

BỘ THÔNG TIN VÀ TRUYỀN THÔNG
HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG



BÁO CÁO CUỐI KỲ
ĐỀ TÀI: XÂY DỰNG WEBSITE QUẢN LÝ BÃI ĐỖ XE THÔNG
MINH

Giảng viên hướng dẫn :

Kim Ngọc Bách

Họ và tên :

Trần Xuân Kiên

Mã sinh viên :

B22DCVT269

Lớp :

E22CQCN02-B

MỤC LỤC

LỜI MỞ ĐẦU	4
1. Giới thiệu	6
1.1. Lý do chọn đề tài	6
1.2. Mục tiêu dự án.....	6
1.3. Phạm vi và đối tượng sử dụng dự án.....	7
2. Công nghệ sử dụng.....	8
3. Phân tích và thiết kế hệ thống.....	8
3.1. Phân tích yêu cầu.....	8
3.2. Thiết kế hệ thống.....	10
4. Triển khai hệ thống	10
4.1. Thiết kế cơ sở dữ liệu	10
4.2. API.....	12
4.3. Sơ đồ cấu trúc Frontend.....	13
4.4. Sơ đồ cấu trúc Backend	15
4.5. Giao diện người dùng	16
5. Thực hiện	21
5.1. Cài đặt môi trường.....	21
5.2. Chạy ứng dụng	22
6. Kết luận và phương hướng phát triển trong tương lai	22
6.1. Kết luận.....	22
6.2. Hạn chế của dự án	23

6.3. Bài học kinh nghiệm.....	24
6.4. Phương hướng phát triển	25

LỜI MỞ ĐẦU

Trong bối cảnh đô thị hóa ngày càng phát triển, nhu cầu sử dụng phương tiện cá nhân tăng cao dẫn đến tình trạng quá tải tại các bãi đỗ xe. Việc quản lý bãi đỗ xe bằng phương pháp truyền thống gặp nhiều khó khăn như tốn kém nhân lực, thời gian chờ đợi lâu, khó kiểm soát số lượng xe ra vào, và dễ xảy ra sai sót trong quá trình vận hành. Điều này đặt ra yêu cầu cấp thiết về một hệ thống quản lý bãi đỗ xe hiện đại, hiệu quả và tiện lợi hơn.

Nhằm giải quyết những vấn đề trên, đề tài "Xây dựng website quản lý bãi đỗ xe thông minh" được thực hiện với mục tiêu xây dựng một hệ thống hỗ trợ việc quản lý bãi đỗ xe một cách tự động, minh bạch và hiệu quả. Hệ thống được phát triển trên nền tảng web, sử dụng công nghệ NodeJS, MySQL và ReactJS, giúp quản lý thông tin xe vào/ra, số lượng chỗ trống, đặt chỗ trước, cũng như cung cấp giao diện thân thiện cho người dùng.

Bên cạnh đó, hệ thống không chỉ giúp chủ bãi xe giảm tải công việc quản lý mà còn mang đến sự thuận tiện cho khách hàng, nâng cao trải nghiệm sử dụng dịch vụ đỗ xe. Với sự phát triển của công nghệ, giải pháp này có thể mở rộng và nâng cấp nhằm đáp ứng nhu cầu thực tế một cách linh hoạt.

Đề tài này không chỉ có ý nghĩa thực tiễn cao mà còn là cơ hội để nhóm nghiên cứu tiếp cận, học hỏi và áp dụng các công nghệ mới vào thực tiễn. Hy vọng rằng hệ thống này sẽ góp phần cải thiện hiệu quả quản lý bãi đỗ xe và mang lại những giá trị thiết thực cho người dùng.

Với việc phát triển hệ thống này, em dự định sẽ ứng dụng vào bãi đỗ xe của Học viện mình nhằm giúp các bạn học sinh cũng như giáo viên có thể theo dõi được chỗ gửi xe của Học viện còn chỗ hay không, có thể giảm tải việc mất thời gian trong việc gửi xe của mình.

1. Giới thiệu

1.1. Lý do chọn đề tài

Việc lựa chọn đề tài “Quản lý bãi đỗ xe thông minh” xuất phát từ nhu cầu thực tế tại Học viện Công nghệ Bưu chính Viễn thông, nơi mà số lượng sinh viên, giảng viên và nhân viên ngày càng gia tăng, kéo theo áp lực lớn lên hệ thống bãi đỗ xe hiện tại. Hiện nay, việc gửi xe tại Học viện vẫn chủ yếu được thực hiện theo phương pháp thủ công, thiếu tính tự động, gây ra nhiều bất tiện như mất thời gian khi gửi hoặc lấy xe, khó kiểm soát lượng xe ra vào, thiếu số liệu thống kê chính xác và không hỗ trợ các hình thức đặt chỗ hay thanh toán tiện lợi. Trước thực trạng đó, hệ thống quản lý bãi đỗ xe thông minh ra đời nhằm số hóa toàn bộ quy trình vận hành bãi xe, giúp người dùng có thể dễ dàng đặt chỗ, thanh toán, đồng thời giúp ban quản lý theo dõi, kiểm soát và tối ưu hoạt động hiệu quả hơn. Đề tài không chỉ giải quyết bài toán quản lý trong thực tế tại Học viện Công nghệ Bưu chính Viễn thông mà còn góp phần hướng tới mô hình trường học thông minh, hiện đại. Đồng thời, đây cũng là cơ hội để sinh viên áp dụng các kiến thức về lập trình web, cơ sở dữ liệu và phát triển phần mềm vào một sản phẩm thực tiễn, có khả năng ứng dụng cao ngay trong chính môi trường học tập của mình.

