

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH**

**KHOA CÔNG NGHỆ THÔNG TIN**

****

**Khóa luận tốt nghiệp**

**Đề tài: Xây dựng website hỗ trợ**

**thanh tra trong các kỳ thi**

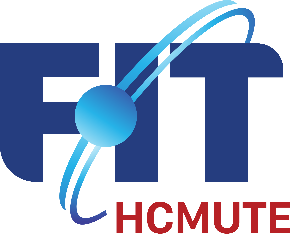
**GVHD: ThS. Nguyễn Minh Đạo**

**Nhóm SVTH:**

**20110617 - Nguyễn Thanh Danh**

**20110627 - Nguyễn Khắc Dương**

*TP. Hồ Chí Minh, ngày 6 tháng 7, năm 2024*

 **CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM**

**Độc lập – Tự do – Hạnh Phúc**

\*\*\*\*\*\*\*

**PHIẾU NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN**

Họ và tên Sinh viên:

1. Nguyễn Khắc Dương MSSV: 20110627
2. Nguyễn Thanh Danh MSSV: 20110617

Ngành: Công nghệ Thông tin

Tên đề tài: Xây dựng website hỗ trợ thanh tra trong các kỳ thi

Họ và tên Giáo viên hướng dẫn: ThS. Nguyễn Minh Đạo

**NHẬN XÉT**

1. Về nội dung đề tài & khối lượng thực hiện

1. Ưu điểm

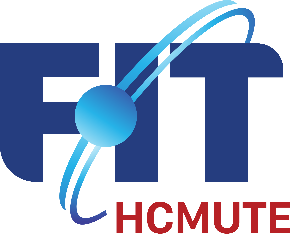
1. Khuyết điểm

1. Đề nghị cho bảo vệ hay không?
2. Đánh giá loại:
3. Điểm:

Tp*. Hồ Chí Minh, ngày tháng năm 202*

Giảng viên hướng dẫn

*(Ký & ghi rõ họ tên)*

 **CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM**

**Độc lập – Tự do – Hạnh Phúc**

\*\*\*\*\*\*\*

**PHIẾU NHẬN XÉT CỦA GIẢNG VIÊN PHẢN BIỆN**

Họ và tên Sinh viên:

1. Nguyễn Khắc Dương MSSV: 20110627

2. Nguyễn Thanh Danh MSSV: 20110617

Ngành: Công nghệ Thông tin

Tên đề tài: Xây dựng website hỗ trợ thanh tra trong các kỳ thi

Họ và tên Giáo viên phản biện: TS. Huỳnh Xuân Phụng

**NHẬN XÉT**

* + - 1. Về nội dung đề tài & khối lượng thực hiện

* + - 1. Ưu điểm

* + - 1. Khuyết điểm

1. Đề nghị cho bảo vệ hay không?
2. Đánh giá loại:
3. Điểm:

Tp*. Hồ Chí Minh, ngày tháng năm 202*

Giảng viên phản biện

*(Ký & ghi rõ họ tên)*

**LỜI CẢM ƠN**

Lời đầu tiên, chúng em xin chân thành cảm ơn Thầy Nguyễn Minh Đạo, người đã trực tiếp giảng dạy, truyền đạt những kiến thức quý báu và tận tình giúp cho nhóm chúng em trong suốt quá trình thực hiện tiểu luận.

Với hạn chế về kinh nghiệm và thời gian, tiểu luận này sẽ không thể tránh được những thiếu sót. Chúng em rất mong nhận được sự chỉ bảo, đóng góp ý kiến của thầy để bọn em có thể bổ sung và nâng cao kiến thức của mình nhằm phục vụ tốt hơn các đồ án sau này.

Xin chân thành cảm ơn.

Tp. Hồ Chí Minh, ngày 6 tháng 7 năm 2024

*Nhóm sinh viên thực hiện*

Nguyễn Khắc Dương & Nguyễn Thanh Danh

**MỤC LỤC**

[DANH MỤC HÌNH 1](#_Toc156035121)

[DANH MỤC BẢNG 4](#_Toc156035122)

[PHẦN 1: MỞ ĐẦU 5](#_Toc156035123)

[**1. Tính cấp thiết của đề tài** 5](#_Toc156035124)

[**2.** **Mục đích của đề tài** 5](#_Toc156035125)

[**3.** **Nhiệm vụ của đề tài** 5](#_Toc156035126)

[**4.** **Cách tiếp cận và phạm vi nghiên cứu** 5](#_Toc156035127)

[**5.** **Ý nghĩa khoa học và thực tiễn** 6](#_Toc156035128)

[PHẦN 2: KẾ HOẠCH THỰC HIỆN 7](#_Toc156035129)

[PHẦN 3: NỘI DUNG 8](#_Toc156035130)

[CHƯƠNG 1: CƠ SỞ LÝ THUYẾT 8](#_Toc156035131)

[**1.1.** **Các khái niệm cơ bản** 8](#_Toc156035132)

[**1.2.** **MERN Stack hoạt động như thế nào?** 16](#_Toc156035133)

[**1.3.** **Cấu trúc của một MERN Stack** 17](#_Toc156035134)

[**1.4.** **Tại sao chọn MERN Stack** 18](#_Toc156035135)

[CHƯƠNG 2: TÌM HIỂU VỀ CÁC MẠNG XÃ HỘI ĐANG HOẠT ĐỘNG 19](#_Toc156035136)

[**2.1.** **Facebook** 19](#_Toc156035137)

[**2.2.** **Twitter**  21](#_Toc156035138)

[**2.3.** **Instagram** 22](#_Toc156035139)

[**2.4.** **Tiktok** 23](#_Toc156035140)

[CHƯƠNG 3: MÔ TẢ PHẦN MỀM 25](#_Toc156035141)

[**3.1. Đặc tả đề tài** 25](#_Toc156035142)

[**3.2.** **Mô hình Usecase** 26](#_Toc156035143)

[**3.3.** **Thiết kế cơ sở dữ liệu** 43](#_Toc156035144)

[**3.4.** **Phân tích thiết kế các thành phần chức năng** 52](#_Toc156035145)

[**3.5.** **Activity Diagram** 59](#_Toc156035146)

[**3.6.** **Phân tích, thiết kế giao diện và chức năng của giao diện** 65](#_Toc156035147)

[CHƯƠNG 4: KIỂM THỬ PHẦN MỀM 76](#_Toc156035148)

[**4.1. Một số test case** 76](#_Toc156035149)

[**4.2. Test report** 80](#_Toc156035150)

[**4.3. Bug report** 81](#_Toc156035215)

[CHƯƠNG 5: CÀI ĐẶT DEMO 89](#_Toc156035216)

[**5.1. Các yêu cầu về môi trường cài đặt và phiên bản phần mềm** 89](#_Toc156035217)

[**5.2. Cài đặt Server** 89](#_Toc156035218)

[**5.3. Cài đặt Client** 90](#_Toc156035219)

[**5.4. Sử dụng Docker để deploy trên môi trường production** 91](#_Toc156035220)

[PHẦN 3: KẾT LUẬN 92](#_Toc156035221)

[**1.** **Kết quả đạt được** 92](#_Toc156035222)

[**2.** **Danh sách các chức năng đã làm được** 92](#_Toc156035223)

[**3.** **Ưu điểm và khuyết điểm** 93](#_Toc156035224)

[**4.** **Hướng phát triển** 93](#_Toc156035225)

[DANH MỤC TÀI LIỆU THAM KHẢO 94](#_Toc156035226)

# **DANH MỤC HÌNH**

[Hình 1: Sơ đồ Gantt 6](#_Toc156384120)

[Hình 2: Giới thiệu NodeJs 7](#_Toc156384121)

[Hình 3: Kiến trúc MERN Stack 15](#_Toc156384122)

[Hình 4: Usecase tổng quan 26](#_Toc156384123)

[Hình 5: Usecase đăng ký 27](#_Toc156384124)

[Hình 6: Usecase đăng nhập 27](#_Toc156384125)

[Hình 7: Usecase quản lý bài viết 27](#_Toc156384126)

[Hình 8: Usecase thống kê 28](#_Toc156384127)

[Hình 9: Use case quản lý người dùng 28](#_Toc156384128)

[Hình 10: Usecase thông báo 29](#_Toc156384129)

[Hình 11: Usecase kết nối người dùng 29](#_Toc156384130)

[Hình 12: Usecase quản lý nhóm 29](#_Toc156384131)

[Hình 13: Danh sách Collections 42](#_Toc156384132)

[Hình 14: Class Diagram 50](#_Toc156384133)

[Hình 15: Sequence Diagram Đăng nhập 51](#_Toc156384134)

[Hình 16: Sequence Diagram Đăng ký 51](#_Toc156384135)

[Hình 17: Sequence Diagram Tìm kiếm 52](#_Toc156384136)

[Hình 18: Sequence Diagram Đăng bài viết 52](#_Toc156384137)

[Hình 19: Sequence Diagram Tương tác bài viết 53](#_Toc156384138)

[Hình 20: Sequence Diagram Kết bạn 53](#_Toc156384139)

[Hình 21: Sequence Diagram Nhắn tin 54](#_Toc156384140)

[Hình 22: Sequence Diagram Thống kê 55](#_Toc156384141)

[Hình 23: Sequence Diagram Quản lý người dùng 55](#_Toc156384142)

[Hình 24: Sequence Diagram Quản lý bài viết 56](#_Toc156384143)

[Hình 25: Sequence Diagram Báo cáo bài viết 56](#_Toc156384144)

[Hình 26: Sequence Diagram Báo cáo người dùng 57](#_Toc156384145)

[Hình 27: Sequence Diagram Quản lý nhóm 57](#_Toc156384146)

[Hình 28: Activity Diagram Đăng kí 58](#_Toc156384147)

[Hình 29: Activity Đăng nhập 58](#_Toc156384148)

[Hình 30: Activity Diagram Bài viết 59](#_Toc156384149)

[Hình 31: Activity Diagram Đăng bài viết 59](#_Toc156384150)

[Hình 32: Activity Diagram Trang cá nhân 60](#_Toc156384151)

[Hình 33: Activity Diagram Nhắn tin 60](#_Toc156384152)

[Hình 34: Activity Diagram Nhóm cộng đồng 61](#_Toc156384153)

[Hình 35: Activity Diagram Thông báo 61](#_Toc156384154)

[Hình 36: Activity Diagram Tìm kiếm 62](#_Toc156384155)

[Hình 37: Activity Diagram Thống kê 62](#_Toc156384156)

[Hình 38: Activity Diagram Quản lý bài viết 63](#_Toc156384157)

[Hình 39: Activity Diagram Quản lý người dùng 63](#_Toc156384158)

[Hình 40: Giao diện Đăng nhập Admin 64](#_Toc156384159)

[Hình 41: Giao diện Dashboard 64](#_Toc156384160)

[Hình 42: Giao diện Quản lý người dùng 65](#_Toc156384161)

[Hình 43: Giao diện quản lý bài viết 65](#_Toc156384162)

[Hình 44: Giao diện Đăng nhập người dùng 66](#_Toc156384163)

[Hình 45: Giao diện đăng ký 66](#_Toc156384164)

[Hình 46: Giao diện Trang chủ 67](#_Toc156384165)

[Hình 47: Giao diện Bài viết 67](#_Toc156384166)

[Hình 48: Giao diện Trang cá nhân 68](#_Toc156384167)

[Hình 49: Giao diện Friends, Requests 68](#_Toc156384168)

[Hình 50: Giao diện Trang cá nhân người dùng khác 69](#_Toc156384169)

[Hình 51: Giao diện Tìm kiếm 69](#_Toc156384170)

[Hình 52: Giao diện Setting 70](#_Toc156384171)

[Hình 53: Giao diện Thông báo 70](#_Toc156384172)

[Hình 54: Giao diện Chọn ảnh cho bài viết 71](#_Toc156384173)

[Hình 55: Giao diện Tạo bài viết với caption 71](#_Toc156384174)

[Hình 56: Giao diện Tin nhắn 72](#_Toc156384175)

[Hình 57: Giao diện Group 72](#_Toc156384176)

[Hình 58: Giao diện group của mình 73](#_Toc156384177)

[Hình 59: Giao diện phê duyệt bài viết 73](#_Toc156384178)

[Hình 60: Giao diện group mình tham gia 74](#_Toc156384179)

[Hình 61: Giao diện modal members, lời mời, requests 74](#_Toc156384180)

