрес	Київ	Львів	2025-05-29	1	550.0	<b>BOOKED</b>	Sleeper(SL)

### Бронювання успішно завершено!

Ідентифікатор транзакції: 29b2c40f-bb30-4f8a-98dc-7abb259b3847

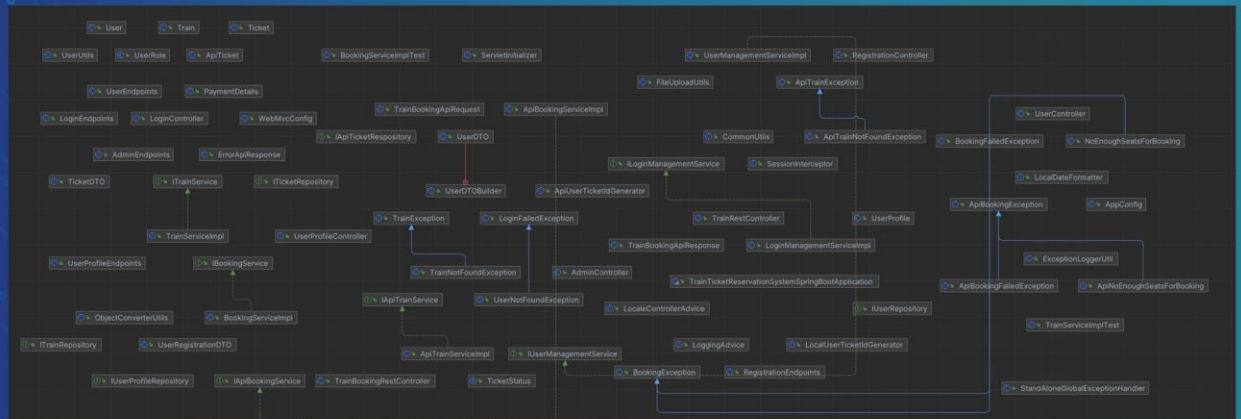
ID квитка	L-1833405987053109249		
Назва автобуса	Швидкий Експрес	Номер автобуса	8
Станція відправлення	Київ	Станція прибуття	Львів
Дата поїздки	2025-05-29	Час	11:23
Кількість пасажирів	1	Клас	Sleeper(SL)
Статус бронювання	<b>BOOKED</b>	Сума	550.0 грн

Будь ласка, збережіть свій ID квитка для подальшого використання та відстеження.



# СЕРВЕРНА ЧАСТИНА: РЕАЛІЗАЦІЯ

13/16



**1. Реєстрація та  
автентифікація користувача**

**2. Пошук рейсів та  
бронювання квитків**

**3. Онлайн-оплата**

**4. Адмін-функції:  
модерація, управління  
розкладами, контроль**

**транзакцій**

**ОСНОВНИЙ  
ФУНКЦІОНАЛ (USER  
FLOW)**

15/16



## ВИСНОВКИ

15/16

### Що було досягнуто:

- Реалізовано веб-застосунок адміністративну панель для управління рейсами, користувач для бронювання автобусних квитків.
- Розроблено адаптивний інтерфейс із використанням Bootstrap.
- Реалізовано апи та автобусами.



## ДОДАТОК Б

### Лістинг Програми

#### Б.1 Лістинг фрагменту pom.xml із основними залежностями проєкту

```

<dependencies>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-web</artifactId>
  </dependency>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-data-jpa</artifactId>
  </dependency>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-security</artifactId>
  </dependency>
  <dependency>
    <groupId>mysql</groupId>
    <artifactId>mysql-connector-java</artifactId>
  </dependency>
  <dependency>
    <groupId>org.thymeleaf</groupId>
    <artifactId>thymeleaf-spring5</artifactId>
  </dependency>
  <dependency>
    <groupId>org.projectlombok</groupId>
    <artifactId>lombok</artifactId>
    <scope>provided</scope>
  </dependency>
  <!-- Додаткові залежності за потреби -->
</dependencies>

```

#### Б.2 Лістинг фрагменту налаштування log4j2.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<Configuration status="WARN">
  <Appenders>
    <Console name="Console" target="SYSTEM_OUT">
      <PatternLayout>
        pattern="%d{HH:mm:ss.SSS} [%t] %-5level
%logger{36} - %msg%n" />
      </Console>
    <File name="FileLogger" fileName="logs/app.log">

```

```

        <PatternLayout
            pattern="%d{yyyy-MM-dd HH:mm:ss} %-5p %c{1}:%L -
%m%n" />
    </File>
</Appenders>
<Loggers>
    <Root level="info">
        <AppenderRef ref="Console" />
        <AppenderRef ref="FileLogger" />
    </Root>
</Loggers>
</Configuration>

```

### Б.3 Лістинг опису сутності User з використанням JPA-анотацій для зв'язків з таблицею бронювань

```

@Entity
@Table(name = "users")
public class User {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    private String name;

    private String email;

    private String password;

    private String role;

    @OneToMany(mappedBy = "user", cascade = CascadeType.ALL)
    private List<Booking> bookings = new ArrayList<>();

    // геттери і сеттери
}

```

### Б.4 Опис сутності User з використанням JPA-анотацій для зв'язків з таблицею бронювань

```

@PostMapping("/register")
public String registerUser(@ModelAttribute("user") User user,
Model model) {
    user.setRole("ROLE_USER");
    boolean isRegistered = userService.registerUser(user);
    if (isRegistered) {

```

```

        return "redirect:/login?success";
    } else {
        model.addAttribute("error", "Користувач із такою
електронною поштою вже існує.");
        return "register";
    }
}

```

#### Б.4 Лістинг реалізація контролера для реєстрації нового користувача

```

@PostMapping("/register")
public String registerUser(@ModelAttribute("user") User user,
Model model) {
    user.setRole("ROLE_USER");
    boolean isRegistered = userService.registerUser(user);
    if (isRegistered) {
        return "redirect:/login?success";
    } else {
        model.addAttribute("error", "Користувач із такою
електронною поштою вже існує.");
        return "register";
    }
}

```

#### Б.5 Лістинг перевірки авторизації користувача у HTTP-запиті через Inteceptor

```

@Override
public boolean preHandle(HttpServletRequest request,
HttpServletRequest response, Object handler) throws Exception {
    HttpSession session = request.getSession(false);
    if (session != null && session.getAttribute("user") != null)
    {
        return true;
    } else {
        response.sendRedirect("/login");
        return false;
    }
}

```

#### Б.6 Лістинг REST-методу для отримання історії бронювань користувача

```

@GetMapping("/booking-history/{userId}")
public ResponseEntity<List<BookingHistoryResponse>>
getBookingHistory(@PathVariable Long userId) {

```

```
List<BookingHistoryResponse> history =
bookingService.getBookingHistory(userId);
return ResponseEntity.ok(history);
}
```

### Б.7 Лістинг коду для додавання нового автобуса через адміністративний контролер

```
@PostMapping("/admin/add-bus")
public String addBus(@ModelAttribute("bus") Bus bus,
RedirectAttributes attributes) {
    busService.saveBus(bus);
    attributes.addFlashAttribute("success", "Новий автобус
успішно додано.");
    return "redirect:/admin/view-buses";
}
```

### Б.8 Лістинг коду, що відповідає за відображення списку користувачів у адміністративній панелі

```
@GetMapping("/admin/view-users")
public String viewUsers(Model model) {
    List<User> users = userService.getAllUsers();
    model.addAttribute("users", users);
    return "admin/view-users";
}
```