1.2. Mục tiêu dự án

- Xây dựng một hệ thống quản lý bãi đỗ xe thông minh dưới dạng ứng dụng web, phục vụ cho nhu cầu gửi xe trong khuôn viên Học viện Công nghệ Bưu chính Viễn thông.
- Tự động hóa quy trình gửi và quản lý xe, bao gồm các chức năng như: đăng ký tài khoản, đăng nhập, đặt chỗ, thanh toán theo giờ.
- Tăng tính tiện lợi cho người dùng (sinh viên, giảng viên, nhân viên) bằng cách cho phép tra cứu tình trạng bãi xe, đặt chỗ online và thanh toán không tiền mặt.

- Cung cấp công cụ quản trị cho ban quản lý, giúp giám sát người dùng, khu vực đỗ xe, lượt gửi xe và thống kê doanh thu nhanh chóng, chính xác.
- Hạn chế tình trạng ùn tắc, mất kiểm soát và thất thoát, góp phần xây dựng môi trường học tập hiện đại, văn minh và an toàn.
- Rèn luyện kỹ năng phát triển phần mềm thực tế cho sinh viên: từ thiết kế giao diện, xây dựng chức năng đến kết nối cơ sở dữ liệu và xử lý logic nghiệp vụ.
- Hướng tới mô hình “Trường học thông minh”, từng bước hiện đại hóa hoạt động quản lý trong Học viện.

1.3. Phạm vi và đối tượng sử dụng dự án

- Đối tượng sử dụng hệ thống: Sinh viên, giảng viên, nhân viên và ban quản lý bãi đỗ xe trong khuôn viên Học viện Công nghệ Bưu chính Viễn thông (HVCNBCVT).
- Loại phương tiện hỗ trợ: Chỉ hỗ trợ gửi xe máy – là phương tiện cá nhân phổ biến nhất trong sinh viên và cán bộ Học viện.
- Chức năng người dùng (frontend):
 - Đăng ký/đăng nhập tài khoản.
 - Đặt chỗ gửi xe theo giờ.
 - Tra cứu tình trạng bãi xe theo thời gian thực (dữ liệu giả trong giai đoạn frontend).
 - Thanh toán trực tuyến (giả lập chức năng trong giai đoạn đầu).
- Chức năng người quản trị (admin dashboard):
 - Quản lý người dùng.
 - Quản lý khu vực đỗ xe (số lượng chỗ, tình trạng sử dụng).
 - Quản lý đơn đặt chỗ và thanh toán.
 - Thống kê lượt gửi xe và doanh thu.

2. Công nghệ sử dụng

Trong dự án “Quản lý bãi đỗ xe thông minh”, em đã sử dụng các công nghệ phổ biến và hiện đại trong lập trình web để xây dựng hệ thống gồm ba lớp: frontend, backend và cơ sở dữ liệu. Cụ thể như sau:

- Frontend (giao diện người dùng):
 - ReactJS: Thư viện JavaScript mạnh mẽ được sử dụng để xây dựng giao diện người dùng tương tác, linh hoạt và dễ mở rộng. React giúp quản lý trạng thái hiệu quả, đồng thời hỗ trợ tái sử dụng các thành phần giao diện (components).
- Backend (xử lý logic và API):
 - Node.js kết hợp với Express.js: Node.js là nền tảng chạy JavaScript phía server, giúp xây dựng các API RESTful hiệu quả. Express.js là framework nhẹ giúp tổ chức và quản lý mã nguồn backend một cách rõ ràng và nhanh chóng.
- Cơ sở dữ liệu:
 - MySQL: Hệ quản trị cơ sở dữ liệu quan hệ phổ biến, dùng để lưu trữ thông tin người dùng, khu vực đỗ xe, lịch sử đặt chỗ và thanh toán. MySQL hỗ trợ truy vấn nhanh, có tính ổn định cao và dễ tích hợp với Node.js.

Việc sử dụng bộ ba React – Node.js – MySQL giúp đảm bảo hệ thống hoạt động ổn định, dễ bảo trì và có thể mở rộng trong tương lai.

3. Phân tích và thiết kế hệ thống

3.1. Phân tích yêu cầu

- Yêu cầu chức năng
 - Đối với người dùng (sinh viên, giảng viên, nhân viên):

- Đăng ký và đăng nhập tài khoản.
- Xem thông tin các khu vực bãi đỗ xe (vị trí, số chỗ còn trống).
- Đặt chỗ gửi xe theo giờ.
- Theo dõi lịch sử đặt chỗ và thanh toán.
- Thanh toán chi phí gửi xe qua hình thức trực tuyến (hoặc giả lập chức năng thanh toán).
- Đối với người quản trị (admin):
 - Quản lý tài khoản người dùng.
 - Quản lý khu vực đỗ xe (thêm, sửa, xóa khu vực, cập nhật số chỗ).
 - Quản lý các đơn đặt chỗ và giao dịch thanh toán.
- Yêu cầu phi chức năng
 - Giao diện thân thiện.
 - Hệ thống phản hồi nhanh, xử lý dữ liệu ổn định.
 - Dữ liệu người dùng và giao dịch được lưu trữ an toàn trong cơ sở dữ liệu.
 - Có thể dễ dàng mở rộng thêm chức năng (ví dụ: quét mã QR, tích hợp RFID, v.v.) trong tương lai.
- Yêu cầu kỹ thuật
 - Giao diện người dùng được phát triển bằng ReactJS.
 - Hệ thống backend sử dụng Node.js và Express.js để xây dựng API.
 - Dữ liệu được lưu trữ và quản lý bằng hệ quản trị cơ sở dữ liệu MySQL.
 - Giai đoạn đầu sử dụng dữ liệu giả để xây dựng và kiểm thử frontend, sau đó kết nối với API thật.