[Hình 62: Testcase Đăng ký 75](#_Toc156384181)

[Hình 63: Testcase Đăng nhập 75](#_Toc156384182)

[Hình 64: Testcase Đăng bài viết 76](#_Toc156384183)

[Hình 65: Testcase Chỉnh sửa thông tin cá nhân 76](#_Toc156384184)

[Hình 66: Testcase Nhắn tin 77](#_Toc156384185)

[Hình 67: Testcase Kết bạn 77](#_Toc156384186)

[Hình 68: Testcase Bình luận 78](#_Toc156384187)

[Hình 69: Testcase Đổi mật khẩu 78](#_Toc156384188)

[Hình 70: Hệ thống quản lý bug Jira 80](#_Toc156384189)

[Hình 71: Bug Đăng kí bằng các kí tự khoảng trắng 82](#_Toc156384190)

[Hình 72: Bug Nhấn nhiều lần nút đăng kí gửi nhiều mã OTP 82](#_Toc156384191)

[Hình 73: Bug Sai định dạng file đăng bài viết 83](#_Toc156384192)

[Hình 74: Bug Sai định dạng file hình đại diện 83](#_Toc156384193)

[Hình 75: Bug Chỉnh sửa bio trên trang cá nhân không có thông báo 84](#_Toc156384194)

[Hình 76: Bug Chỉnh sửa thông tin cá nhân về cài đặt thông báo 84](#_Toc156384195)

[Hình 77: Bug Nhanh tay nhấn vào tin nhắn 85](#_Toc156384196)

[Hình 78: Bug Gửi tin nhắn chỉ có ký tự khoảng cách 85](#_Toc156384197)

[Hình 79: Bug Cập nhật lại số bạn bè và danh sách bạn bè 86](#_Toc156384198)

[Hình 80: Bug Thay đổi mật khẩu với mật khẩu mới giống mật khẩu hiện tại 86](#_Toc156384199)

[Hình 81: Bug Thay đổi mật khẩu với mật khẩu mới có ít kí tự 87](#_Toc156384200)

[Hình 82: Bug Thông báo khi bình luận bài viết của chính mình 87](#_Toc156384201)

[Hình 83: Tổ chức thư mục, file Server 88](#_Toc156384202)

[Hình 84: Tổ chức thư mục, file Client 89](#_Toc156384203)

[Hình 85: Docker Deploy 90](#_Toc156384204)

# **DANH MỤC BẢNG**

[Bảng 1: Bảng kế hoạch thực hiện 5](#_Toc154161360)

[Bảng 2: Một số câu lệnh cơ bản trên MongoDB 12](#_Toc154161361)

[Bảng 3: Chức năng của Website 24](#_Toc154161362)

[Bảng 4: Tác nhân, Fuction và Usecase 25](#_Toc154161363)

[Bảng 5: Usecase Diagram Đăng ký 30](#_Toc154161364)

[Bảng 6: Usecase Diagram Đăng nhập 30](#_Toc154161365)

[Bảng 7: Usecase Diagram Quản lý người dùng User 32](#_Toc154161366)

[Bảng 8: Usecase Diagram Quản lý người dùng Admin 33](#_Toc154161367)

[Bảng 9: Usecase Diagram Quản lý bài viết User 35](#_Toc154161368)

[Bảng 10: Usecase Diagram Quản lý bài viết Admin 36](#_Toc154161369)

[Bảng 11: Usecase Diagram Thống kê 36](#_Toc154161370)

[Bảng 12: Usecase Diagram Nhắn tin 37](#_Toc154161371)

[Bảng 13: Usecase Diagram Bạn bè 38](#_Toc154161372)

[Bảng 14: Usecase Diagram Tìm kiếm người dùng 39](#_Toc154161373)

[Bảng 15: Usecase Diagram Thông báo 40](#_Toc154161374)

[Bảng 16: Usecase Diagram Quản lý nhóm 41](#_Toc154161375)

[Bảng 17: Bảng Users 43](#_Toc154161376)

[Bảng 18: Bảng Conversations 43](#_Toc154161377)

[Bảng 19: Bảng Messages 44](#_Toc154161378)

[Bảng 20: Bảng Notifications 44](#_Toc154161379)

[Bảng 21: Bảng Posts 45](#_Toc154161380)

[Bảng 22: Bảng Reacts 46](#_Toc154161381)

[Bảng 23: Bảng Comments 46](#_Toc154161382)

[Bảng 24: Bảng Report users 47](#_Toc154161383)

[Bảng 25: Bảng Report posts 47](#_Toc154161384)

[Bảng 26: Bảng Groups 48](#_Toc154161385)

[Bảng 27: Bảng UserToGroup 48](#_Toc154161386)

[Bảng 28: Tổng hợp các testcase 64](#_Toc154161387)

# **PHẦN 1: MỞ ĐẦU**

1. **Tính cấp thiết của đề tài**

Lĩnh vực giáo dục là lĩnh vực luôn được quan tâm từ xưa đến nay đặc biệt là các kỳ thi dùng để đánh giá năng lực của chúng ta. Trong các kỳ thi lớn, việc quản lý và theo dõi thí sinh trở nên rất phức tạp, đòi hỏi nhiều công sức và nhân lực của các giám thị và thanh tra. Vì sự quan trọng của các kỳ thi mà nhiều thí sinh sử dụng nhiều hình thức gian lận như sử dụng tài liệu trong khi thi, nhờ người để thi hộ,.... Cho đến ngày nay việc gian lận này vẫn tồn tại và ngày càng tinh vi hơn để qua mắt cán bộ coi thi làm ảnh hưởng đến kết quả thi thực tế của nhiều thí sinh.

Trong bối cảnh xã hội ngày càng phát triển, việc quản lý thi sinh và đảm bảo công bằng cho kỳ thi ngày càng quan trọng. Vì những lý do trên, đề tài “Xây dựng website hỗ trợ thanh tra trong các kỳ thi” được chọn làm đề tài cho khóa luận của nhóm em.

1. **Mục đích của đề tài**

Mục đích chính của đề tài là vận dụng những kiến thức nền tảng đã học để tìm hiểu những công nghệ mới như MERN stack để phát triển trang web hỗ trợ thanh tra trong các kỳ thi, cụ thể được nói đến là tính năng điểm danh, quản lý lịch thi và báo cáo các sự cố hay các biên bản của phòng thi.

1. **Nhiệm vụ của đề tài**

Các nhiệm vụ chính cần thực hiện trong đề tài:

* Trình bày cơ sở lý thuyết về các công nghệ sử dụng.
* Tìm hiểu, phân tích và thiết kế hệ thống của một website hỗ trợ cho thanh tra trong các kỳ thi.
* Phân tích các tính năng mới và áp dụng vào đề tài.

1. **Cách tiếp cận và phạm vi nghiên cứu**

Đối tượng nghiên cứu đề tài là nhu cầu trong công tác kiểm tra của thanh tra, Phòng đào tạo và các hình thức gian lận của thí sinh trong các kỳ thi.

Bên cạnh đó ta còn nghiên cứu những công nghệ, những tính năng và cách thức và áp dụng các công nghệ hiện đại như nhận diện khuôn mặt, điểm danh tự động, quản lý phòng thi, đảm bảo tính an toàn, chính xác và dễ sử dụng.

1. **Ý nghĩa khoa học và thực tiễn**

Về khía cạnh khoa học, đề tài sẽ giúp ta hiểu được công nghệ MERN stack cũng như ứng dụng công nghệ hiện đại như nhận diện khuôn mặt, điểm danh tự động vào quản lý và thanh tra thi cử, mở ra hướng phát triển mới cho lĩnh vực này, bên cạnh đó báo cáo đóng góp vào việc số hóa, hiện đại hóa công tác tổ chức và quản lý các kỳ thi.

Về giá trị thực tiễn, đề tài sẽ nêu chi tiết cách hoạt động cũng như quy trình và công nghệ tạo ra nó giúp tăng cường hiệu quả, tính minh bạch và công bằng trong các kỳ thi thông qua việc ứng dụng các công nghệ hiện đại. Hỗ trợ đội ngũ thanh tra, giám sát thi thực hiện nhiệm vụ một cách dễ dàng, chính xác hơn.

# **PHẦN 2: KẾ HOẠCH THỰC HIỆN**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **THỜI GIAN** | **CÔNG VIỆC** | **HOÀN THÀNH** |
| 1 | 15/09/2023 - 25/09/2023 | Lên kế hoạch thực hiện | 21/09/2023 |
| 2 | 26/09/2023 - 10/10/2023 | Tìm hiểu công nghệ | 5/10/2023 |
| 3 | 11/10/2023 - 18/10/2023 | Thiết kế và xây dựng database | 13/10/2023 |
| 4 | 19/10/2023 - 30/10/2023 | Thiết kế giao diện | 2/11/2023 |
| 5 | 1/11/2023 - 19/11/2023 | Thiết kế và xây dựng APIs | 15/11/2023 |
| 6 | 20/11/2023 - 5/12/2023 | Tích hợp APIs backend với frontend | 7/12/2023 |
| 7 | 6/12/2023 - 12/12/2023 | Kiểm thử và sửa lỗi | 17/12/2023 |
| 8 | 13/12/2023 - 15/12/2023 | Hoàn thiện báo cáo | 20/12/2023 |

Bảng 1: Bảng kế hoạch thực hiện

Hình 1: Sơ đồ Gantt

# **PHẦN 3: NỘI DUNG**

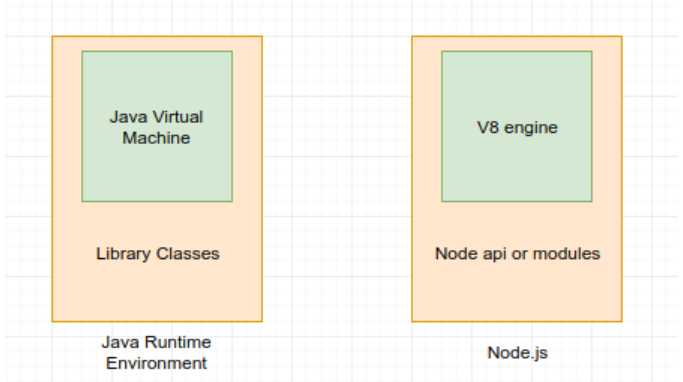
# **CHƯƠNG 1: CƠ SỞ LÝ THUYẾT**

* 1. **Các khái niệm cơ bản**
     1. MERN Stack là gì?

MERN stack là sự kết hợp của các công nghệ phổ biến nhất hiện nay và các công nghệ này đều liên quan đến JavaScript như MongoDB, ExpressJS, ReactJS, NodeJS. Lập trình viên thường dùng MERN stack để xây dựng ứng dụng React Universal.

* + 1. NodeJS là gì?
       1. *Giới thiệu NodeJS*

NodeJS là một môi trường runtime chạy JavaScript đa nền tảng và có mã nguồn mở, được sử dụng để chạy các ứng dụng web bên ngoài trình duyệt của client. Nền tảng này được phát triển bởi Ryan Dahl vào năm 2009, được xem là một giải pháp hoàn hảo cho các ứng dụng sử dụng nhiều dữ liệu nhờ vào mô hình hướng sự kiện (event-driven) không đồng bộ.



Hình 2: Giới thiệu NodeJs

* + - 1. *Lịch sử ra đời*

Năm 2009, NodeJS được viết bởi lập trình viên Ryan Dahl. Node Js được tạo lần đầu cho hệ điều hành Linux sử dụng. Nó được phát triển vào bảo trì bởi Ryan Dahl và được tài trợ bởi Joyent.

Tháng 1 năm 2010, trình quản lý gói npm cho NodeJS được ra mắt.

Tháng 6 năm 2011, phiên bản Windows gốc của NodeJS được triển khai bởi Microsoft và Joyent.

Tháng 12 năm 2014, do xung đột nội bộ nên NodeJS bị chia rẽ, IO.js được hình thành.

Tháng 6 năm 2015, cộng đồng IO.js và cộng đồng NodeJS đã thống nhất bỏ phiếu để hợp tác với nhau dưới dạng NodeJS Foundation.

