

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG  
KHOA CÔNG NGHỆ THÔNG TIN

NGUYỄN DUY ĐẠT

**SỬ DỤNG**  
**REACTJS, DJANGO, SQLITE XÂY**  
**DỰNG HỆ THỐNG XEM PHIM**  
**TRỰC TUYẾN**

THÁI NGUYÊN, NĂM 2024

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG  
KHOA CÔNG NGHỆ THÔNG TIN



Ảnh 3x4

# ĐỒ ÁN/KHÓA LUẬN TỐT NGHIỆP ĐẠI HỌC

Ngành: Kỹ thuật phần mềm

Đề tài:

## XÂY DỰNG HỆ THỐNG XEM PHIM TRỰC TUYẾN

Sinh viên thực hiện: NGUYỄN DUY ĐẠT

Lớp: CNTT18G                      Hệ chính quy

Giảng viên hướng dẫn: TS NGUYỄN THẾ VINH

Thái Nguyên, năm 2024

## MỤC LỤC

<b>LỜI MỞ ĐẦU .....</b>	<b>6</b>
<b>LỜI CAM ĐOAN .....</b>	<b>7</b>
<b>CHƯƠNG 1 : CƠ SỞ LÝ THUYẾT.....</b>	<b>8</b>
1.1 Tìm hiểu về reactjs.....	8
1.1.1 Tổng quát về reactjs .....	8
1.1.2 Virtual DOM .....	9
1.1.3 SPA (Single Page Application) .....	9
1.1.4 MPA (Multi-Page Application).....	9
1.1.5 Component(Thành phần).....	9
1.1.6 JSX (JavaScript XML) .....	10
1.1.7 React Hook .....	10
1.1.8 React Router .....	12
1.1.9 Ưu nhược điểm.....	12
1.2 Tìm hiểu Django .....	13
1.2.1 Tổng quát về Django .....	13
1.2.2 Tại sao lại sử dụng Django.....	13
1.2.3 Ưu nhược điểm của Django .....	16
1.2.4 Cần làm gì để làm việc với Django .....	18
1.2.5 Các nền tảng lớn đã sử dụng Django .....	18
1.2.6 các bước để tạo dự án Django .....	20
1.2.7 Django Rest Framework.....	21
1.3 Tìm hiểu về SQLite .....	22
1.3.1 SQLite là gì?.....	22
1.3.2 Tính năng của SQLite .....	23
1.3.3 Ưu điểm:.....	24
1.3.4 nhược điểm cần xem xét: .....	25
1.3.5 Sqlite trong Django .....	25
1.4 Tìm hiểu về API.....	28
1.4.1 Khái niệm .....	28
1.4.2 Ưu điểm .....	28

1.4.3	Nhược điểm .....	29
1.4.4	Ứng dụng của API .....	30
1.5	Công cụ phát triển Visual Studio Code .....	31
1.5.1	Tổng quát về Visual tudio Code.....	31
1.5.2	Các bước cài đặt Visual Studio Code.....	32
<b>CHƯƠNG 2 : KHẢO SÁT HIỆN TRẠNG .....</b>		<b>34</b>
2.1	Thực trạng và giải pháp .....	34
2.2	Xác định yêu cầu chung của hệ thống .....	34
2.3	Yêu cầu kĩ thuật.....	35
2.4	Yêu cầu chi tiết .....	35
<b>CHƯƠNG 3 : PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG .....</b>		<b>38</b>
3.1	Tác nhân.....	38
3.2	Sơ đồ Usecase .....	38
3.2.1	Usecase tổng quát.....	38
3.2.2	Usecase phân rã.....	39
3.3	Đặc tả Use case .....	41
3.3.1	Usecase Đăng kí .....	41
3.3.2	Usecase đăng nhập .....	41
3.3.3	Usecase tìm kiếm.....	42
3.3.4	Usecase xem phim.....	43
3.3.5	usecase thêm lịch sử .....	43
3.3.6	Usecase xem danh sách lịch sử xem .....	44
3.3.7	Usecase bình luận.....	44
3.3.8	Usecase quản lí phim.....	46
3.5.9	Thêm tài khoản.....	<b>Error! Bookmark not defined.</b>
3.4	Sơ đồ trình tự .....	47
3.4.1	Đăng kí .....	47
3.4.2	Đăng nhập.....	47
3.4.3	Tìm kiếm .....	48
3.4.4	Xem phim .....	49
3.4.5	Thêm lịch sử .....	49

3.4.6 xem danh sách lịch sử .....	50
3.4.7 Bình luận .....	50
3.4.8 Quản Li Phim .....	51
3.4.9 Thêm tài khoản.....	<b>Error! Bookmark not defined.</b>
3.5 Biểu đồ hoạt động.....	52
3.5.1 Đăng kí .....	52
3.5.2 Đăng nhập.....	52
3.5.3 Tìm kiếm .....	53
3.5.4 Xem Phim.....	54
3.5.5 Thêm lịch sử.....	54
3.5.6 xem danh sách lịch sử .....	55
3.5.7 Bình luận .....	55
3.5.8 Quản lí phim.....	56
3.5.9 Thêm tài khoản.....	<b>Error! Bookmark not defined.</b>
3.6 Biểu Đồ lớp.....	57
<b>CHƯƠNG 4: XÂY DỰNG HỆ THỐNG.....</b>	<b>58</b>
4.1 Trang chủ.....	58
4.2 Trang danh mục phim.....	58
4.3 Trang đăng kí.....	59
4.4 Trang đăng nhập.....	59
4.5 Trang lịch sử xem.....	60
4.6 Trang tìm kiếm .....	60
4.7 Trang thông tin chi tiết phim.....	61
4.8 Trang xem phim .....	61
4.9 Trang quản lí .....	62
<b>KẾT LUẬN.....</b>	<b>63</b>
<b>TÀI LIỆU THAM KHẢO .....</b>	<b>64</b>

## LỜI MỞ ĐẦU

Ngày nay, các ứng dụng công nghệ thông tin đã được áp dụng trong mọi lĩnh vực nghiên cứu khoa học kỹ thuật cũng như trong đời sống, máy tính trở thành công cụ trợ giúp đắc lực cho con người trong lưu trữ, phân tích và xử lý thông tin. Ở nước ta, cùng với sự phát triển chung của nền kinh tế, tin học đang ngày được ứng dụng rộng rãi trong công tác quản lý và từng bước khẳng định sức mạnh cũng như vị trí quan trọng của mình.

Trong ngành CNTT, các phần mềm được thiết kế theo nhu cầu sử dụng, có 3 mô hình lập trình phần mềm là: Mô hình máy đơn, client/server và web-based. Trong đó mô hình web-based là mô hình được nhiều ưa thích vì nó có thể cho phép user ở bất kì đâu, chỉ cần có kết nối Internet là dễ dàng truy cập vào và sử dụng. Có rất nhiều lĩnh vực ứng dụng trên nền tảng web-based như là quản lý nhân sự, thương mại điện tử, y tế, giáo dục và đào tạo, vui chơi giải trí,... Trong đó đề án này, với mục đích xây dựng một hệ thống xem phim trực tuyến, em chọn mô hình thiết kế web-based để xây dựng 1 trang web cho phép user có thể truy cập vào để xem phim trực tuyến.

Hệ thống được sử dụng các công nghệ hàng đầu hiện nay: Reactjs, Django và HTQLCS Sqlite với những tính năng cơ bản như đăng ký user, xem phim, tìm phim và các chức năng người quản lí, user khác.

## *LỜI CAM ĐOAN*

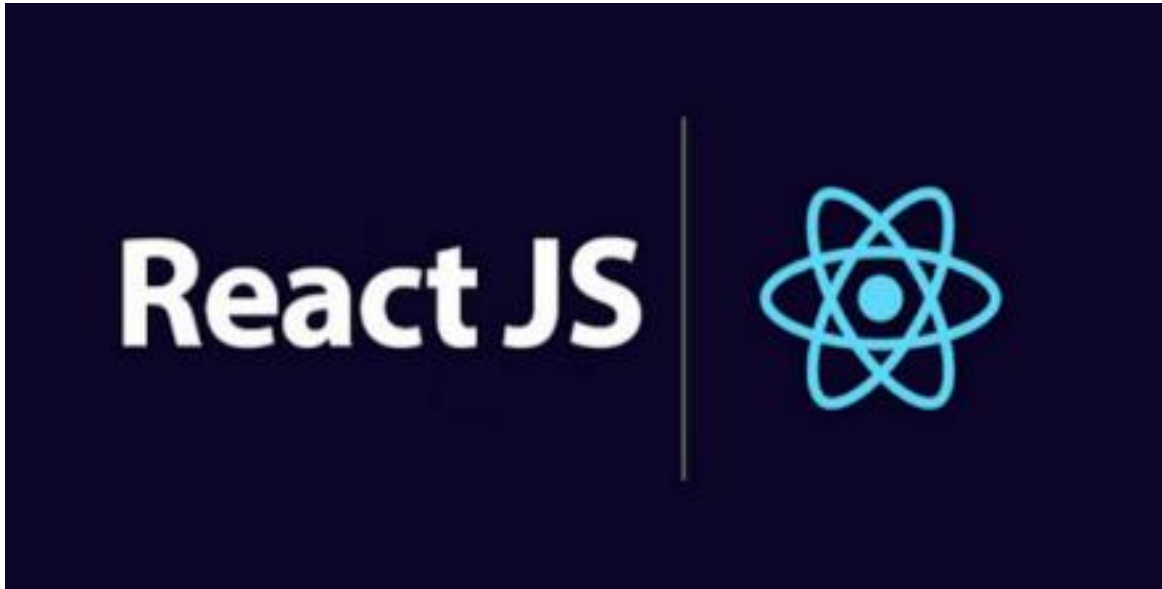
Em xin cam đoan rằng đề án này là công trình nghiên cứu của bản thân mình. Những phần có sử dụng tài liệu tham khảo có trong đề án đã được liệt kê và nêu rõ ra tại phần tài liệu tham khảo. Đồng thời những số liệu hay kết quả trình bày trong đề án đều mang tính chất trung thực.

Nếu như sai em xin chịu hoàn toàn trách nhiệm và chịu tất cả các kỷ luật của bộ môn cũng như nhà trường đề ra.

# CHƯƠNG 1 : CƠ SỞ LÝ THUYẾT

## 1.1 Tìm hiểu về reactjs

### 1.1.1 Tổng quát về reactjs



#### ❖ Định nghĩa

ReactJS là một thư viện JavaScript được phát triển bởi Facebook để xây dựng giao diện người dùng cho các ứng dụng web hiệu quả và dễ bảo trì. Nó tập trung vào việc xây dựng các thành phần giao diện người dùng (UI) có thể tái sử dụng, giúp quản lý trạng thái của ứng dụng một cách hiệu quả và tạo ra giao diện người dùng động

Một số điểm chính của ReactJS bao gồm:

- Thành phần (Component)-Based: ReactJS sử dụng mô hình lập trình dựa trên thành phần, cho phép bạn tạo ra các thành phần UI độc lập có thể tái sử dụng và tự động hóa các phần của giao diện người dùng.
- Virtual DOM: ReactJS sử dụng một cấu trúc dữ liệu gọi là Virtual DOM để cải thiện hiệu suất và tăng tốc độ render của ứng dụng. Virtual DOM giúp giảm thiểu số lượng thao tác với DOM thực tế, làm giảm thời gian render và tối ưu hóa hiệu suất.



- Quản lý Trạng thái (State) của ứng dụng: ReactJS cung cấp các công cụ cho việc quản lý trạng thái của ứng dụng một cách hiệu quả thông qua việc sử dụng trạng thái cục bộ và trạng thái toàn cục, giúp dễ dàng điều khiển và cập nhật giao diện người dùng.
- Thư viện Redux: Trong một số trường hợp, ReactJS được kết hợp với Redux, một thư viện quản lý trạng thái dựa trên Flux, để quản lý trạng thái ứng dụng một cách linh hoạt và hiệu quả hơn..

### **1.1.2 Virtual DOM**

Virtual DOM là một biểu diễn ảo của DOM được React sử dụng để tối ưu hóa hiệu suất. Thay vì cập nhật DOM trực tiếp, Reactjs tương tác với Virtual DOM. Sau khi có sự thay đổi, React so sánh Virtual DOM mới với trạng thái trước đó và chỉ cập nhật những phần thay đổi vào DOM thực tế. Điều này giúp giảm tải cho trình duyệt và tăng hiệu suất render.

### **1.1.3 SPA (Single Page Application)**

SPA là một kiến trúc ứng dụng web mà một trang web chỉ tải một lần duy nhất trong quá trình sử dụng, sau đó các nội dung khác được tải bằng cách sử dụng JavaScript thông qua AJAX

### **1.1.4 MPA (Multi-Page Application)**

MPA là một kiến trúc ứng dụng web truyền thống, mỗi lần chuyển đổi giữa các trang sẽ yêu cầu tải lại toàn bộ trang.

### **1.1.5 Component(Thành phần)**

Trong ngữ cảnh của ReactJS hoặc mô hình lập trình dựa trên thành phần, "component" (hoặc "thành phần") là một khối xây dựng UI độc lập có thể tái sử dụng. Mỗi thành phần có thể chứa mã JSX để mô tả giao diện người dùng của một phần nhỏ của ứng dụng.

Một số đặc điểm của thành phần trong ReactJS bao gồm:

- Tính độc lập: Mỗi thành phần tồn tại và hoạt động độc lập, tức là nó có thể được sử dụng lại một cách dễ dàng trong nhiều phần khác nhau của ứng dụng.
- Tính Tái sử dụng: Thành phần được thiết kế để có thể được tái sử dụng nhiều lần trong ứng dụng mà không cần phải viết mã lại.
- Tính Tách biệt: Mỗi thành phần quản lý trạng thái và hiển thị của riêng mình, giúp dễ dàng quản lý và duy trì mã nguồn.
- Tính Tích hợp: Các thành phần có thể được kết hợp với nhau để tạo thành các giao diện phức tạp bằng cách lồng nhau hoặc kết hợp với nhau.

Ví dụ về các thành phần trong ReactJS có thể bao gồm nút, thanh trượt, biểu đồ, biểu mẫu đăng nhập, hộp thoại cảnh báo, và nhiều hơn nữa. Bằng cách kết hợp các thành phần này lại với nhau, bạn có thể xây dựng các giao diện người dùng phức tạp và mạnh mẽ cho ứng dụng web của mình.

### 1.1.6 JSX (JavaScript XML)

JSX là một phần quan trọng của React, cho phép viết mã HTML trong JavaScript. JSX giúp tạo ra mã nguồn dễ đọc và dễ hiểu.

### 1.1.7 React Hook