3.2. Thiết kế hệ thống

- **Kiến trúc tổng thể**

- Hệ thống được thiết kế theo mô hình client-server gồm 3 lớp chính:
 - Frontend (Client): Giao diện người dùng xây dựng bằng ReactJS, chạy trên trình duyệt, cho phép người dùng tương tác với hệ thống.
 - Backend (Server/API): Xử lý logic nghiệp vụ và cung cấp các API RESTful, phát triển bằng Node.js + Express.js.
 - Cơ sở dữ liệu (Database): Lưu trữ toàn bộ dữ liệu người dùng, khu vực bãi xe, đặt chỗ và thanh toán, sử dụng MySQL.

- **Các phân hệ chính**

- Phân hệ người dùng
 - Đăng ký / Đăng nhập / Đăng xuất.
 - Xem thông tin khu vực bãi đỗ xe.
 - Đặt chỗ gửi xe.
 - Thanh toán (giả lập hoặc tích hợp thanh toán thực).
 - Xem lịch sử đặt chỗ, thanh toán.
- Phân hệ quản trị (Admin)
 - Quản lý tài khoản người dùng.
 - Quản lý khu vực đỗ xe (thêm/sửa/xóa).
 - Quản lý các đơn đặt chỗ và trạng thái thanh toán.

4. Triển khai hệ thống

4.1. Thiết kế cơ sở dữ liệu

- Bảng Users:

```
-- Tạo bảng users
) CREATE TABLE users (
    id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(100) NOT NULL,
    email VARCHAR(100) NOT NULL UNIQUE,
    password VARCHAR(100) NOT NULL,
    phone VARCHAR(20) NOT NULL,
    role ENUM('user', 'admin') DEFAULT 'user',
    token VARCHAR(255),
    createdAt TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

- Bảng Areas:

```
-- Tạo bảng areas
CREATE TABLE areas (
    id INT AUTO_INCREMENT PRIMARY KEY,
    areaId VARCHAR(10) NOT NULL UNIQUE,
    name VARCHAR(100) NOT NULL,
    totalSpaces INT NOT NULL,
    availableSpaces INT NOT NULL,
    pricePerHour DECIMAL(10,2) NOT NULL,
    status ENUM('active', 'maintenance', 'closed') DEFAULT 'active',
    createdAt TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

- Bảng Bookings:

```
-- Tạo bảng bookings
) CREATE TABLE bookings (
    id INT AUTO_INCREMENT PRIMARY KEY,
    userId INT NOT NULL,
    areaId VARCHAR(10) NOT NULL,
    startTime DATETIME NOT NULL,
    endTime DATETIME NOT NULL,
    status ENUM('pending', 'confirmed', 'cancelled', 'completed') DEFAULT 'pending',
    createdAt TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (userId) REFERENCES users(id),
    FOREIGN KEY (areaId) REFERENCES areas(areaId)
);
```

- Bảng Payments:

```
-- Tạo bảng payments
CREATE TABLE payments (
  id INT AUTO_INCREMENT PRIMARY KEY,
  bookingId INT NOT NULL,
  userId INT NOT NULL,
  amount DECIMAL(10,2) NOT NULL,
  status ENUM('pending', 'completed', 'failed', 'refunded') DEFAULT 'pending',
  paymentMethod VARCHAR(50) NOT NULL,
  createdAt TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
  FOREIGN KEY (bookingId) REFERENCES bookings(id),
  FOREIGN KEY (userId) REFERENCES users(id)
);
```

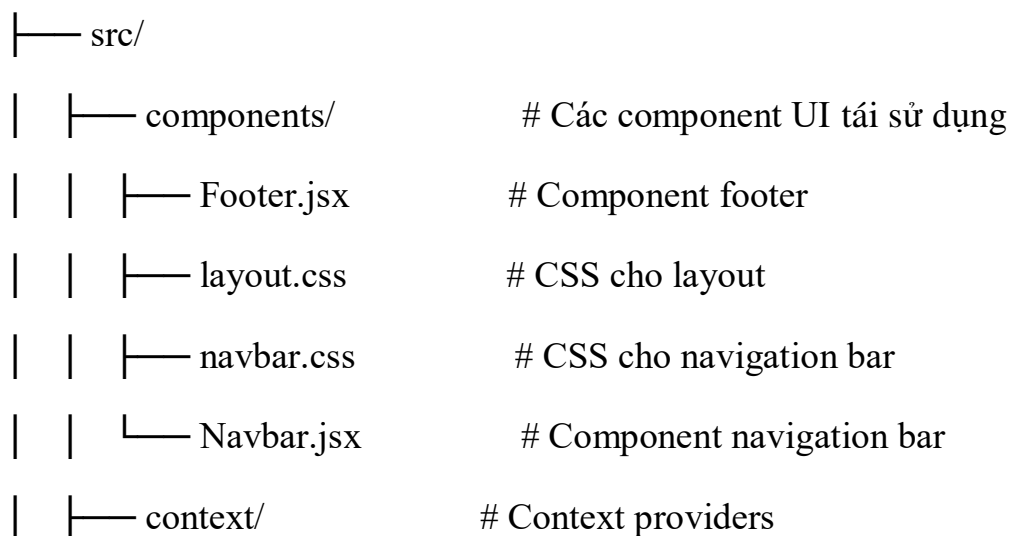
4.2. API

- **Auth & Register**
 - POST /register – Đăng ký tài khoản người dùng
- **User (Admin)**
 - GET /admin/users – Lấy danh sách tất cả người dùng
- **Khu vực bãi đỗ xe (Area)**
 - API cho người dùng:
 - POST /area/ – Tạo khu vực bãi đỗ mới
 - GET /area/ – Lấy danh sách tất cả khu vực
 - GET /area/available – Lấy danh sách khu vực còn trống
 - GET /area/:id – Lấy thông tin chi tiết một khu vực
 - API cho admin (qua /admin/areas):
 - GET /admin/areas – Lấy danh sách khu vực
 - POST /admin/areas – Thêm khu vực
 - PUT /admin/areas/:areaId – Cập nhật khu vực
 - DELETE /admin/areas/:areaId – Xóa khu vực
- **Đặt chỗ (Booking)**