* + - 1. *Ưu và nhược điểm*

Ưu điểm:

* Node.js sử dụng JavaScript cả trên server-side và client-side, giúp đơn giản hóa quá trình phát triển và chia sẻ mã nguồn.
* Hiệu suất cao: Node.js sử dụng kiến trúc non-blocking I/O, giúp tối ưu hóa hiệu suất cho các ứng dụng xử lý nhiều kết nối đồng thời.
* Môi trường thời gian thực: Node.js thích hợp cho các ứng dụng yêu cầu xử lý thời gian thực, như trò chơi trực tuyến, ứng dụng trò chơi, và chat trực tuyến.
* Nhanh chóng xây dựng MVP (Minimum Viable Product): Node.js giúp tạo ra các sản phẩm MVP một cách nhanh chóng, giúp các doanh nghiệp đưa sản phẩm ra thị trường nhanh hơn.

Nhược điểm:

* Khả năng quản lý dự án: Với ứng dụng lớn và phức tạp, quản lý mã nguồn có thể trở nên khó khăn nếu không có cấu trúc dự án tốt.
* Không thích hợp cho các ứng dụng CPU-bound: Node.js không phải là lựa chọn tốt cho các ứng dụng đòi hỏi nhiều xử lý CPU, như các thuật toán phức tạp và tính toán.
  + - 1. *Ứng dụng trong đề tài*
* Hệ thống Notification: Thông báo cho người dùng tương tự các mạng xã hội hiện hành như Facebook hay Instagram.
* WebSocket server: Cung cấp giao thức giao tiếp hai chiều mạnh mẽ do có độ trễ thấp và dễ dàng xử lý khi xảy ra lỗi. WebSocket được sử dụng cho những trường hợp yêu cầu thời gian thực như trong trò chuyện.
* Ứng dụng truy vấn tới cơ sở dữ liệu NoSQL: Với cơ sở dữ liệu MongoDB thì NodeJS là sự lựa chọn hoàn hảo vì nó có nhiều thư viện hỗ trợ mạnh.
  + 1. ExpressJS là gì?
       1. *Giới thiệu ExpressJS*

Expressjs là một framework được xây dựng trên nền tảng của Nodejs. Nó cung cấp các tính năng mạnh mẽ để phát triển web hoặc mobile. Expressjs hỗ trợ các method HTTP và middleware tạo ra API vô cùng mạnh mẽ và dễ sử dụng.

* + - 1. *Lịch sử phát triển*

Năm 2009, Express được tạo ra bởi TJ Holowaychuk như một framework nhỏ gọn để xây dựng ứng dụng web trong Node.js. Ban đầu, Express chỉ là một phần mở rộng của Connect, một middleware framework cho Node.js.

Năm 2010, Phiên bản đầu tiên của Express (Express 1.0) được phát hành. Với Express, người phát triển có thể dễ dàng xây dựng và quản lý các tuyến đường (routes) và middleware trong ứng dụng web.

Năm 2014, Express 4.0 được phát hành. Đây là phiên bản mang lại nhiều thay đổi lớn và cải tiến so với phiên bản trước đó. Express 4.0 đã tách riêng phần middleware của Connect và tích hợp nó vào Express, giúp tăng tính linh hoạt và hiệu suất của framework.

Từ năm 2014 đến nay, Express tiếp tục phát triển và trở thành một trong những framework phát triển web phổ biến nhất trong cộng đồng Node.js. Nó có một cộng đồng lớn và năng động, cung cấp rất nhiều các module và middleware hữu ích cho việc xây dựng ứng dụng web.

* + - 1. *Ưu và nhược điểm*

Ưu điểm

* Đơn giản và dễ sử dụng: ExpressJS có cú pháp đơn giản và dễ hiểu, giúp lập trình viên dễ dàng nắm bắt và triển khai các tính năng.
* Linh hoạt: ExpressJS không áp đặt một cấu trúc cụ thể, cho phép lập trình viên tự do tùy chỉnh và xây dựng ứng dụng theo ý muốn.
* Hiệu suất cao: ExpressJS được xây dựng trên Node.js, nền tảng có hiệu suất cao, cho phép xử lý nhanh chóng các yêu cầu web đồng thời và có khả năng mở rộng tốt
* Hỗ trợ định tuyến và middleware: ExpressJS giúp việc xây dựng các thành phần định tuyến ít tốn thời gian nhưng lại hiệu quả và đơn giản hơn, cung cấp hệ thống middleware mạnh mẽ, cho phép thực hiện các chức năng như xác thực, ghi log, nén dữ liệu và xử lý lỗi một cách linh hoạt và dễ dàng.

Nhược điểm:

* Thiếu cấu trúc: Do ExpressJS không áp đặt một cấu trúc nghiêm ngặt, việc tổ chức dự án và quản lý mã nguồn có thể trở nên khó khăn, đặc biệt khi ứng dụng phát triển lớn và phức tạp.
* Khả năng mở rộng: Khi ứng dụng phát triển lớn và phức tạp, việc quản lý mã nguồn và mở rộng có thể trở nên khó khăn với ExpressJS. Cần có sự kiểm soát cẩn thận để tránh sự phức tạp và rối rắm trong việc quản lý các module và tương tác giữa chúng
  + - 1. *Ứng dụng trong để tài*

Các chức năng của ExpressJS được sử dụng trong đề tài:

* Cấu hình các lớp trung gian để trả về các HTTP request.
* Cho phép sử dụng với các hành động khác nhau dựa trên các phương thức HTTP và URL bằng define router.
* Trả về các trang HTML dựa vào các tham số.
  + 1. ReactJS là gì?
       1. *Giới thiệu ReactJS*

React (còn được gọi là Reactjs hay React.js) là một Thư viện javascript được tạo ra bởi sự cộng tác giữa Facebook và Instagram. Nó cho phép những nhà phát triển web tạo ra giao diện người dùng nhanh chóng.

Phần Views của Reactjs thường được hiển thị bằng việc chủ yếu dung các component mà chứa các component cụ thể hoặc các thẻ HTML. Một trong những đặc trưng duy nhất của Reactjs là việc render dữ liệu không những có thể thực hiện ở tầng server mà còn ở tầng client.

* + - 1. *Lịch sử phát triển*

ReactJS được hiểu nôm na là một thư viện trong đó có chứa nhiều JavaScript mã nguồn mở. Mục đích của việc tạo ReactJS là tạo ra các ứng dụng web nhanh, hiệu quả và viết ít code.

Năm 2011, Jordan Walke, một nhân viên của Facebook, phát triển ReactJS ban đầu để giải quyết các vấn đề về hiệu suất và khả năng mở rộng trong việc xây dựng giao diện người dùng phức tạp của Facebook.

Năm 2013, ReactJS được công bố công khai lần đầu tiên tại hội nghị F8 của Facebook. Nó ban đầu được giới thiệu là một "thư viện giao diện người dùng" và nhanh chóng thu hút sự quan tâm của cộng đồng phát triển web.

Năm 2015, ReactJS trở thành mã nguồn mở hoàn toàn, cho phép các nhà phát triển khác cùng đóng góp vào việc phát triển và cải tiến ReactJS. Điều này tạo ra sự tăng trưởng nhanh chóng của cộng đồng React và sự phát triển của các tính năng và công cụ xung quanh nó.

Năm 2016, React Native, một framework cho phép xây dựng ứng dụng di động đa nền tảng bằng ReactJS, được giới thiệu. Điều này giúp mở rộng phạm vi ứng dụng của ReactJS từ web sang các nền tảng di động.

Từ năm 2016 đến nay, ReactJS tiếp tục phát triển và trở thành một trong những công nghệ phát triển giao diện người dùng phổ biến nhất. Nó đã trở thành một phần của nhiều dự án lớn và được sử dụng bởi nhiều công ty công nghệ hàng đầu trên thế giới.

* + - 1. *Ưu và nhược điểm*

Ưu điểm:

* Dễ sử dụng: Thư viện react chỉ tập trung vào việc tạo ra UI hiệu quả và được phân loại thành kiểu “V” trong mô hình MVC (Model-View-Controller).
* Component-Based: React sử dụng mô hình component, giúp tạo ra mã nguồn dễ hiểu, tái sử dụng và bảo trì. Components có thể được tái sử dụng ở nhiều nơi trong ứng dụng hoặc giữa các dự án khác nhau.
* Hiệu suất cao: React sử dụng Virtual DOM để tối ưu hóa hiệu suất bằng cách cập nhật chỉ các phần tử có thay đổi, không cần phải làm mới toàn bộ DOM. Điều này giúp giảm độ trễ và tăng tốc quá trình render.

Nhược điểm:

* ReactJS chỉ phục vụ cho tầng View do chỉ là View Library (không phải MVC framework). Đo dó React sẽ không có phần Model và Controller, 2-way binding hay là Ajax.
* Poor Document: Đó là một nhược điểm khá phổ biến đối với các công nghệ cập nhật liên tục. Các công nghệ cập nhật và tăng tốc nhanh đến mức không có thời gian để tạo tài liệu phù hợp. Để khắc phục điều này, các nhà phát triển tự viết hướng dẫn với việc phát triển các bản phát hành và công cụ mới trong các dự án hiện tại của họ.
* Khó tiếp cận cho người mới học.
  + - 1. *Ứng dụng trong đề tài*
* ReactJS kết hợp với Redux để tạo nên front end.
* Redux: Là một thành phần quan trọng trong react, được dùng để quản lí các state.
  + 1. MongoDB là gì?
       1. *NoSQL là gì?*

NoSQL là một thuật ngữ viết tắt của "Not Only SQL" (không chỉ SQL) và thường được sử dụng để chỉ các loại hệ quản trị cơ sở dữ liệu không phụ thuộc vào mô hình quan hệ SQL truyền thống. NoSQL là một phương pháp lưu trữ và truy xuất dữ liệu phổ biến trong các ứng dụng web và hệ thống có yêu cầu lưu trữ lớn và khả năng mở rộng cao.

Trong khi hệ thống SQL truyền thống sử dụng các bảng, hàng và cột để lưu trữ dữ liệu theo mô hình quan hệ, NoSQL sử dụng các cơ chế lưu trữ dữ liệu không có cấu trúc hoặc cấu trúc linh hoạt hơn. Các hệ thống NoSQL thường dựa trên các mô hình dữ liệu như key-value, document, columnar hoặc graph.

* + - 1. *MongoDB*

MongoDB là một hệ quản trị cơ sở dữ liệu phi quan hệ (NoSQL) phổ biến. Nó được phát triển bởi công ty MongoDB Inc. và là một phần của họ các công nghệ cơ sở dữ liệu NoSQL. MongoDB được thiết kế để lưu trữ và xử lý các tài liệu theo mô hình cơ sở dữ liệu tài liệu (document database).

Một tài liệu trong MongoDB là một bản ghi dưới dạng JSON-like (JavaScript Object Notation), được gọi là BSON (Binary JSON). Tài liệu này có thể chứa các trường và giá trị khác nhau, cho phép linh hoạt trong việc lưu trữ dữ liệu không đồng nhất. Điều này rất hữu ích trong các ứng dụng web và di động, nơi cấu trúc dữ liệu có thể thay đổi linh hoạt theo thời gian.

*c. Một số câu lệnh cơ bản trong MongoDB*

|  |  |  |
| --- | --- | --- |
| CSDL | MySQL | MongoDB |
| Tạo csdl | CREATE DATABASE test; | use test; |
| Tạo bảng | CREATE TABLE students (ten\_cot - kieu\_du\_lieu); | db.createCollection('students'); |
| Tạo bản ghi | INSERT INTO students ('name', 'gender') VALUES (name, 'male'); | db.students.insert ({name:name, gender: 'male'}); |
| Cập nhật | UPDATE students SET name = 'name update' WHERE id = 1; | db.students.update ({\_id: 1},  {$set:{ name: 'name update' }}); |
| Xóa bản ghi | DELETE FROM students Where id = 1; | db.students.remove ({\_id: 1}); |
| Tìm kiếm all | SELECT \* FROM students; | db.students.find ({}); |
| Tìm kiếm | SELECT \* FROM students WHERE name = 'name'; | db.students.find ({name: 'name' }); |

Bảng 2: Một số câu lệnh cơ bản trên MongoDB

1. *Ưu và nhược điểm*

Ưu điểm:

* Linh hoạt và không đồng nhất: MongoDB cho phép lưu trữ dữ liệu không đồng nhất, giúp linh hoạt trong việc lưu trữ dữ liệu có cấu trúc linh hoạt hoặc thay đổi theo thời gian.
* Khả năng mở rộng ngang: MongoDB có khả năng mở rộng ngang tự nhiên, cho phép thêm các nút máy chủ mới vào cụm để tăng khả năng xử lý dữ liệu và khả năng chịu tải của hệ thống.
* Tích hợp tốt với các ngôn ngữ lập trình: MongoDB hỗ trợ các thư viện và trình điều khiển cho nhiều ngôn ngữ lập trình phổ biến, giúp phát triển ứng dụng dễ dàng trên các nền tảng khác nhau.
* Truy vấn phong phú: MongoDB cung cấp một ngôn ngữ truy vấn mạnh mẽ và linh hoạt, cho phép truy vấn dữ liệu phức tạp và tìm kiếm văn bản đầy đủ.
* Hiệu năng cao: MongoDB có tốc độ truy vấn cao hơn cơ sở dữ liệu quan hệ.

Nhược điểm:

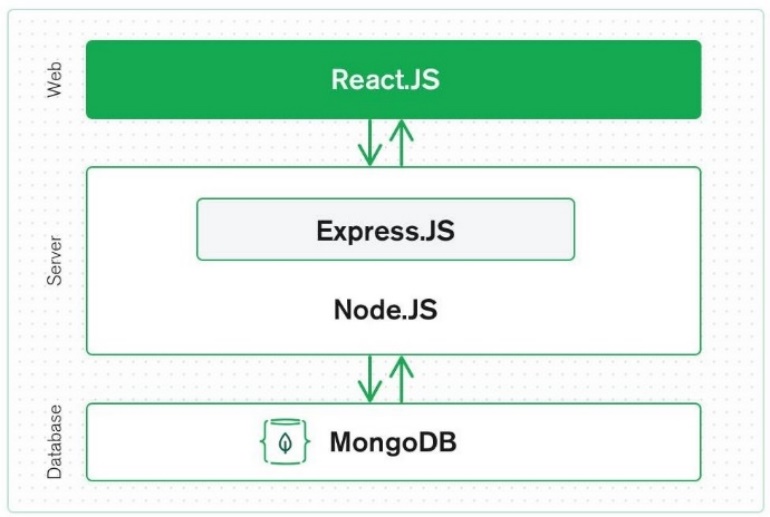
* Không hỗ trợ giao dịch ACID đa nhiệm: MongoDB không hỗ trợ giao dịch ACID (Atomicity, Consistency, Isolation, Durability) đa nhiệm, nghĩa là nó không cung cấp tính chất chuẩn mực về giao dịch như hệ quản trị cơ sở dữ liệu quan hệ.
* Khái quát hóa dữ liệu: Vì MongoDB không yêu cầu mô hình dữ liệu cố định, điều này có thể dẫn đến việc khái quát hóa dữ liệu và khó khăn trong việc đảm bảo tính nhất quán và kiểm soát dữ liệu.
* Tài nguyên hệ thống: MongoDB tiêu tốn nhiều tài nguyên hệ thống hơn so với một số hệ quản trị cơ sở dữ liệu khác, nhất là khi xử lý các hoạt động truy vấn phức tạp hoặc lưu trữ dữ liệu lớn.

1. *Khi nào nên dùng MongoDB*

Khi các ứng dụng có hệ thống thời gian thực đòi hỏi phản hồi nhanh, các dự án big data yêu cầu truy vấn nhanh hay các hệ thống có lượng phản hồi lớn thì nên ưu tiên và các ứng dụng yêu cầu dữ liệu linh hoạt, đòi hỏi khả năng mở rộng ngang. Tuy nhiên, cần xem xét tuỳ theo dự án và trường hợp cụ thể để sử dụng cơ sở dữ liệu quan hệ hay MongoDB nhằm đem lại hiệu quả cao.

## **MERN Stack hoạt động như thế nào?**

MERN cho phép chúng ta dễ dàng xây dựng kiến trúc 3 tầng (frontend, backend, database) hoàn toàn sử dụng JavaScript và JSON.



Hình 3: Kiến trúc MERN Stack

* + 1. React.js Front End

Tầng trên cùng của MERN Stack là React.js, là một thư viện JavaScript phía người dùng (frontend) phổ biến để xây dựng giao diện người dùng tương tác. React sử dụng các thành phần (components) để phân chia giao diện thành các phần nhỏ hơn, dễ quản lý và tái sử dụng. Nó cũng hỗ trợ việc quản lý trạng thái ứng dụng và tương tác với API phía máy chủ thông qua các yêu cầu HTTP.

* + 1. ExpressJS and NodeJS Server

Tầng tiếp theo là ExpressJS, framework web ứng dụng phía máy chủ cho Node.js. Express cung cấp các công cụ và tính năng như định tuyến (routing), middleware và quản lý yêu cầu HTTP. Nó giúp xây dựng các tuyến đường (routes) và xử lý logic của ứng dụng.

Bằng cách thực hiện các HTTP XML Request (XHR) hoặc GETs hoặc POSTs từ React.js front-end, hỗ trợ xây dựng các API, xử lý yêu cầu và gửi dữ liệu đến tầng giao diện người dùng và truy cập và cập nhật dữ liệu trong cơ sở dữ liệu MongoDB.

* + 1. MongoDB Database

Cơ sở dữ liệu đó là MongoDB: là một cơ sở dữ liệu NoSQL, phù hợp cho việc lưu trữ dữ liệu theo cấu trúc tài liệu JSON được gửi đến máy chủ ExpressJS. MongoDB được sử dụng để lưu trữ và truy xuất dữ liệu trong ứng dụng MERN stack và có thể được lưu dưới dạng local trên máy hoặc trên cloud bằng MongoDB Atlas.

* 1. **Cấu trúc của một MERN Stack**
* *Webpack Configs*: MERN sử dụng Webpack cho việc bundle mô-đun. Có bốn loại cấu hình cung cấp: webpack.config.dev.js (cho development), webpack.config.prod.js (cho production), webpack.config.server.js (cho bundling server in production) và webpack.config.babel.js (dành cho babel-plugin-webpack- loaders cho việc render các assets được đưa vào thông qua webpack).
* Server:

+ Package.json: Tệp tin chứa các thông tin về các module và dependencies của server.

+ Node\_modules: Thư mục chứa các module và dependencies của server.

+ Server.js: Tệp tin chứa mã nguồn của server, bao gồm cấu hình và các tuyến đường (routes) của ứng dụng.

+ Routes: Thư mục chứa các tệp tin xử lý các tuyến đường (routes) của ứng dụng.

+ Controllers: Thư mục chứa các tệp tin xử lý logic của các tuyến đường (routes).

+ Models: Thư mục chứa các tệp tin mô hình hóa dữ liệu và tương tác với cơ sở dữ liệu

* Client:

+ Package.json: Tệp tin chứa các thông tin về các module và dependencies của client.

+ Node\_modules: Thư mục chứa các module và dependencies của client.

+ Public: Thư mục chứa các tệp tin tĩnh (ví dụ: HTML, CSS, hình ảnh) của ứng dụng.

+ Components: Thư mục chứa các thành phần (components) của giao diện người dùng.

+ Pages: Thư mục chứa các trang (pages) của ứng dụng.

+ Services: Thư mục chứa các tệp tin xử lý tương tác với API của server.

+ App.js: Tệp tin gốc của ứng dụng React, chứa cấu trúc chính của giao diện.

+ Index.js: Tệp tin khởi động ứng dụng React.

* 1. **Tại sao chọn MERN Stack**

MERN Stack được tạo ra để cung cấp quá trình phát triển mượt mà hơn. Nó là một nhóm các framework, cơ sở dữ liệu và môi trường thời gian chạy dựa trên JavaScript. Công nghệ này cung cấp một khuôn khổ end-to-end cho các nhà phát triển và nó đang trở nên phổ biến nhanh hơn.

Lợi ích của việc chọn MERN Stack:

* Đồng nhất ngôn ngữ: MERN Stack sử dụng JavaScript như ngôn ngữ chính cho cả phần backend và frontend. Điều này giúp giảm sự phân tách giữa các nhóm phát triển và tăng tính nhất quán trong mã nguồn của toàn bộ ứng dụng.
* Mã nguồn mở và phát triển nhanh: Các thành phần của MERN Stack đều là mã nguồn mở (open-source), có một cộng đồng lớn và phát triển nhanh chóng và linh hoạt. Các công cụ như MongoDB, Express, React và Node.js thường có các module và thư viện phong phú, giúp giảm thời gian và công sức trong việc xây dựng ứng dụng web.
* Tích hợp dễ dàng: MERN Stack được thiết kế để tích hợp tốt với nhau. Express và Nodejs cung cấp một môi trường chạy phía máy chủ mạnh mẽ, trong khi React cung cấp một giao diện người dùng tương tác. MongoDB là cơ sở dữ liệu linh hoạt và dễ sử dụng với JavaScript.
* Xây dựng ứng dụng đơn trang (Single-Page Applications - SPA): MERN Stack cho phép xây dựng ứng dụng đơn trang hiệu quả với React. SPA giúp tăng trải nghiệm người dùng bằng cách tải lại nội dung trang một cách mượt mà và giảm đáng kể thời gian phản hồi.
* Phát triển ứng dụng đa nền tảng: Với MERN Stack, bạn có thể phát triển ứng dụng web đa nền tảng, có thể chạy trên các nền tảng khác nhau như máy tính để bàn, di động và web. Điều này giúp tăng khả năng tiếp cận đến người dùng và mở rộng phạm vi sử dụng của ứng dụng.

# **CHƯƠNG 2: MÔ TẢ PHẦN MỀM**

**2.1. Đặc tả đề tài**

* + 1. Mô tả chung về hệ thống website mạng xã hội

Yêu cầu hệ thống: Hệ thống website hỗ trợ thanh tra trong các kỳ thi là nơi thanh tra có thể tìm kiếm, xem thông tin của các sinh viên và phòng thi, điểm danh sinh viên trong phòng thi và gửi báo cáo, biên bản. Hệ thống quản trị quản lý thông tin về phòng thi, sinh viên và thanh tra.

* + 1. Các tính năng có trong website
* Ngôn ngữ lập trình: JavaScript
* Hệ thống cơ sở dữ liệu: MongoDB

Hệ thống website có những chức năng lớn sau:

|  |  |
| --- | --- |
| **Chức năng** | **Mô tả** |
| **Các chức năng dành cho admin** | |
| Đăng nhập, đăng xuất | Chức năng đăng nhập: dùng để đăng nhập vào hệ thống |
| Chức năng đăng xuất: dùng để thoát khỏi hệ thống |
| Quản lý báo cáo | Xem thông tin báo cáo |
| Quản lý phòng thi | Xem thông tin phòng thi, lịch thi |
| Đổ dữ liệu lịch thi |
| Quản lý sinh viên | Xem thông tin sinh viên |
| Thêm và sửa thông tin sinh viên |
| Import danh sách sinh viên |
| Import hình ảnh sinh viên |
| Nhận diện khuôn mặt | Train hình ảnh của sinh viên |
| Quản lý thanh tra | Xem thông tin thanh tra |
| Thêm và sửa thông tin thanh tra |
| Import danh sách thanh tra |
| Quản lý tài khoản | Xem thông tin tài khoản |
| Thêm và sửa thông tin tài khoản |
| Đặt lại mật khẩu |
| Ban và unban tài khoản |
| **Các chức năng dành cho phòng đào tạo** | |
| Đăng nhập, đăng xuất | Chức năng đăng nhập: dùng để đăng nhập vào hệ thống |
| Chức năng đăng xuất: dùng để thoát khỏi hệ thống |
| Quản lý phòng thi | Xem thông tin phòng thi, lịch thi |
| Upload file lịch thĩ |
| **Các chức năng dành cho thanh tra** | |
| Đăng nhập, đăng xuất | Chức năng đăng nhập: dùng để đăng nhập vào hệ thống |
| Chức năng đăng xuất: dùng để thoát khỏi hệ thống |
| Quản lý sinh viên | Xem thông tin sinh viên |
| Tìm kiếm sinh viên |
| Quản lý phòng thi | Tìm kiếm phòng thi |
| Xem thông tin phòng thi, lịch thi |
| Điểm danh sinh viên |
| Xuất thông tin phòng thi ra file excel |
| Nhận diện khuôn mặt | Lấy thông tin khuôn mặt đã được training |
| Quản lý báo cáo | Xem thông tin báo cáo |
| Tạo báo cáo |
| Chỉnh sửa báo cáo |

Bảng 3: Chức năng của Website

* 1. **Mô hình Usecase**
     1. Xác định tác nhân, function và Usecase

Qua khảo sát, kết hợp với việc tham khảo tài liệu và thảo luận nhóm, website hỗ trợ thanh tra trong các kỳ thi sẽ có các tác nhân sau và gom nhóm chức năng như sau:

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tác nhân | Function | Usecase |
| 1 | Admin | Đăng nhập vào hệ thống | Đăng nhập |
| Thanh tra |
| Phòng đào tạo |
| 2 | Admin | * Xem thông tin báo cáo | Quản lý báo cáo |
| Thanh tra | * Xem thông tin báo cáo * Tạo báo cáo * Chỉnh sửa báo cáo |
| 3 | Admin | * Xem thông tin phòng thi * Đổ dữ liệu lịch thi | Quản lý phòng thi |
| Phòng đào tạo | * Xem thông tin phòng thi * Upload file lịch thi |
| Thanh tra | * Tìm kiếm phòng thi * Xem thông tin phòng thi * Điểm danh sinh viên * Xuất thông tin phòng thi ra file excel |
| 4 | Admin | * Xem thông tin sinh viên * Thêm và sửa thông tin sinh viên * Import danh sách sinh viên * Import hình ảnh sinh viên | Quản lý sinh viên |
| Thanh tra | * Tìm kiếm sinh viên * Xem thông tin sinh viên |
| 5 | Admin | Train dữ liệu hình ảnh của sinh viên | Nhận diện khuôn mặt |
| Thanh tra | Lấy thông tin khuôn mặt đã được train |
| 6 | Amin | * Xem thông tin thanh tra * Thêm và sửa thông tin thanh tra * Import danh sách thanh tra | Quản lý thanh tra |
| 7 | Admin | * Xem thông tin tài khoản * Thêm và sửa thông tin tài khoản * Đặt lại mật khẩu * Ban và unban tài khoản | Quản lý tài khoản |

Bảng 4: Tác nhân, Fuction và Usecase

* + 1. Lược đồ Usecase

1. Usecase tổng quan

A diagram of a network

Description automatically generated

Hình 4: Usecase tổng quan

1. Usecase đăng nhập

A diagram of a diagram

Description automatically generated

Hình 6: Usecase đăng nhập

1. Usecase quản lý báo cáo

A diagram of a flowchart

Description automatically generated

Hình 7: Usecase quản lý báo cáo

1. Usecase quản lý phòng thi

A diagram of a person with text

Description automatically generated

Hình 8: Usecase quản lý phòng thi

1. Use case quản lý sinh viên

A diagram of a person with text

Description automatically generated

Hình 9: Use case quản lý sinh viên

1. Usecase nhận diện khuôn mặt

A diagram of a diagram

Description automatically generated

Hình 10: Usecase nhận diện khuôn mặt

1. Usecase Quản lý thanh tra

A diagram of a diagram

Description automatically generated

Hình 11: Usecase quản lý thanh tra

1. Usecase Quản lý tài khoản

A diagram of a diagram

Description automatically generated

Hình 12: Usecase quản lý tài khoản

* + 1. Usecase Diagram

a) Đăng nhập

|  |  |
| --- | --- |
| Usecase Id | UC – 1.1 |
| Usecase Name | Đăng nhập |
| Description | Cho phép người dùng đăng nhập vào tài khoản khi bắt đầu sử dụng |
| Actor | Thanh tra, Admin, Phòng đào tạo |
| Priority | Rất cao |
| Trigger | Người dùng muốn đăng nhập vào website |
| Pre-Condition | Người dùng đã có tài khoản |
| Post-Condition | Chuyển đến trang chủ hoặc thông báo lỗi |
| Basic Flow | 1. Hệ thống hiển thị trang đăng nhập  **{login}**  2. Điền Tài khoản và Mật khẩu  3. Xác nhận đăng nhập bằng cách nhấn nút Login  4. Lưu session và giữ trạng thái đăng nhập |
| Alternative flow | 1. Đăng nhập không thành công:  - Hệ thống kiểm tra thông tin trên TextBox sai  - Hệ thống thông báo tới người dùng  - Chuyển tới **{login}** |

Bảng 6: Usecase Diagram Đăng nhập

b) Quản lý báo cáo Admin

|  |  |
| --- | --- |
| Usecase Id | UC – 1.2.1 |
| Usecase Name | Quản lý báo cáo |
| Description | Hiển thị các báo cáo được thanh tra submit lên hệ thống |
| Actor | Admin |
| Priority | Rất cao |
| Trigger | Người quản trị xem các biên bản hoặc báo cáo sự cố được thanh tra gửi về |
| Pre-Condition | Tài khoản có quyền quản trị |
| Post-Condition | Hiển thị các báo cáo ở dạng bảng và xem chi tiết của từng báo cáo |
| Basic Flow | 1. Chọn menu báo cáo  2. Chọn các điều kiện để lọc báo cáo  3. Thực hiện subflow **Load Reports**  4. Thực hiện subflow **Show Reports** |
| Sub flow | S1: Load Reports  1. Hệ thống kiểm tra các điều kiện lọc đã được chọn.  2. Hệ thống kiểm tra tài khoản có quyền hay không.  3. Hệ thống hiển thị báo cáo của phòng thi theo điều kiện lọc.  S2: Show report  1. Người quản trị nhấn vào báo cáo muốn xem.  2. Hệ thống kiểm tra quyền của người dùng.  3. Hệ thống hiển thị thông tin chi tiết báo cáo gồm loại báo cáo, ghi chú, hình ảnh của báo cáo. |
| Alternative flow | 1. Lấy thông tin báo cáo không thành công:  - Hệ thống thông báo tới người dùng. |

Bảng 8: Usecase Diagram Quản lý người dùng Admin

c) Quản lý báo cáo Thanh tra

|  |  |
| --- | --- |
| Usecase Id | UC – 1.2.2 |
| Usecase Name | Quản lý báo cáo |
| Description | Hiển thị thông tin của báo cáo |
| Actor | Thanh tra |
| Priority | Rất cao |
| Trigger | Người dùng muốn xem, tạo, chỉnh sửa báo cáo |
| Pre-Condition | Người dùng đã có tài khoản |
| Post-Condition | Hiển thị báo cáo, tạo và cập nhật báo cáo |
| Basic Flow | 1. Người truy cập vào phòng thi của ngày thi hôm đó.  2. Thực hiện subflow **Load Reports**  3. Thực hiện subflow **Show Report**  4. Thực hiện subflow **Add Report**  5. Hệ thống lưu lại thông tin |
| Sub flow | S1: Load Reports  1. Hệ thống kiểm tra tài khoản có quyền hay không  2. Hệ thống hiển thị số lượng báo cáo của phòng thi  3. Hệ thống hiện thị một số thông tin như loại báo cáo, note của báo cáo khi người dùng ấn vào nút “Báo cáo”  4. Nhấn icon xóa ở cuối dòng để xóa báo cáo.  S2: Show report  1. Người dùng nhấn vào báo cáo muốn xem  2. Hệ thống kiểm tra quyền của người dùng  3. Hệ thống hiện thị thông tin báo cáo gồm loại báo cáo, ghi chú, hình ảnh của báo cáo  4. Người dùng nhập thông tin muốn thay đổi  5. Xác nhận chỉnh sửa bằng nút “Chỉnh sửa”  S3: Add Report  1. Người dùng chọn icon + để tạo báo cáo  2. Hệ thống hiển thị thông tin của báo cáo trên các textbox  3. Người dùng nhập thông tin muốn thay đổi  4. Người dùng xác nhận thay đổi bằng cách nhấn nút “Tạo báo cáo” |
| Alternative flow | 1. Lấy thông tin báo cáo không thành công:  - Hệ thống thông báo tới người dùng  2. Tạo hoặc chỉnh sửa báo cáo không thành công:  - Hệ thống kiểm tra thông tin trên TextBox không có thông tin  - Hệ thống thông báo tới người dùng |

Bảng 7: Usecase Diagram Quản lý người dùng User

d) Quản lý phòng thi Admin

|  |  |
| --- | --- |
| Usecase Id | UC – 1.3.1 |
| Usecase Name | Quản lý bài viết |
| Description | Hiển thị các bài viết của bạn bè ở trang home (bảng tin) và tương tác, báo cáo hoặc xóa bài viết của bản thân. |
| Actor | User |
| Priority | Rất cao |
| Trigger | Người dùng muốn xem bài viết và thao tác với bài viết |
| Pre-Condition | Người dùng đã có tài khoản |
| Post-Condition | Hiển thị bài viết và các chức năng liên quan |
| Basic Flow | 1. Sau khi đăng nhập hoặc nhấn vào trang home sẽ chuyển hướng đến trang home.  2. Bài viết sẽ được lấy về và hiển thị theo dạng xếp chồng lên nhau.  3. Thực hiện subflow **Like Post**  4. Bấm vào bài viết và hiện lên modal chi tiết, nhập vào ô comment sau đó nhấn gửi hoặc enter sẽ đăng tải comment.  5. Thực hiện subflow **Load Post**  6. Chọn report và chọn lý do để báo cáo bài viết.  7. Chọn Delete và chọn lý do để xóa bài viết  8. Hệ thống lưu lại thông tin |
| Subflow | S1: Like Post  1. Người dùng nhấn nút like bài viết  2. Hệ thống kiểm tra xem người dùng đã like bài viết hay chưa  3. Hệ thống lưu lại thông tin  S2: Load Post  1. Hệ thống kiểm tra bài viết của người dùng hay không  2. Hệ thống hiển thị nút Delete Post nếu là bài viết của mình và Report nếu là bài viết của người dùng khác |
| Alternative flow | 1. Lấy bài viết từ server không thành công:  - Trang web hiển thị không có bài viết.  2. Like bài viết không thành công:  - Trên giao diện vẫn hiển thị đã like tuy nhiên ở server không ghi nhận và lần load sau người dùng sẽ thấy bài viết ở trạng thái chưa like.  3. Comment bài viết không thành công:  - Dòng văn bản sẽ không được hiển thị lên comment section của bài viết.  4. Báo cáo bài viết không thành công:  - Hiển thị thông báo report fail. |

Bảng 9: Usecase Diagram Quản lý bài viết User

e) Quản lý phòng thi Phòng đào tạo

|  |  |
| --- | --- |
| Usecase Id | UC – 1.3.2 |
| Usecase Name | Quản lý bài viết |
| Description | Hiển thị các bài viết của bạn bè ở trang home (bảng tin) và tương tác, báo cáo hoặc xóa bài viết của bản thân. |
| Actor | Phòng đào tạo |
| Priority | Rất cao |
| Trigger | Người dùng muốn xem bài viết và thao tác với bài viết |
| Pre-Condition | Người dùng đã có tài khoản |
| Post-Condition | Hiển thị bài viết và các chức năng liên quan |
| Basic Flow | 1. Sau khi đăng nhập hoặc nhấn vào trang home sẽ chuyển hướng đến trang home.  2. Bài viết sẽ được lấy về và hiển thị theo dạng xếp chồng lên nhau.  3. Thực hiện subflow **Like Post**  4. Bấm vào bài viết và hiện lên modal chi tiết, nhập vào ô comment sau đó nhấn gửi hoặc enter sẽ đăng tải comment.  5. Thực hiện subflow **Load Post**  6. Chọn report và chọn lý do để báo cáo bài viết.  7. Chọn Delete và chọn lý do để xóa bài viết  8. Hệ thống lưu lại thông tin |
| Subflow | S1: Like Post  1. Người dùng nhấn nút like bài viết  2. Hệ thống kiểm tra xem người dùng đã like bài viết hay chưa  3. Hệ thống lưu lại thông tin  S2: Load Post  1. Hệ thống kiểm tra bài viết của người dùng hay không  2. Hệ thống hiển thị nút Delete Post nếu là bài viết của mình và Report nếu là bài viết của người dùng khác |
| Alternative flow | 1. Lấy bài viết từ server không thành công:  - Trang web hiển thị không có bài viết.  2. Like bài viết không thành công:  - Trên giao diện vẫn hiển thị đã like tuy nhiên ở server không ghi nhận và lần load sau người dùng sẽ thấy bài viết ở trạng thái chưa like.  3. Comment bài viết không thành công:  - Dòng văn bản sẽ không được hiển thị lên comment section của bài viết.  4. Báo cáo bài viết không thành công:  - Hiển thị thông báo report fail. |

Bảng 9: Usecase Diagram Quản lý bài viết User

f) Quản lý phòng thi Thanh tra

|  |  |
| --- | --- |
| Usecase Id | UC – 1.3.3 |
| Usecase Name | Quản lý phòng thi |
| Description | Xem thông tin phòng thi và điểm danh sinh viên |
| Actor | Thanh tra |
| Priority | Rất cao |
| Trigger | User muốn xem thông tin phòng thi và điểm danh sinh viên |
| Pre-Condition | Người dùng đã có tài khoản |
| Post-Condition | Hiển thị các bài viết và các chức năng liên quan |
| Basic Flow | 1. Chọn chức năng Tra cứu phòng thi.  2. Thực hiện subflow **Search Exams**  3. Chọn chức năng Lịch thi hôm nay  4. Thực hiện subflow **Attendacne**  5. Thực hiện subflow **Export Excel**  5. Hệ thống lưu lại thông tin |
| Subflow | S1: Search Exams  1. Hệ thống kiểm tra quyền của người dùng  2. Hệ thống lấy thông tin của các năm  3. Người dùng chọn năm, kỳ, ngày, khu vực, thời gian và phòng thi từ các ô select để lấy đúng phòng thi mong đợi  4. Hệ thống hiển thị thông tin phòng thi  S2: Attendacne  1. Người dùng chọn phòng thi  2. Hệ thống kiểm tra quyền của người dùng  3. Hệ thống hiển thị thông tin của phòng thi  4. Người dùng chọn điểm danh để điểm danh sinh viên trong phòng thi  5. Hệ thống lấy thông tin camera và hiển thị khung điểm danh  6. Sinh viên đưa khuôn mặt vào camera để hệ thống lấy thông tin và điểm danh  7. Người dùng chọn mục hình ảnh và chọn hình ảnh cần điểm danh  8. Hệ thống lấy thông tin từ hình ảnh và điểm danh cho sinh viên trong ảnh  9. Hệ thống lưu lại thông tin  S2: Export Excel  1. Người dùng chọn ngày thi  2. Hệ thống lấy thông tin của các phòng thi  3. Người dùng nhấn nút “Xuất file excel”  4. Hệ thống download file excel chứa thông tin của các phòng thi về máy của người dùng  5. Chọn một phòng thi và nhấn nút “Xuất file excel” để download file excel chứa thông tin của phòng thi đó |
| Alternative flow | 1. Lấy thông tin camera không thành công:  - Trang web hiển thị thông báo lỗi khi lấy thông tin camera. |

Bảng 10: Usecase Diagram Quản lý bài viết Admin

g) Quản lý sinh viên Admin

|  |  |
| --- | --- |
| Usecase Id | UC – 1.4.1 |
| Usecase Name | Thống kê |
| Description | Admin thống kê người dùng và bài viết |
| Actor | Admin |
| Priority | Cao |
| Trigger | Admin thống kê người dùng và bài viết |
| Pre-Condition | Người dùng đã đăng nhập vào hệ thống |
| Post-Condition | Hiển thị nội dung thống kê hoặc thông báo lỗi |
| Basic Flow | 1. Người dùng chọn nút “Dashboard”  2. Thực hiện subflow **Load Statistic** |
| Sub flow | S1: Load Statistic  1. Hệ thống kiểm tra số lượng người dùng mới và số post mới  2. Hệ thống so sánh số lượng với ngày hôm qua và hiển thị dưới dạng biểu đồ  3. Hệ thống hiển thị người mới đăng nhập và người dùng nhiều bài viết nhất  4. Hiển thị thông tin của đoạn hội thoại (tên, ảnh, tin nhắn cuối) |
| Alternative flow | 1. Lấy thống từ server không thành công:  - Trang web hiển thị không có thống kê. |

Bảng 11: Usecase Diagram Thống kê

h) Quản lý sinh viên Thanh tra

|  |  |
| --- | --- |
| Usecase Id | UC – 1.4.2 |
| Usecase Name | Quản lý sinh viên |
| Description | Tìm kiếm và hiển thị thông tin sinh viên |
| Actor | Thanh tra |
| Priority | Rất cao |
| Trigger | Người dùng muốn xem thông tin sinh viên |
| Pre-Condition | Người dùng đã đăng nhập vào hệ thống |
| Post-Condition | Hiển thị thông tin sinh viên hoặc thông báo lỗi |
| Basic Flow | 1. Người dùng chọn Tra cứu sinh viên  2. Thực hiện subflow **Load Students**  3. Thực hiện subflow **Search Students**  4. Nhấn vào sinh viên muốn xem  5. Hệ thống hiển thị thông tin của sinh viên được chọn  6. Hệ thống lưu lại thông tin |
| Sub flow | S1: Load Students  1. Hệ thống kiểm tra quyền của người dùng  2. Hiển thị thông tin của 20 sinh viên  S2: Search Students  **{search}**  1. Người dùng nhập mã số sinh viên hoặc tên của sinh viên muốn tìm kiếm  2. Nhấn nút search hoặc enter để tìm kiếm sinh viên  3. Hệ thống kiểm tra quyền của người dùng  4. Hệ thống hiển thị 20 sinh viên có thông tin trùng với sinh viên được tìm kiếm |
| Alternative flow | 1. Tìm kiếm không thành công  - Hệ thống kiểm tra thông tin trên TextBox có nội dung nào hay không  - Chuyển đến **{search}** |

Bảng 12: Usecase Diagram Nhắn tin

i) Nhận diện khuôn mặt Admin

|  |  |
| --- | --- |
| Usecase Id | UC – 1.5.1 |
| Usecase Name | Nhận diện khuôn mặt |
| Description | Train hình ảnh được thêm vào |
| Actor | Admin |
| Priority | Rất cao |
| Trigger | Người dùng training dữ liệu hình ảnh để điểm danh |
| Pre-Condition | Người dùng đã đăng nhập vào hệ thống |
| Post-Condition | Hiển thị đã gửi train thành công hoặc thất bại |
| Basic Flow | 1. Vào trang Sinh viên  2. Thực hiện subflow **Train Data**  3. Hệ thống lưu lại thông tin |
| Subflow | S1: Train Data  1. Người dùng chọn nút “Train hình ảnh”  2. Hệ thống kiểm tra quyền của người dùng  3. Hệ thống kiểm tra những hình ảnh được thêm vào gần đây  4. Hệ thống dùng các hình ảnh này để train và đánh label, lưu thành file json để sử dụng |
| Alternative flow | 1. Train dữ liệu không thành công  - Hệ thống hiển thị thông báo không thành công |

Bảng 13: Usecase Diagram Bạn bè

k) Nhận diện khuôn mặt Thanh tra

|  |  |
| --- | --- |
| Usecase Id | UC – 1.5.2 |
| Usecase Name | Nhận diện khuôn mặt |
| Description | Lấy dữ liệu khuôn mặt được đã được train |
| Actor | Thanh tra |
| Priority | Rất cao |
| Trigger | Người dùng lấy thông tin khuôn mặt để điểm danh |
| Pre-Condition | Người dùng đã đăng nhập vào hệ thống |
| Post-Condition | Hiển thị thông báo nếu thất bại |
| Basic Flow | 1. Mở website  2. Thực hiện subflow **Load Data**  3. Hệ thống lưu lại thông tin |
| Subflow | S1: Load Data  1. Hệ thống kiểm tra quyền của người dùng  2. Hệ thống lấy thông tin khuôn mặt để diểm danh |
| Alternative flow | 1. Tìm kiếm người dùng không thành công  - Hệ thống hiển thị thông báo không thành công |

Bảng 14: Usecase Diagram Tìm kiếm người dùng

l) Thông báo

|  |  |
| --- | --- |
| Usecase Id | UC – 1.7 |
| Usecase Name | Thông báo |
| Description | Xem thông báo và gửi thông báo |
| Actor | User |
| Priority | Cao |
| Trigger | Người dùng muốn xem thông báo và gửi thông báo |
| Pre-Condition | Người dùng đã đăng nhập vào hệ thống |
| Post-Condition | Hiển thị thông báo hoặc hiển thị không có thông báo |
| Basic Flow | 1. Mở thanh Sidebar  2. Thực thiện subflow **Load Notifies**  3. Thực hiện subflow **Send Notify** |
| Subflow | S1: Load Notifies  1. Người dùng chọn chức năng Thông báo  2. Hệ thống kiểm tra loại thông báo  3. Hệ thống hiện thị Accept, Reject nếu là lời mời kết bạn  S2: Send Notify  1. Người dùng tương tác với người dùng khác  2. Hệ thống lưu lại  3. Hệ thống gửi thông báo tới người được tương tác |
| Alternative flow | 1. Gửi thông báo không thành công  - Hệ thống không gửi thông báo tới người được tương tác |

Bảng 15: Usecase Diagram Thông báo

m) Quản lý nhóm

|  |  |
| --- | --- |
| Usecase Id | UC – 1.8 |
| Usecase Name | Quản lý nhóm |
| Description | Thêm thành viên và phê duyệt bài viết |
| Actor | Leader |
| Priority | Cao |
| Trigger | Trưởng nhóm thêm thành viên và phê duyệt bài viết |
| Pre-Condition | Người dùng đã đăng nhập vào hệ thống |
| Post-Condition | Hiển thị thông báo thành công hoặc thất bại |
| Basic Flow | 1. Trưởng nhóm chọn group của mình  2. Trưởng nhóm thêm thành viên bằng cách thực hiện subflow **Add member**  3. Trưởng nhóm duyệt bài viết bằng cách thực hiện subflow **Approve post**  4. Hệ thống lưu lại thông tin |
| Subflow | S1: Add member  1. Trưởng nhóm chọn phần request  2. Hệ thống hiển thị những user muốn vào nhóm  3. Trưởng nhóm chọn Accept, Reject để thêm hoặc từ chối người dùng vào nhóm  4. Hệ thống lưu lại  S2: Approve post  1. Trưởng nhóm chọn phần Approve  2. Hệ thống hiển thị những post muốn đăng  3. Trưởng nhóm chọn Accept, Reject để thêm hoặc từ chối bài viết đăng trong nhóm  4. Hệ thống lưu lại |
| Alternative flow | 1. Load người dùng hoặc bài viết không thành công  - Không hiển thị người dùng hoặc bài viết |

Bảng 16: Usecase Diagram Quản lý nhóm

* 1. **Thiết kế cơ sở dữ liệu**
     1. Danh sách collections NOSQL

A screenshot of a computer

Description automatically generated

Hình 13: Danh sách Collections

* + 1. Từ điển dữ liệu

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Bảng Accounts** | | | | | |
| **TT** | **Thuộc tính** | **Diễn giải** | **Kiểu dữ liệu** | **Ràng buộc** | **Ghi chú** |
| 1 | \_id | Id của tài khoản | ObjectId | PRIMARY KEY |  |
| 2 | username | Tên đăng nhập của tài khoản | String | REQUIRED |  |
| 3 | password | Mật khẩu của tài khoản | String | REQUIRED |  |
| 4 | full\_name | Tên người dùng | String | REQUIRED |  |
| 5 | email | Email của người dùng | Object | REQUIRED |  |
| 6 | avatar | Avatar của người dùng | String | DEFAULT  NULL |  |
| 7 | inpector | Mã id của thanh tra | Array | DEFAULT  NULL |  |
| 8 | friend\_request | Lời mời kết bạn | Array | DEFAULT  NULL |  |
| 9 | friend\_request\_sent | Người mà người dùng muốn kết bạn | Array | DEFAULT  NULL |  |
| 10 | online | Người dùng hoạt động hay không | Boolean | DEFAULT  NULL |  |
| 11 | last\_online | Thời gian gần nhất mà người dùng truy cập | Date | DEFAULT  NULL |  |
| 12 | banned | Bị ban hay chưa | Boolean | REQUIRED |  |
| 13 | Role | Có phải admin hay không | Boolean | REQUIRED |  |
| 14 | reset\_token | Token được rest khi hết hạn | String | DEFAULT  NULL |  |
| 15 | created\_at | Thời gian tạo tài khoản | Timestamp | DEFAULT |  |
| 16 | updated\_at | Thời gian cập nhật | Timestamp | DEFAULT |  |

Bảng 17: Bảng Users

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Bảng Conversations** | | | | | |
| **TT** | **Thuộc tính** | **Diễn giải** | **Kiểu dữ liệu** | **Ràng buộc** | **Ghi chú** |
| 1 | \_id | Id của hội thoại | ObjectId | PRIMARY KEY |  |
| 2 | users | Người dùng trong hội thoại | Array | REQUIRED |  |
| 3 | last\_message | Message gần nhất trong hội thoại | ObjectId | DEFAULT  NULL |  |
| 4 | is\_deleted | Cuộc hội thoại đã được xóa hay chưa | Object | DEFAULT  NULL |  |
| 5 | created\_at | Thời gian tạo hội thoại | Timestamp | DEFAULT |  |
| 6 | updated\_at | Thời gian cập nhật | Timestamp | DEFAULT |  |

Bảng 18: Bảng Conversations

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Bảng Messages** | | | | | |
| **TT** | **Thuộc tính** | **Diễn giải** | **Kiểu dữ liệu** | **Ràng buộc** | **Ghi chú** |
| 1 | \_id | Id của tin nhắn | ObjectId | PRIMARY KEY |  |
| 2 | conversation\_id | Id của hội thoại | ObjectId | REQUIRED |  |
| 3 | sender | Id người gửi | ObjectId | REQUIRED |  |
| 4 | content | Nội dung tin nhắn | String | DEFAULT  NULL |  |
| 5 | media | Link hình ảnh | Array | DEFAULT  NULL |  |
| 6 | removed | Tin nhắn đã bị xóa hay chưa | Boolean | DEFAULT  NULL |  |
| 7 | reader | Người đọc tin nhắn | Array | DEFAULT  NULL |  |
| 8 | created\_at | Thời gian tạo tin nhắn | Timestamp | DEFAULT |  |
| 9 | updated\_at | Thời gian cập nhật | Timestamp | DEFAULT |  |

Bảng 19: Bảng Messages

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Bảng Notifications** | | | | | |
| **TT** | **Thuộc tính** | **Diễn giải** | **Kiểu dữ liệu** | **Ràng buộc** | **Ghi chú** |
| 1 | \_id | Id của thông báo | ObjectId | PRIMARY KEY |  |
| 2 | user\_id | Id của người dùng nhận thông báo | ObjectId | REQUIRED |  |
| 3 | sender\_id | Id người gửi thông báo | ObjectId | REQUIRED |  |
| 4 | content | Nội dung thông báo | String | REQUIRED |  |
| 5 | content\_id | Id bài post | ObjectId | DEFAULT NULL |  |
| 6 | read | Đã đọc hay chưa | Boolean | REQUIRED |  |
| 7 | reponse | Trả lời hay chưa với thông báo lời mời kết bạn | Boolean | DEFAULT NULL |  |
| 8 | created\_at | Thời gian tạo thông báo | Timestamp | DEFAULT |  |
| 9 | updated\_at | Thời gian cập nhật | Timestamp | DEFAULT |  |

Bảng 20: Bảng Notifications

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Bảng Posts** | | | | | |
| **TT** | **Thuộc tính** | **Diễn giải** | **Kiểu dữ liệu** | **Ràng buộc** | **Ghi chú** |
| 1 | \_id | Id của bài viết | ObjectId | PRIMARY KEY |  |
| 2 | creator | Id của người tạo | ObjectId | REQUIRED |  |
| 3 | group | Id của group | ObjectId | DEFAULT  NULL |  |
| 4 | content | Nội dung bài viết | String | DEFAULT  NULL |  |
| 5 | media | Hình ảnh của bài viết | Array | DEFAULT  NULL |  |
| 6 | reacts | Id của react | Array | DEFAULT  NULL |  |
| 7 | deleted\_by | Được xóa bởi ai (user hay admin) | String | DEFAULT  NULL |  |
| 8 | created\_at | Thời gian tạo bài viết | Timestamp | DEFAULT |  |
| 9 | updated\_at | Thời gian cập nhật | Timestamp | DEFAULT |  |

Bảng 21: Bảng Posts

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Bảng Reacts** | | | | | | |
| **TT** | **Thuộc tính** | **Diễn giải** | **Kiểu dữ liệu** | **Ràng buộc** | **Ghi chú** |
| 1 | \_id | Id của react | ObjectId | PRIMARY KEY |  |
| 2 | post | Id của bài viết | ObjectId | REQUIRED |  |
| 3 | reacted\_by | Id của người react | ObjectId | REQUIRED |  |
| 4 | emoji | Loại emoji | String | REQUIRED |  |
| 5 | created\_at | Thời gian tạo bài viết | Timestamp | DEFAULT |  |
| 6 | updated\_at | Thời gian cập nhật | Timestamp | DEFAULT |  |

Bảng 22: Bảng Reacts

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Bảng Comments** | | | | | |
| **TT** | **Thuộc tính** | **Diễn giải** | **Kiểu dữ liệu** | **Ràng buộc** | **Ghi chú** |
| 1 | \_id | Id của comment | ObjectId | PRIMARY KEY |  |
| 2 | user | Id của người comment | ObjectId | REQUIRED |  |
| 3 | post | Id của bài viết đã comment | ObjectId | REQUIRED |  |
| 4 | cmt\_level | Level của comment | Number | REQUIRED |  |
| 5 | reply\_to | Reply comment nào | ObjectId | DEFAUL NULL |  |
| 6 | content | Nội dung đã comment | String | DEFAUL NULL |  |
| 7 | deleted\_by | Được xóa bởi ai (user hay admin) | String | DEFAUL NULL |  |
| 8 | created\_at | Thời gian tạo bài viết | Timestamp | DEFAULT |  |
| 9 | updated\_at | Thời gian cập nhật | Timestamp | DEFAULT |  |

Bảng 23: Bảng Comments

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Bảng Report users** | | | | | |
| **TT** | **Thuộc tính** | **Diễn giải** | **Kiểu dữ liệu** | **Ràng buộc** | **Ghi chú** |
| 1 | \_id | Id của report | ObjectId | PRIMARY KEY |  |
| 2 | user | Id của người bị report | ObjectId | REQUIRED |  |
| 3 | reported\_by | Id của người report | ObjectId | REQUIRED |  |
| 4 | reason | Lý do report | String | REQUIRED |  |
| 5 | created\_at | Thời gian tạo bài viết | Timestamp | DEFAULT |  |
| 6 | updated\_at | Thời gian cập nhật | Timestamp | DEFAULT |  |

Bảng 24: Bảng Report users

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Bảng Report posts** | | | | | |
| **TT** | **Thuộc tính** | **Diễn giải** | **Kiểu dữ liệu** | **Ràng buộc** | **Ghi chú** |
| 1 | \_id | Id của report | ObjectId | PRIMARY KEY |  |
| 2 | post | Id của bài viết bị report | ObjectId | REQUIRED |  |
| 3 | reported\_by | Id của người report | ObjectId | REQUIRED |  |
| 4 | reason | Lý do report | String | REQUIRED |  |
| 5 | created\_at | Thời gian tạo bài viết | Timestamp | DEFAULT |  |
| 6 | updated\_at | Thời gian cập nhật | Timestamp | DEFAULT |  |

Bảng 25: Bảng Report posts

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Bảng Groups** | | | | | |
| **TT** | **Thuộc tính** | **Diễn giải** | **Kiểu dữ liệu** | **Ràng buộc** | **Ghi chú** |
| 1 | \_id | Id của group | ObjectId | PRIMARY KEY |  |
| 2 | name | Tên của group | String | REQUIRED |  |
| 3 | description | Giới thiệu về group | String | DEFAUL NULL |  |
| 4 | cover | Ảnh của group | String | DEFAUL NULL |  |
| 5 | deleted | Được xóa hay chưa | String | DEFAUL NULL |  |
| 6 | created\_at | Thời gian tạo group | Timestamp | DEFAULT |  |
| 7 | updated\_at | Thời gian cập nhật | Timestamp | DEFAULT |  |

Bảng 26: Bảng Groups

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Bảng UserToGroup** | | | | | |
| **TT** | **Thuộc tính** | **Diễn giải** | **Kiểu dữ liệu** | **Ràng buộc** | **Ghi chú** |
| 1 | \_id | Id của user trong group | ObjectId | PRIMARY KEY |  |
| 2 | group | Id của group | ObjectId | REQUIRED |  |
| 3 | user | Id của user | ObjectId | REQUIRED |  |
| 4 | status | Trạng thái của user trong group | String | REQUIRED |  |
| 5 | created\_at | Thời gian tạo document | Timestamp | DEFAULT |  |
| 6 | updated\_at | Thời gian cập nhật | Timestamp | DEFAULT |  |

Bảng 27: Bảng UserToGroup

* + 1. Class Diagram

A screenshot of a computer

Description automatically generated

Hình 14: Class Diagram

* 1. **Phân tích thiết kế các thành phần chức năng**
     1. Biểu đồ sequence diagram đăng nhập

A diagram of a diagram

Description automatically generated

Hình 15: Sequence Diagram Đăng nhập

* + 1. Biểu đồ sequence diagram quản lý báo cáo

A diagram of a diagram

Description automatically generated with medium confidence

Hình 16: Sequence Diagram Quản lý báo cáo của thanh tra

A diagram of a diagram

Description automatically generated

* + 1. Biểu đồ sequence diagram quản lý phòng thi

A diagram of a diagram

Description automatically generated

Hình 17: Sequence Diagram Quản lý phòng thi của thanh tra

A diagram of a diagram

Description automatically generated

* + 1. Biểu đồ sequence diagram quản lý sinh viên

A diagram of a diagram

Description automatically generated

Hình 18: Sequence Diagram Quản lý sinh viên của thanh tra

A diagram of a computer

Description automatically generated with medium confidence

* + 1. Biểu đồ sequence diagram nhận diện khuôn mặt

A diagram of a diagram

Description automatically generated

Hình 19: Sequence Diagram Nhận diện khuôn mặt của admin

A diagram of a diagram of a line

Description automatically generated with medium confidence

Hình 19: Sequence Diagram Nhận diện khuôn mặt của thanh tra

* + 1. Biểu đồ sequence diagram Quản lý thanh tra

A diagram of a structure

Description automatically generated

Hình 20: Sequence Diagram Kết bạn

* + 1. Biểu đồ sequence diagram quản lý tài khoản

A diagram of a diagram

Description automatically generated with medium confidence

Hình 21: Sequence Diagram Nhắn tin

* 1. **Activity Diagram**
     1. Activity Diagram Đăng nhập

**A diagram of a diagram

Description automatically generated with medium confidence**

Hình 29: Activity Đăng nhập

* + 1. Activity Diagram Quản lý báo cáo
       1. Admin

A screenshot of a diagram

Description automatically generated

Hình 30: Activity Diagram Bài viết

* + - 1. Thanh tra

A diagram of a diagram

Description automatically generated with medium confidence

Hình 30: Activity Diagram Xem báo cáo

A diagram of a diagram

Description automatically generated with medium confidence

Hình 30: Activity Diagram Tạo báo cáo

A diagram of a diagram

Description automatically generated with medium confidence

Hình 30: Activity Diagram Chỉnh sửa báo cáo

* + 1. Activity Diagram Quản lý phòng thi
       1. Admin

A screenshot of a diagram

Description automatically generated

Hình 31: Activity Diagram Đăng bài viết

* + - 1. Phòng đào tạo
      2. Thanh tra

A diagram of a diagram

Description automatically generated with medium confidence

A diagram of a diagram

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated

* + 1. Activity Diagram Quản lý sinh viên
       1. Admin

A screenshot of a diagram

Description automatically generated

Hình 32: Activity Diagram Trang cá nhân

* + - 1. Thanh tra

A diagram of a diagram

Description automatically generated with medium confidence

* + 1. Activity Diagram Nhận diện khuôn mặt
       1. Admin

A diagram of a diagram

Description automatically generated with medium confidence

Hình 33: Activity Diagram Nhắn tin

* + - 1. Thanh tra

A screenshot of a diagram

Description automatically generated

Hình 33: Activity Diagram Nhắn tin

* + 1. Activity Diagram Quản lý thanh tra

A diagram of a group of text

Description automatically generated with medium confidence

Hình 34: Activity Diagram Nhóm cộng đồng

* + 1. Activity Diagram Quản lý tài khoản

A diagram of a diagram

Description automatically generated with medium confidence

Hình 35: Activity Diagram Thông báo

* 1. **Phân tích, thiết kế giao diện và chức năng của giao diện**
     1. Giao diện admin

Truy cập đường dẫn <http://localhost:3000/login> để vào trang đăng nhập dành cho admin với username và password là (admin001/admin001) để truy cập vào hệ thống:

A screenshot of a login box

Description automatically generated

Hình 40: Giao diện Đăng nhập Admin

Trang báo cáo: Hiển thị các báo cáo được filter theo năm và kỳ và chọn báo cáo để có thể xem thông tin của báo cáo được chọn

A screenshot of a computer

Description automatically generated

Hình 41: Giao diện Dashboard

Trang lịch thi: Admin có thể chọn tòa nhà được sử dụng để thi được filer theo năm, kỳ và ngày. Ngoài ra có thể thực hiện đổ dữ liệu từ file lịch thi được phòng đào tạo upload lên database để cập nhật cho website

A screenshot of a computer

Description automatically generated

Hình 42: Giao diện Quản lý người dùng

Trang chi tiết phòng thi: Admin chọn phòng thi từ ca thi và tầng để xem thông tin sinh viên và điểm danh của phòng

A screenshot of a computer

Description automatically generated

Hình 43: Giao diện quản lý bài viết

Trang tài khoản: Admin có thể tìm kiếm bằng tên đăng nhập hoặc tên, chọn tài khoản để ban hoặc unban cho tài khoản, ấn nút “Tạo” để tạo tài khoản

A screenshot of a computer

Description automatically generated

Hình 43: Giao diện quản lý bài viết

Trang sinh viên: Admin có thể tìm kiếm bằng tên hoặc mã số sinh viên, chọn sinh viên để xem hoặc chỉnh sửa thông tin, ấn nút “Thêm” để tạo sinh viên. Ngoài ra còn các chắc năng như import danh sách sinh viên, hình ảnh sinh viên và ấn “Train ảnh” để tạo dữ liệu khuôn mặt cho thanh tra điểm danh

A screenshot of a computer

Description automatically generated

Hình 43: Giao diện quản lý bài viết

Trang thanh tra: Admin có thể tìm kiếm bằng tên hoặc mã thanh tra, chọn thanh tra để xem hoặc chỉnh sửa thông tin, ấn nút “Thêm” để thêm thanh tra. Ngoài ra còn các chắc năng như import danh sách thanh tra

A screenshot of a computer

Description automatically generated

Hình 43: Giao diện quản lý bài viết

* + 1. Giao diện client

Giao diện đăng nhập: Thanh tra nhập tên đăng nhập và mật khẩu được cung cấp để đăng nhập vào website.

A screenshot of a login box

Description automatically generated

Hình 44: Giao diện Đăng nhập người dùng

Giao diện trang cứu sinh viên: bao gồm sidebar, nhập thông tin sinh viên muốn tìm kiếm ấn Enter hoặc nút search để tìm kiếm

A screenshot of a computer

Description automatically generated

Hình 46: Giao diện Trang chủ

Giao diện thông tin sinh viên: Khi nhấn vào sinh viên sẽ hiển thị thông tin sinh viên

A screenshot of a computer

Description automatically generated

Hình 47: Giao diện Bài viết

Trang lịch thi: Thanh tra có thể tra cứu tòa nhà được sử dụng để thi filer theo năm, kỳ và ngày.

A screenshot of a computer

Description automatically generated

Hình 42: Giao diện Quản lý người dùng

Trang chi tiết phòng thi: Admin chọn phòng thi từ ca thi và tầng để xem thông tin sinh viên và điểm danh của phòng

A screenshot of a computer

Description automatically generated

Hình 43: Giao diện quản lý bài viết

Giao diện trang chủ: bao gồm danh sách tòa nhà được thi hôm nay, danh sách các sinh viên nghi vấn và có nút xuất danh sách để xuất danh sách tất cả lịch thi trong ngày

A screenshot of a computer

Description automatically generated

Hình 48: Giao diện Trang chủ

Tại đây khi click chọn phòng sẽ ra thông tin phòng thi có các chức năng như báo cáo, điểm danh sinh viên và xuất file điểm danh sinh viên

A screenshot of a computer

Description automatically generated

Hình 49: Giao diện Friends, Requests

Còn đây là chức năng điểm danh sinh viên bằng camera bên cạnh đó là chức năng điểm danh bằng hình ảnh

A screenshot of a computer

Description automatically generated

Hình 50: Giao diện Trang cá nhân người dùng khác

Giao diện báo cáo: Người dùng chọn báo cáo để các báo cáo trong phòng, tạo hoặc xóa báo cáo.

A screenshot of a phone

Description automatically generated

Hình 51: Giao diện Tìm kiếm

A screenshot of a computer

Description automatically generated

Hình 51: Giao diện Tìm kiếm

# **CHƯƠNG 3: KIỂM THỬ PHẦN MỀM**

## **3.1. Một số test case**

Danh sách các 60 testcase được thực hiện tại đường dẫn sau: <https://docs.google.com/spreadsheets/d/1fOsGUICFbjuhU-ICBQNU66psS_DzXO5A/edit?usp=sharing&ouid=102622952634724094154&rtpof=true&sd=true>

4.1.1. Testcase Đăng nhập

A screenshot of a computer

Description automatically generated

Hình 63: Testcase Đăng nhập

4.1.2. Testcase Xem báo cáo của admin

A screenshot of a computer

Description automatically generated

Hình 64: Testcase Đăng bài viết

4.1.4. Testcase Giao diện lịch thi của admin

A screenshot of a computer

Description automatically generated

Hình 65: Testcase Chỉnh sửa thông tin cá nhân

4.1.5. Testcase Chỉnh sửa tài khoản

A screenshot of a computer

Description automatically generated

Hình 66: Testcase Nhắn tin

4.1.6. Testcase Import hình ảnh sinh viên

A screenshot of a computer

Description automatically generated

Hình 67: Testcase Kết bạn

4.1.6. Testcase Xem thanh tra

A screenshot of a document

Description automatically generated

Hình 68: Testcase Bình luận

4.1.6. Testcase Đổi mật khẩu

A screenshot of a computer

Description automatically generated

Hình 69: Testcase Đổi mật khẩu

## **4.2. Test report**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tính năng** | **Số testcase** | **Số testcase pass** | **Số testcase không pass** |
| Bình luận bài viết | 4 | 3 | 1 |
| Đăng bài viết | 6 | 5 | 1 |
| Thích bài viết | 2 | 2 | 0 |
| Hủy thích bài viết | 1 | 1 | 0 |
| Chỉnh sửa thông tin Bio | 2 | 1 | 1 |
| Thông báo | 1 | 0 | 1 |
| Chỉnh sửa ảnh đại diện | 3 | 2 | 1 |
| Thay đổi mật khẩu | 5 | 3 | 2 |
| Đăng nhập | 7 | 7 | 0 |
| Đăng ký | 8 | 6 | 2 |
| Nhắn tin | 8 | 6 | 2 |
| Profile | 7 | 6 | 1 |
| Tìm kiếm người dùng | 6 | 6 | 0 |
| Tổng | 60 | 48 | 12 |
| Tỷ lệ đã testcase đã test | 100% | | |
| Tỷ lệ testcase pass | 80% | | |

Bảng 28: Tổng hợp các testcase

## **4.3. Bug report**

4.3.1. Hệ thống quản lý bug

Jira là một ứng dụng có khả năng quản lý dự án, kiểm soát và theo dõi lỗi hoặc vấn đề phát sinh trong doanh nghiệp, tổ chức. Jira được ví như nhân viên kiểm thử mà bất kỳ tổ chức, doanh nghiệp nào cũng cần có và biết cách sử dụng nó là kỹ năng cần thiết trong thời điểm này.

Phát triển phần mềm Jira được thực hiện bởi các kỹ sư của công ty phần mềm Atlassian đến từ Australia. Jira đưa ra các nhận định, cảnh báo dựa trên trọng tâm và hiệu quả công việc mà doanh nghiệp đạt được. Người dùng có thể sử dụng thường xuyên, xuyên suốt quá trình làm việc để kịp thời phát hiện vấn đề phát sinh.



Hình 70: Hệ thống quản lý bug Jira

Ứng dụng của Jira:  
*Quản lý công việc*

* Theo Dõi Tiến Độ: Jira cung cấp các công cụ mạnh mẽ để theo dõi tiến độ của các công việc và nhiệm vụ trong dự án. Người dùng có thể xem biểu đồ Gantt, bảng Kanban, hoặc theo dõi thời gian thực.
* Quản Lý Nhiệm Vụ: Jira cho phép tạo và quản lý các nhiệm vụ, mô tả chi tiết công việc cần thực hiện, gán người thực hiện, và xác định thời hạn.

*Theo dõi lỗi và cải thiện chất lượng*

* Quản Lý Lỗi (Bug): Jira là một công cụ mạnh mẽ để theo dõi và quản lý lỗi trong quá trình kiểm thử. Người dùng có thể tạo lỗi, gán người xử lý, theo dõi trạng thái, và gắn liên kết với các nhiệm vụ khác.
* Quản Lý Cải Thiện (Improvement): Bên cạnh việc quản lý lỗi, Jira còn cho phép theo dõi các yêu cầu cải thiện, phản hồi, và các công việc nhằm nâng cao chất lượng sản phẩm.

*Tích hợp với công cụ kiểm thử*

* Xử Lý Tác Vụ Kiểm Thử: Jira có khả năng tích hợp với các công cụ kiểm thử phổ biến như Selenium, TestRail, và các công cụ khác, giúp tự động hóa và theo dõi quá trình kiểm thử.
* Báo Cáo và Phân Tích: Jira cung cấp khả năng tạo và tùy chỉnh báo cáo về tiến độ kiểm thử, lượng lỗi, và các thống kê khác để hỗ trợ quyết định quản lý.

*Quản lý người dùng và phân quyền*

* Quản Lý Người Dùng: Jira cho phép quản lý người dùng và gán các vai trò khác nhau, giúp phân chia trách nhiệm và quyền lợi.
* Phân Quyền: Xác định quyền lợi của từng thành viên trong dự án, đảm bảo rằng thông tin quan trọng chỉ được tiếp cận bởi những người cần biết.

*Theo dõi thời gian và báo cáo chi tiết*

* Theo Dõi Thời Gian: Jira có khả năng theo dõi thời gian làm việc của từng công việc, giúp đánh giá và quản lý tài nguyên hiệu quả.
* Báo Cáo Chi Tiết: Tạo và tùy chỉnh báo cáo để theo dõi và đánh giá hiệu suất của dự án, từ đó giúp quản lý đưa ra quyết định đúng đắn.

Jira không chỉ là một công cụ quản lý dự án mà còn là một nền tảng linh hoạt và mở rộng được ứng dụng rộng rãi trong nhiều ngành và quy trình làm việc khác nhau.

4.3.2. Chi tiết các bug được phát hiện

a) Đăng ký bằng các ký tự khoảng trắng

Khi đăng ký với họ và tên để trống vẫn chuyển hướng đến trang OTP để xác thực mà không kiểm tra nội dung trống hay không: <https://kduong.atlassian.net/browse/SW-1>

A screenshot of a computer

Description automatically generated

Hình 71: Bug Đăng kí bằng các kí tự khoảng trắng

b) Nhấn nhiều lần nút đăng kí gửi nhiều mã OTP

Khi nhấn nhiều lần nút đăng kí trong khi chuyển hướng đến trang OTP để xác thực thì sẽ gửi nhiều mã OTP: <https://kduong.atlassian.net/browse/SW-2>

A screenshot of a computer

Description automatically generated

Hình 72: Bug Nhấn nhiều lần nút đăng kí gửi nhiều mã OTP

c) Sai định dạng file đăng bài viết

Trên trang chủ sau khi đăng nhập, ta chọn nút đăng bài viết với file định dạng không cho phép: <https://kduong.atlassian.net/browse/SW-3>

A screenshot of a computer

Description automatically generated

Hình 73: Bug Sai định dạng file đăng bài viết

d) Sai định dạng file hình đại diện

Trên trang chỉnh sửa hồ sơ, ta thay đổi hình đại hiện với file định dạng không cho phép: <https://kduong.atlassian.net/browse/SW-4>

A screenshot of a computer

Description automatically generated

Hình 74: Bug Sai định dạng file hình đại diện

e) Chỉnh sửa bio trên trang cá nhân không có thông báo

Trên trang chