



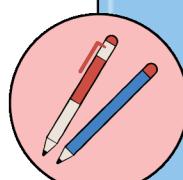
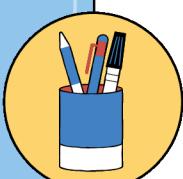
Báo cáo kết thúc môn
Công nghệ Phần mềm

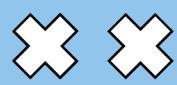
Đề tài:

Xây Dựng Trang Web Tìm Kiếm Phòng Trọ

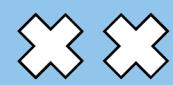
GVHD: TS. Nguyễn Bảo Ân

Sinh viên thực hiện: Nhóm Park





Nội dung



1. Mở đầu

2. Cơ sở lý thuyết

3. Phân tích yêu cầu

4. Thiết kế hệ thống

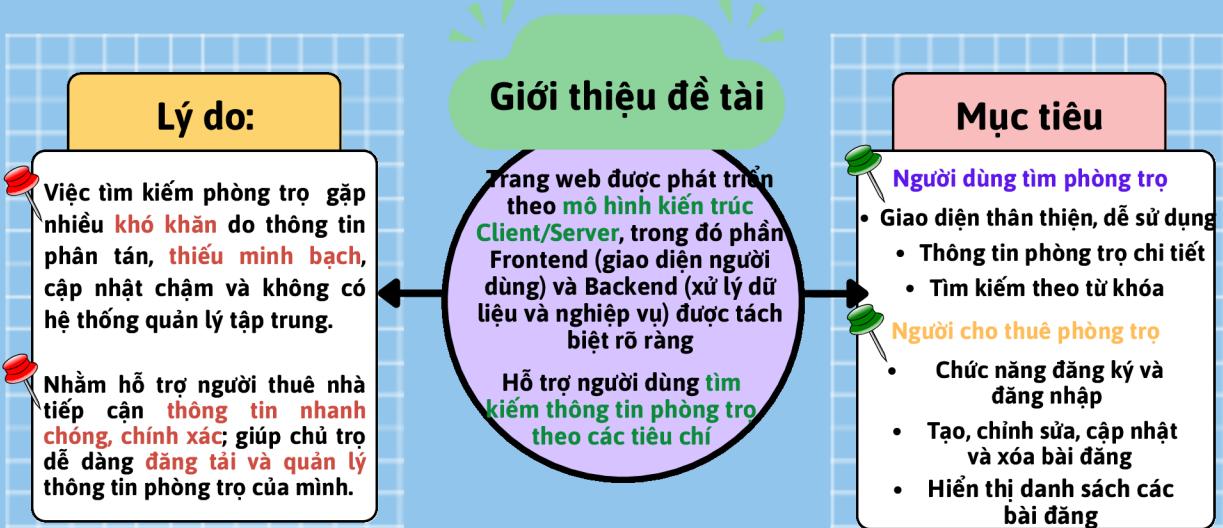
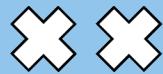
5. Triển khai và công nghệ sử dụng

6. Quản lý dự án

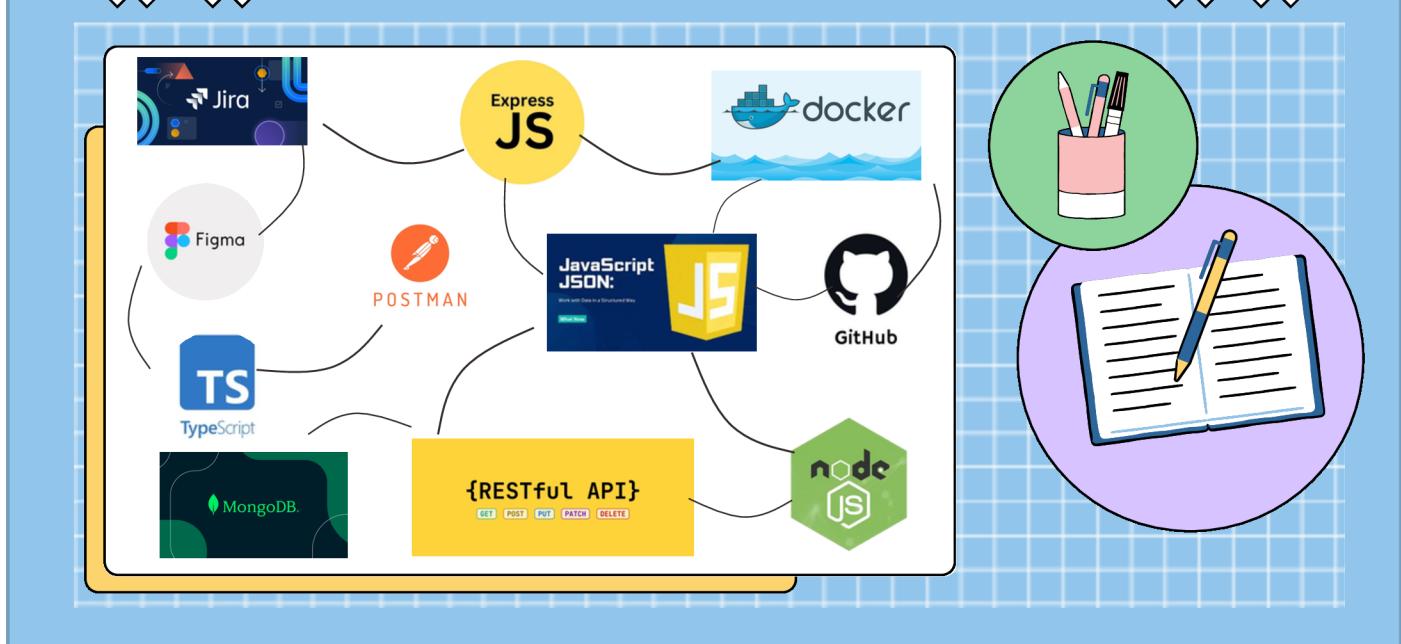
7. Kiểm thử

8. Đánh giá và kết luận

1.TỔNG QUAN



2. Cơ sở lý thuyết



3. Phân tích yêu cầu

Phi chức năng



Chức năng chính

- ✓ Quản lý người dùng
- ✓ Quản lý phòng trọ
- ✓ Quản lý bài đăng cho thuê
- ✓ Tương tác người dùng
- ✓ Quản lý hệ thống
- ✓ Bảo mật và xác thực
- ✓ RESTfull API

• Tính hiệu năng

• Tính sẵn sàng

• Khả năng mở rộng

• Bảo mật

• Tính khả dụng

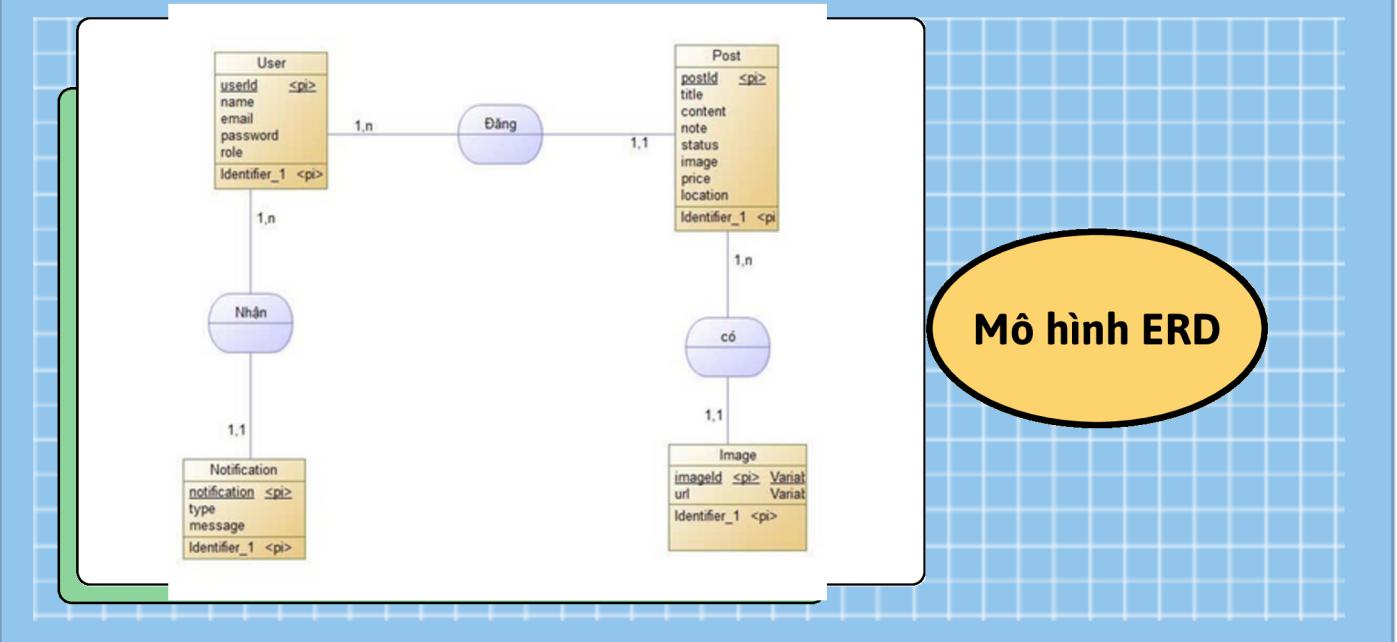
• Dễ bảo trì và phát triển

• Khả năng kiểm thử

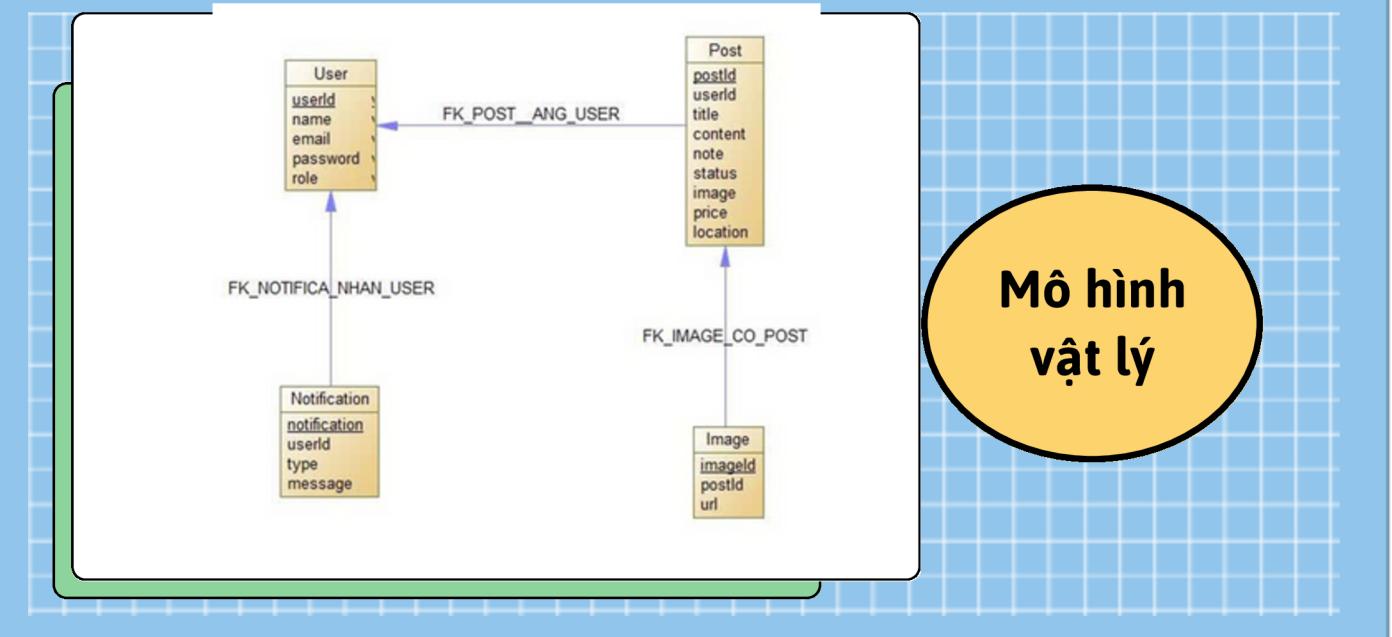
• Khả năng triển khai và vận hành



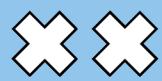
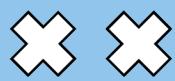
4. Thiết kế hệ thống



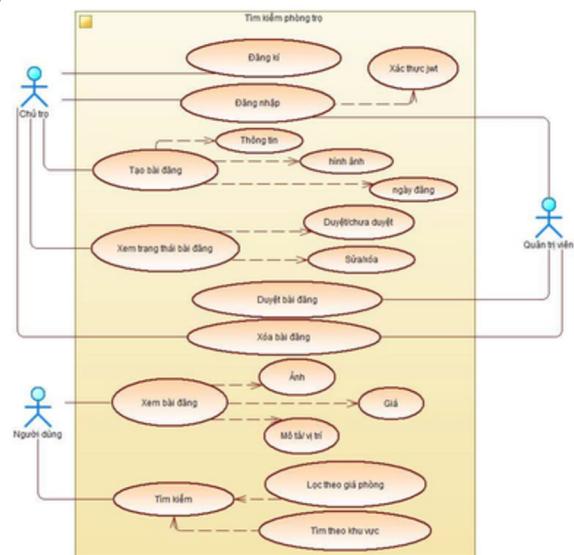
4. Thiết kế hệ thống



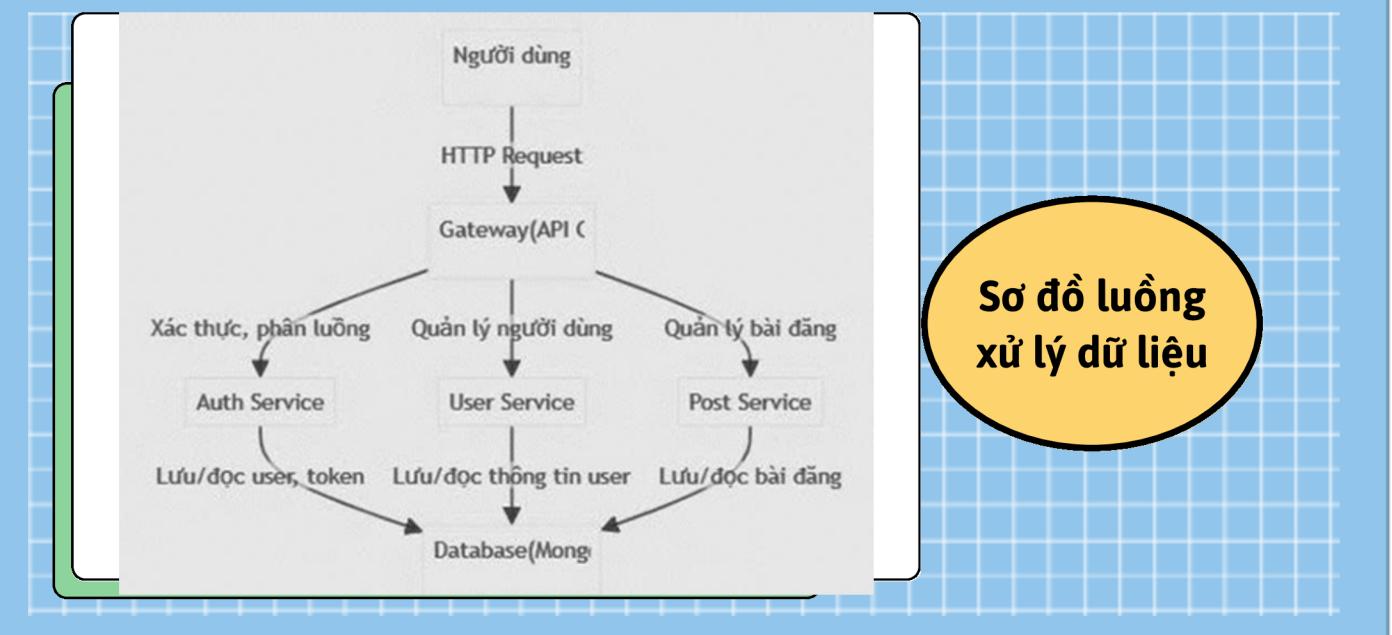
4. Thiết kế hệ thống

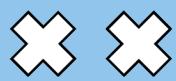


Sơ đồ
Use case

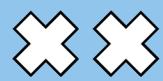


4. Thiết kế hệ thống

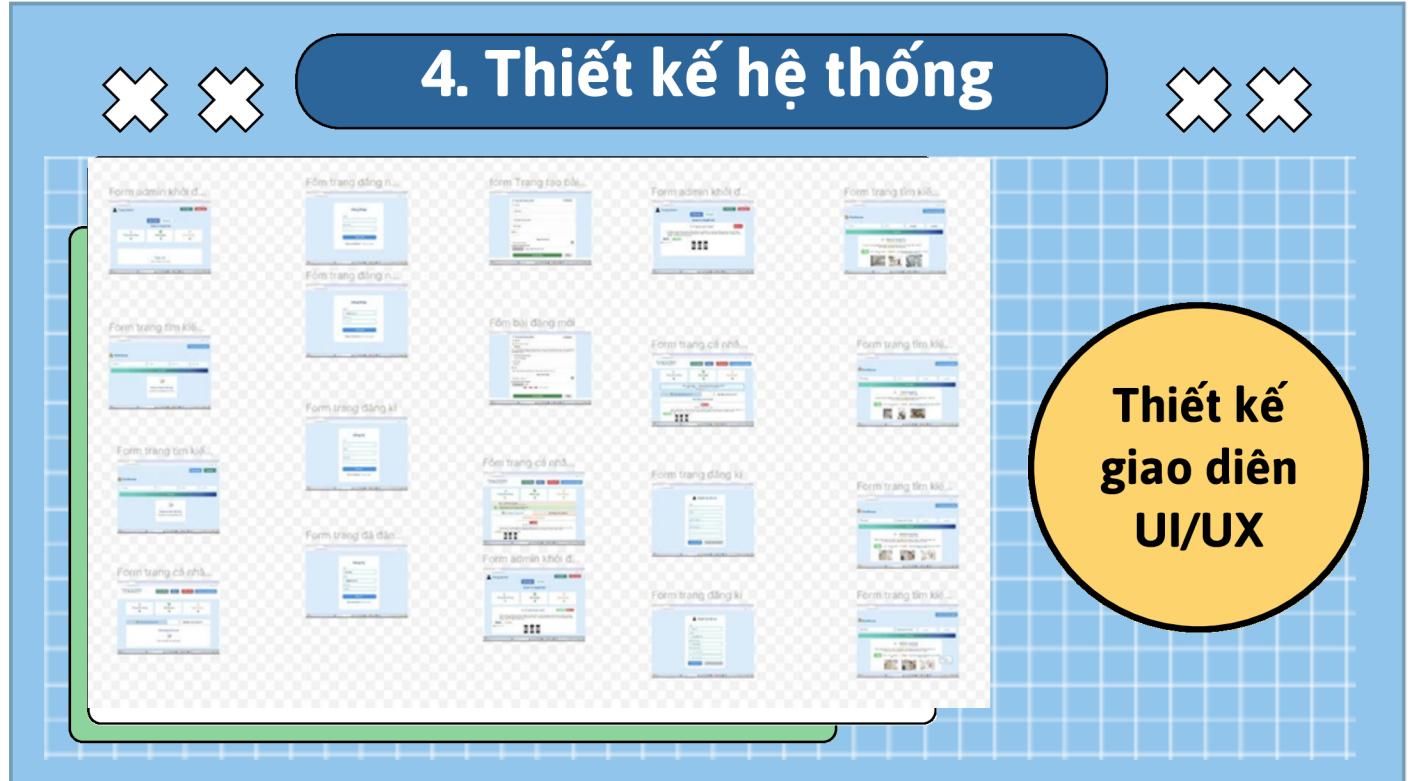
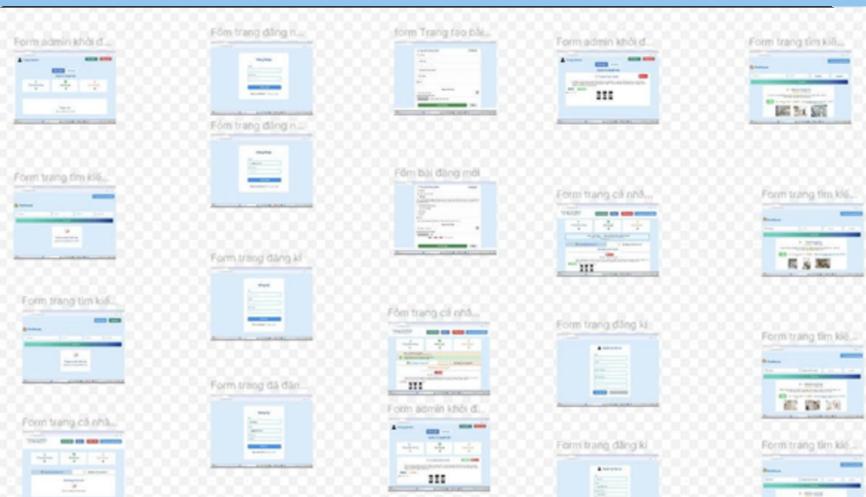




4. Thiết kế hệ thống



Thiết kế
giao diện
UI/UX



× × 5.Triển khai và công nghệ sử dụng × ×

Công nghệ

Ngôn ngữ lập trình:
TypeScript và JavaScript

Framework: Express.js

Thư viện: bcrypt.js,
jsonwebtoken, cors, dotenv,
multer

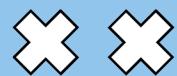
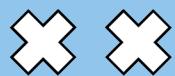
Quy trình CI/CD

- 1.Phát hiện thay đổi
- 2.Cài đặt môi trường
(Node.js, MongoDB container)
- 3.Cài đặt dependencies
(npm install)
- 4.Kiểm thử
- 5.Build ứng dụng
- 6.Triển khai tự động lên
nền tảng

Triển khai ứng dụng

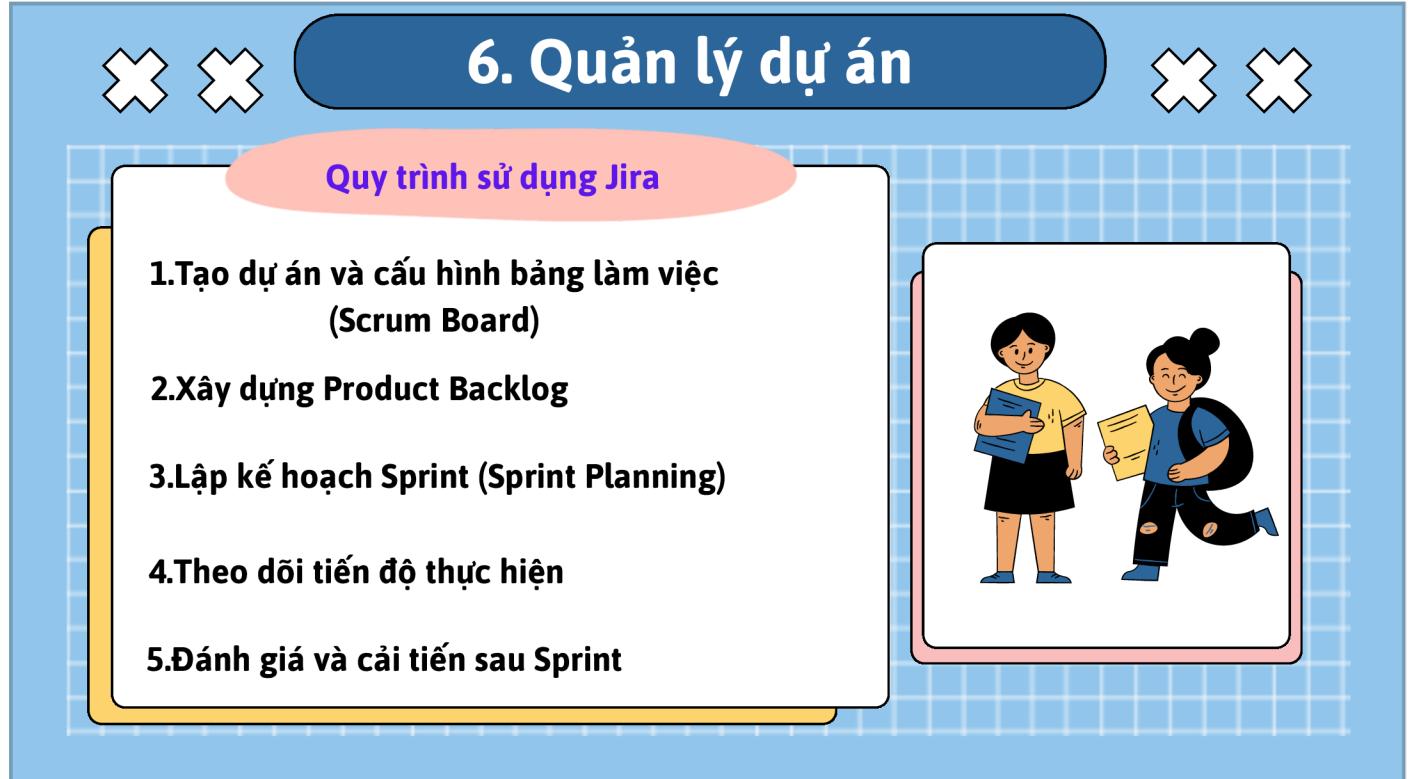
- 1.Chuẩn bị thư mục
- 2.Tạo file .env
- 3.Build và khởi động
hệ thống
- 4.Truy cập hệ thống
- 5.Dừng và dọn hệ thống

6. Quản lý dự án

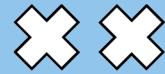
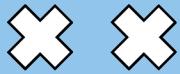


Quy trình sử dụng Jira

- 1.Tạo dự án và cấu hình bảng làm việc (Scrum Board)
- 2.Xây dựng Product Backlog
- 3.Lập kế hoạch Sprint (Sprint Planning)
- 4.Theo dõi tiến độ thực hiện
- 5.Đánh giá và cải tiến sau Sprint



7. Kiểm thử



Đăng ký người dùng

- Kiểm tra đăng ký thành công.
- Kiểm tra email đã tồn tại (kỳ vọng trả về lỗi).
- Kiểm tra định dạng email không hợp lệ.
- Kiểm tra mật khẩu trống hoặc quá ngắn.

Đăng nhập người dùng

- Đăng nhập thành công và trả về token, role.
- Đăng nhập sai mật khẩu hoặc email.
- Kiểm tra phản hồi khi gửi thiếu trường email hoặc password.

Thông tin người dùng

- Token hợp lệ -> trả về thông tin người dùng.
- Token sai hoặc thiếu -> trả về lỗi 401 Unauthorized.
- ID không tồn tại -> trả về lỗi 404 Not Found (nếu có xử lý).

7. Kiểm thử

The screenshot shows the Postman application interface. A yellow box highlights the request body and response body sections. The request body contains a JSON object with fields: name, email, and password. The response body shows a JSON object with a message indicating successful registration and a user object containing an ID and email. A green circle on the right side of the screenshot contains the text "Postman -- đăng ký tài khoản".

POST http://localhost:3000/api/auth/register

```
1 {  
2   "name": "Thành Tin",  
3   "email": "ptting@gmail.com",  
4   "password": "2222"  
5 }
```

201 Created - 79 ms - 349 B

```
{  
  "message": "Đăng ký thành công",  
  "user": {  
    "id": "6081e05445e06ae5abccdb7c",  
    "email": "ptting@gmail.com"  
  }  
}
```

**Postman --
đăng ký tài
khoản**

7. Kiểm thử

The screenshot shows a Postman interface with a blue header bar containing the title "7. Kiểm thử". Below the header, there are two large white "X" icons on the left and right sides. The main area displays a POST request to "http://localhost:3000/api/auth/login". The request body is JSON:

```
1 {
2   "email": "pttin@gmail.com",
3   "password": "2222"
4 }
```

The response status is 200 OK, with a response time of 79 ms and a response size of 501 B. The response body is:

```
1 {
2   "message": "Đăng nhập thành công",
3   "token": "eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJsb2dpbiIsInR5cCI6IkpxVC39.eyJpc3MiOiJsb2dpbiJ9.eyJleHAiOjE2NjQyOTMwNzMsImV4cCI6MTc1MzMONDI0NSeI2XhuijoxNzUzNDMnNjQ1fQ.eyJxIjoiZ2VybWQ0I0I1ZD00gXZtgINoY1ZTA2YWU1YkYjY2R1N2M1LCyb2x1IjoidXNlcjIiSImlhCI6MTc1MzMONDI0NSeI2XhuijoxNzUzNDMnNjQ1fQ",
4   "role": "user"
5 }
```

To the right of the response, there is a green circle with a black border containing the text "Đăng nhập trả về token".

7. Kiểm thử

The screenshot shows the Postman interface with a successful API call. The URL is `http://localhost:3000/api/auth/user/6881e05465e06ae5abccdb7c`. The Authorization tab is selected, showing "Bearer Token" as the auth type. A token value `JvJwouEWoFxEbk2T4ej5mek5O2PgE` is entered. The response status is `200 OK` with a `12 ms` duration and `320 B` size. The response body is a JSON object:

```
{  
  "_id": "6881e05465e06ae5abccdb7c",  
  "name": "Thành Tin",  
  "email": "pttin@gmail.com"  
}
```

A green circle on the right contains the text: **Dùng token để trả về thông tin tài khoản**.