- POST /booking/ – Tạo đơn đặt chỗ
- GET /booking/user/:userId – Lấy danh sách đặt chỗ của người dùng
- GET /booking/:id – Lấy chi tiết đơn đặt chỗ
- PATCH /booking/:id/status – Cập nhật trạng thái đơn đặt chỗ
- GET /admin/bookings – Lấy tất cả đơn đặt chỗ
- PUT /admin/bookings/:bookingId/status – Cập nhật trạng thái đơn đặt chỗ
- DELETE /admin/bookings/:bookingId – Xóa đơn đặt chỗ
- **Thanh toán (Payment)**
 - POST /payment/ – Tạo thanh toán mới
 - GET /payment/user/:userId – Lấy danh sách thanh toán của người dùng
 - GET /payment/:id – Lấy chi tiết thanh toán
 - PATCH /payment/:id/status – Cập nhật trạng thái thanh toán
 - GET /admin/payments – Lấy danh sách tất cả thanh toán

4.3. Sơ đồ cấu trúc Frontend

smart-parking/



- | | └─ AuthContext.jsx # Context quản lý xác thực
- | └─ pages/ # Các trang của ứng dụng
- | | └─ admin.css # CSS cho trang admin
- | | └─ AdminDashboard.jsx # Trang dashboard admin
- | | └─ auth.css # CSS cho trang xác thực
- | | └─ booking.css # CSS cho trang booking
- | | └─ Booking.jsx # Trang đặt chỗ
- | | └─ Login.jsx # Trang đăng nhập
- | | └─ payment.css # CSS cho trang thanh toán
- | | └─ Payment.jsx # Trang thanh toán
- | | └─ PaymentSuccess.jsx # Trang thanh toán thành công
- | | └─ Register.jsx # Trang đăng ký
- | └─ services/ # Các service gọi API
- | | └─ adminService.js # Service cho admin API
- | | └─ api.js # Cấu hình API chung
- | | └─ areaService.js # Service cho area API
- | | └─ authService.js # Service cho auth API
- | | └─ bookingService.js # Service cho booking API
- | | └─ paymentService.js # Service cho payment API
- └─ App.css # CSS chính của ứng dụng
- └─ App.jsx # Component chính

- |— index.css # CSS global
- |— index.js # Entry point của ứng dụng
- |— routes.jsx # Định nghĩa routing
- |— .gitignore # Git ignore file
- |— package-lock.json # Lock file dependencies
- |— package.json # Dependencies và scripts

4.4. Sơ đồ cấu trúc Backend

backend/

- |— config/
 - | — database.js # Cấu hình kết nối database
- |— controllers/ # Các controller xử lý logic nghiệp vụ
 - | — adminController.js # Controller quản trị
 - | — areaController.js # Controller quản lý khu vực
 - | — authController.js # Controller xác thực
 - | — bookingController.js # Controller đặt chỗ
 - | — paymentController.js # Controller thanh toán
 - | — registerController.js # Controller đăng ký
- |— database/
 - | — schema.sql # Schema cơ sở dữ liệu
- |— middleware/ # Các middleware
 - | — auth.js # Middleware xác thực
- |— models/ # Các model dữ liệu

— Areas.js	# Model khu vực
— Booking.js	# Model đặt chỗ
— Payment.js	# Model thanh toán
— User.js	# Model người dùng
— routes/	# Định nghĩa các route API
— adminRoutes.js	# Routes cho admin
— area.js	# Routes cho khu vực
— auth.js	# Routes xác thực
— booking.js	# Routes đặt chỗ
— index.js	# Route chính
— payment.js	# Routes thanh toán
— register.js	# Routes đăng ký
— index.js	# Entry point của ứng dụng
— package-lock.json	# Lock file dependencies
— package.json	# Dependencies và scripts

4.5. Giao diện người dùng

Bãi Đỗ Xe PTIT

Đăng nhập

Đăng ký

Đăng nhập

Email:

Mật khẩu:

Đăng nhập

Chưa có tài khoản? [Đăng ký ngay](#)

Trang Đăng nhập

Bãi Đỗ Xe PTIT

Đăng nhập

Đăng ký

Đăng ký tài khoản

Họ và tên:

Email:

Số điện thoại:

Mật khẩu:

Xác nhận mật khẩu:

Đăng ký

Đã có tài khoản? [Đăng nhập](#)

Trang Đăng ký

Bãi Đỗ Xe PTIT

Trần Xuân KiênĐặt chỗĐăng xuất

Đặt Chỗ Đỗ Xe

Chọn khu vực và thời gian đỗ xe của bạn

Khu Vực Đỗ Xe

Chọn khu vực

Ngày Đặt

mm/dd/yyyy

Giờ Bắt Đầu

--:-- --

Giờ Kết Thúc

--:-- --

Tiếp tục thanh toán

Trang Đặt chỗ

Bãi Đỗ Xe PTIT

Trần Xuân KiênĐặt chỗĐăng xuất

Thanh Toán

Vui lòng kiểm tra thông tin và chọn phương thức thanh toán

Thông Tin Đặt Chỗ

Khu vực:

Khu vực B

Thời gian bắt đầu:

6/2/2025, 10:13:00 AM

Thời gian kết thúc:

6/2/2025, 10:13:00 PM

Tổng tiền:

120,000 VND

Phương Thức Thanh Toán

Thẻ tín dụng

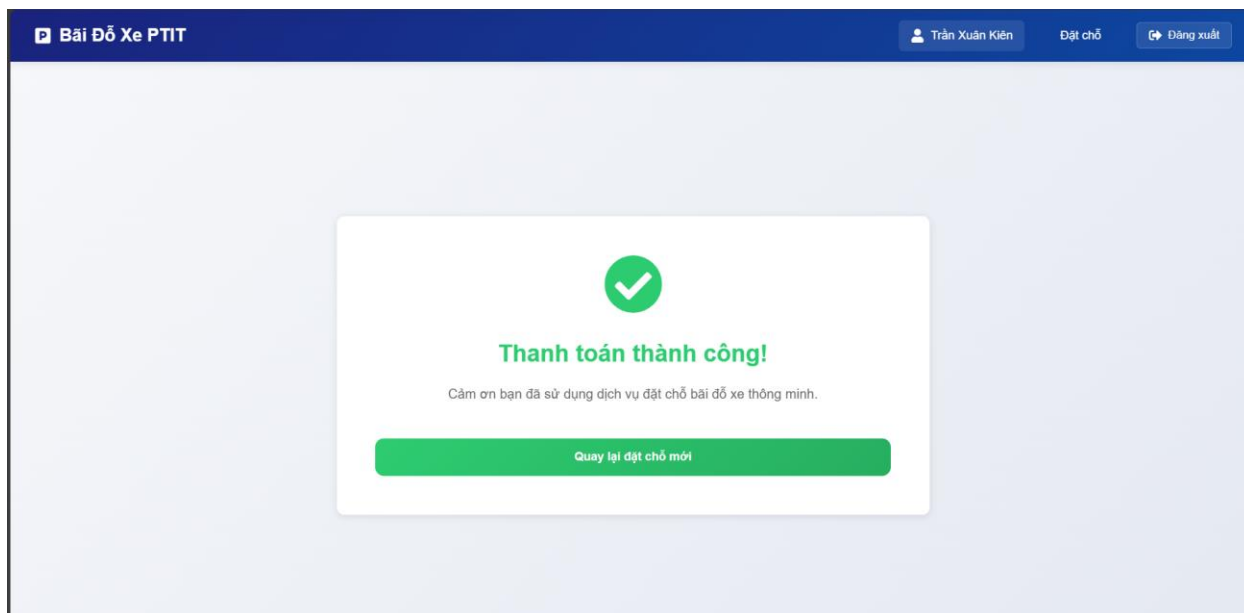
Tiền mặt

QR Code

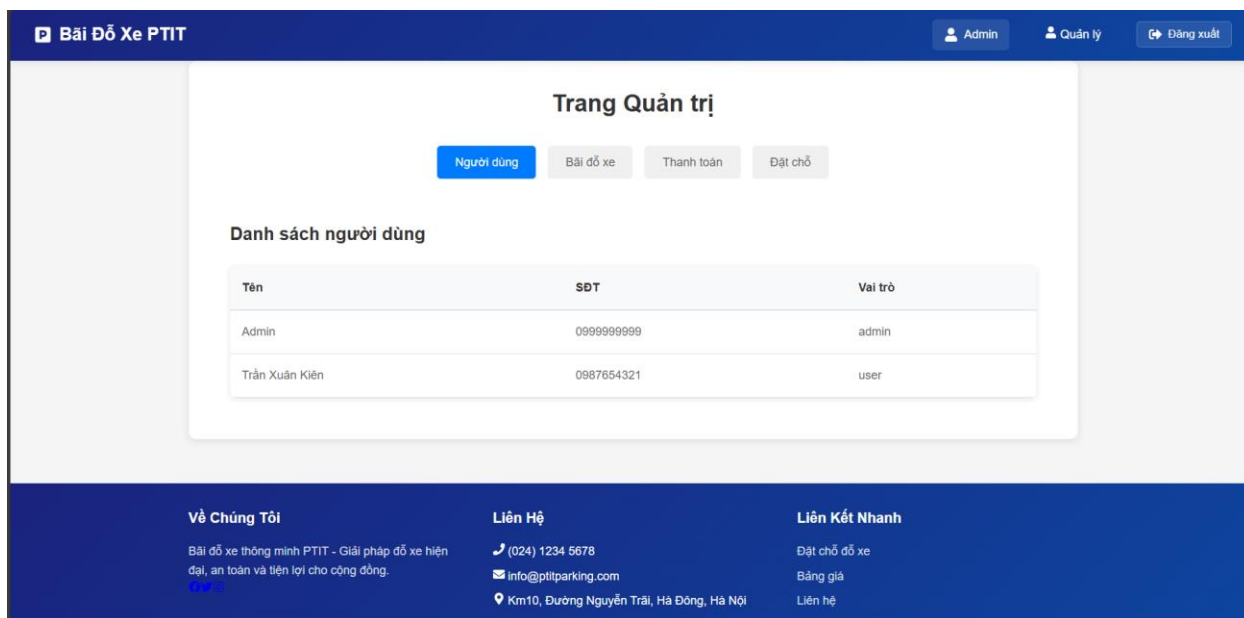
Thanh Toán

Trang Thanh toán

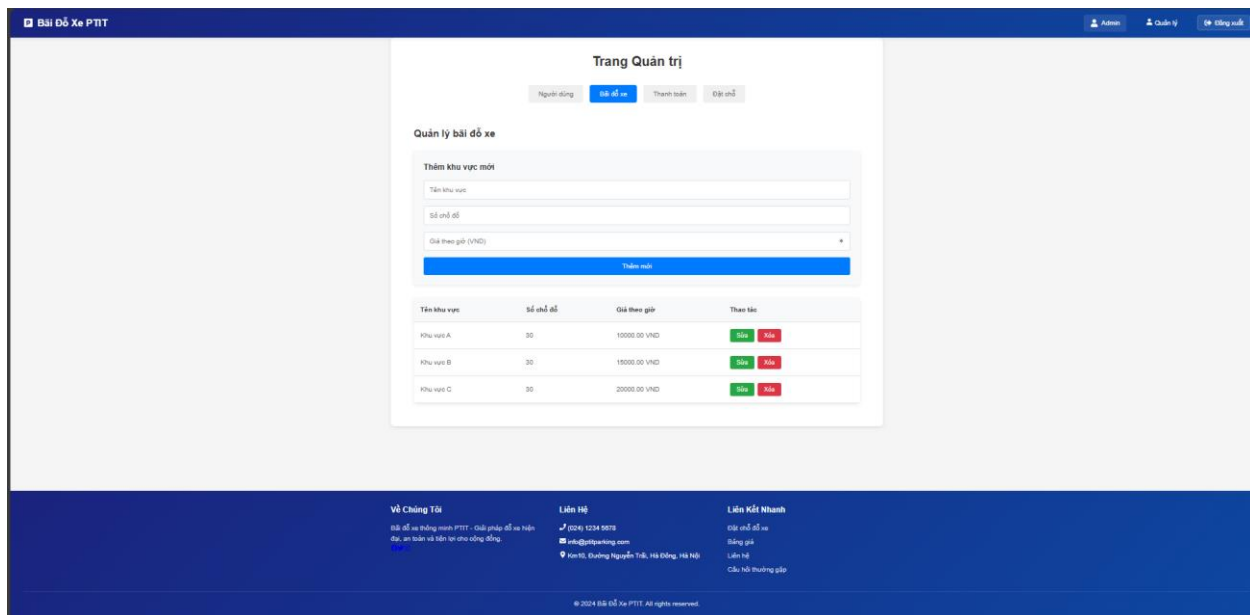
18



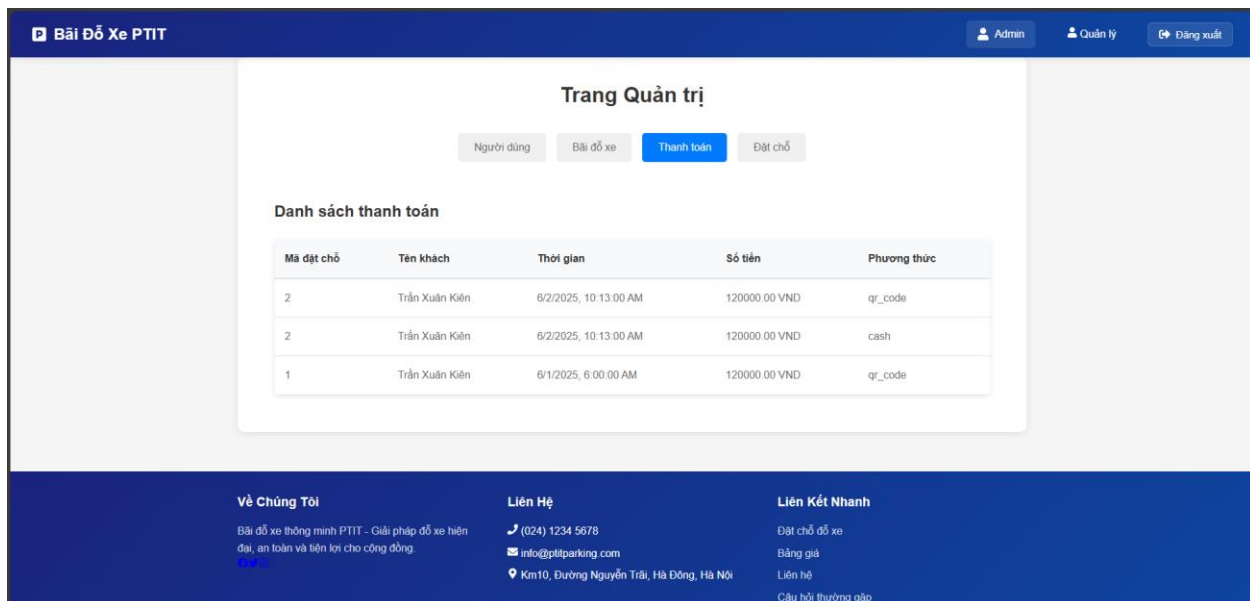
Trang Thanh toán thành công



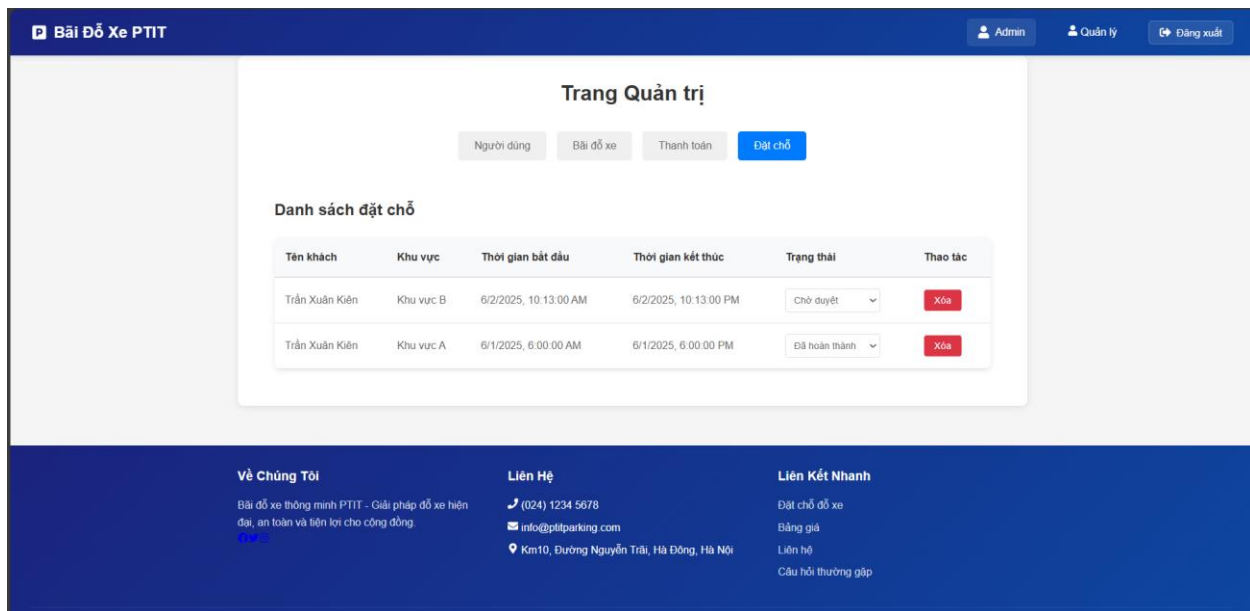
Trang Quản trị - Quản lý người dùng



Trang Quản trị - Quản lý khu vực



Trang Quản trị - Quản lý thanh toán



Trang Quản trị - Quản lý đặt chỗ

5. Thực hiện

5.1. Cài đặt môi trường

- Yêu cầu hệ thống:
 - Node.js version 16.x hoặc cao hơn
 - NPM package manager
 - Git version control
 - Code editor (VS Code khuyến nghị)
- Các bước cài đặt:

1. Clone repository:

```
git clone https://github.com/XuanKien65/Parking_Management.git
```

```
cd parking-management
```

2. Cài đặt dependencies cho backend:

```
cd .\backend\
```

```
npm install
```

3. Cài đặt dependencies cho frontend:

```
cd .\smart-parking\
```

```
npm install
```

5.2. Chạy ứng dụng

Development mode:

- Terminal 1 - Backend:
 - cd backend
 - npm start
- Terminal 2 - Frontend:
 - cd smart-parking
 - npm start

6. Kết luận và phương hướng phát triển trong tương lai

6.1. Kết luận

Qua quá trình nghiên cứu và xây dựng, hệ thống quản lý bãi đỗ xe thông minh đã cơ bản hoàn thiện với các chức năng chính như đăng ký/đăng nhập, đặt chỗ, thanh toán, và quản lý khu vực dành cho admin. Ứng dụng đã đáp ứng nhu cầu thực tế trong việc hỗ trợ sinh viên đại học tìm kiếm và đặt chỗ gửi xe máy một cách thuận tiện, nhanh chóng và minh bạch. Việc áp dụng các công nghệ hiện đại như ReactJS, NodeJS và MySQL giúp đảm bảo tính linh hoạt, hiệu suất và khả năng mở rộng cho hệ thống.

Tuy vẫn còn một số hạn chế cần cải thiện trong các phiên bản tiếp theo, nhưng dự án này đã đặt nền móng vững chắc cho việc triển khai và nâng cấp hệ thống trong tương lai, đặc biệt là việc tối ưu trải nghiệm người dùng.

6.2. Hạn chế của dự án

Mặc dù hệ thống đã đáp ứng được các chức năng cơ bản, nhưng vẫn còn tồn tại một số hạn chế nhất định cần được khắc phục trong tương lai:

- Chưa tích hợp hệ thống kiểm soát thực tế tại bãi đỗ xe
 - Hệ thống hiện chỉ hoạt động trên môi trường giả lập, chưa kết nối với các thiết bị phần cứng như máy quét QR, camera, barrier ra/vào, ...
 - Việc xác thực vào/ra bãi đỗ vẫn còn thủ công, chưa tự động hóa.
- Chưa có cơ chế chống gian lận đặt chỗ
 - Hiện tại người dùng có thể đặt nhiều chỗ cùng lúc hoặc đặt nhưng không sử dụng, gây lãng phí tài nguyên.
 - Thiếu cơ chế xác thực khi sử dụng dịch vụ (ví dụ: kiểm tra mã QR tại cổng ra vào).
- Giao diện còn đơn giản, chưa tối ưu trải nghiệm người dùng
 - Một số giao diện còn sơ khai, chưa đẹp mắt hoặc dễ thao tác đối với người dùng phổ thông.
 - Chưa có hướng dẫn rõ ràng cho người dùng mới trong quá trình đặt chỗ hoặc thanh toán.
- Thiếu tính năng hỗ trợ cho người quản trị hệ thống (admin)
 - Chưa có chức năng phân quyền người dùng chi tiết (chỉ có admin tổng).
 - Việc thống kê doanh thu, báo cáo hoạt động còn đơn giản, chưa có biểu đồ hoặc phân tích nâng cao.
- Hệ thống bảo mật còn cơ bản

- Chưa áp dụng các biện pháp mã hóa mật khẩu người dùng, kiểm soát phiên đăng nhập hoặc chống tấn công như SQL Injection, XSS, ...
- Không có xác thực hai bước (2FA) hoặc giới hạn số lần đăng nhập sai.
- Chưa có khả năng hoạt động ngoại tuyến hoặc khi mất kết nối
 - Ứng dụng phụ thuộc hoàn toàn vào kết nối internet; trong trường hợp mất mạng, người dùng sẽ không thể truy cập hoặc sử dụng dịch vụ.

6.3. Bài học kinh nghiệm

Trong quá trình thực hiện dự án ứng dụng quản lý bãi đỗ xe thông minh, em đã tích lũy được nhiều bài học kinh nghiệm quý báu không chỉ về mặt kỹ thuật mà còn về tư duy phát triển phần mềm và khả năng tự học:

- Nâng cao tư duy phân tích và thiết kế hệ thống
 - Việc bắt đầu từ một ý tưởng và từng bước xây dựng thành một ứng dụng hoàn chỉnh giúp tôi hiểu rõ tầm quan trọng của việc phân tích yêu cầu người dùng, lên kế hoạch phát triển, và thiết kế hệ thống một cách hợp lý.
 - Em học được cách chia nhỏ bài toán, từ thiết kế cơ sở dữ liệu đến cấu trúc giao diện, giúp công việc trở nên dễ kiểm soát hơn.
- Nhận thức rõ tầm quan trọng của trải nghiệm người dùng
 - Trong quá trình xây dựng giao diện, tôi nhận ra rằng trải nghiệm người dùng là yếu tố rất quan trọng. Việc tạo ra một giao diện rõ ràng, dễ sử dụng sẽ giúp người dùng tiếp cận ứng dụng nhanh hơn và hiệu quả hơn.
- Rèn luyện tính kỷ luật và quản lý thời gian
 - Làm việc cá nhân đòi hỏi em phải chủ động lên kế hoạch, phân chia thời gian hợp lý giữa các giai đoạn như thiết kế, code, kiểm thử và viết báo cáo.

- Em học được cách tự đặt deadline và bám sát tiến độ mà không có ai giám sát, điều này giúp em trưởng thành hơn trong cách làm việc.

6.4. Phương hướng phát triển

Trong tương lai, hệ thống quản lý bãi đỗ xe thông minh sẽ tiếp tục được cải tiến và mở rộng theo các hướng sau nhằm nâng cao hiệu quả sử dụng, cải thiện trải nghiệm người dùng và mở rộng khả năng triển khai thực tế:

- Hoàn thiện và nâng cao tính năng gói gửi xe theo tháng
 - Cung cấp nhiều loại gói gửi xe theo tháng (ví dụ: theo ngày trong tuần, theo khung giờ).
 - Tự động nhắc nhở người dùng khi gói gần hết hạn hoặc đã hết hạn.
 - Hỗ trợ thanh toán tự động định kỳ thông qua ví điện tử hoặc tài khoản ngân hàng.
- Cải tiến giao diện và trải nghiệm người dùng (UI/UX)
 - Thiết kế lại giao diện theo hướng trực quan, đơn giản và thân thiện với sinh viên.
 - Tối ưu hiển thị cho thiết bị di động, đảm bảo thao tác nhanh chóng khi đặt chỗ hoặc thanh toán.
 - Tăng cường khả năng tiếp cận (accessibility) cho người dùng có nhu cầu đặc biệt.
- Tích hợp thông báo thời gian thực (Real-time Notification)
 - Gửi thông báo đẩy khi có thay đổi trạng thái đặt chỗ, xác nhận thanh toán, nhắc nhở gia hạn, hoặc thông báo khi bãi đỗ đầy.
 - Sử dụng WebSocket hoặc Firebase để cập nhật thông tin theo thời gian thực.
- Bổ sung bản đồ và tính năng định vị GPS
 - Hiển thị bản đồ vị trí các bãi đỗ trong trường đại học.

- Cho phép người dùng tìm bãi gần nhất còn chỗ trống thông qua định vị GPS.
- Dự đoán khả năng hết chỗ theo thời gian dựa trên dữ liệu lịch sử.
- Tích hợp mã QR và/hoặc thẻ RFID
 - Mỗi đặt chỗ sẽ sinh ra một mã QR để người dùng quét khi vào/ra bãi gửi xe.
 - Hỗ trợ thêm thẻ sinh viên hoặc RFID để nhận diện tự động, giúp kiểm soát nhanh và an toàn hơn.
- Phát triển hệ thống quản trị mở rộng
 - Thêm các vai trò khác ngoài admin như nhân viên hỗ trợ, bảo vệ bãi xe, ...
 - Cho phép phân quyền chi tiết theo chức năng, đảm bảo bảo mật thông tin.
- Thống kê, phân tích và báo cáo dữ liệu
 - Xây dựng dashboard trực quan cho admin với các biểu đồ, số liệu theo thời gian thực.
 - Báo cáo doanh thu, lượt gửi xe, tần suất sử dụng theo khu vực, theo ngày/tuần/tháng.
 - Áp dụng mô hình học máy (machine learning) để dự đoán nhu cầu gửi xe trong tương lai.
- Mở rộng quy mô và khả năng triển khai
 - Chuẩn hóa kiến trúc phần mềm để có thể dễ dàng triển khai ở các trường đại học khác hoặc khu công nghiệp, ký túc xá,...
 - Hỗ trợ đa ngôn ngữ để phù hợp với các môi trường sử dụng khác nhau.
- Tăng cường bảo mật và an toàn dữ liệu

- Mã hóa dữ liệu người dùng, đặc biệt là thông tin thanh toán và tài khoản.
- Triển khai các cơ chế chống gian lận khi đặt chỗ hoặc sử dụng gói tháng.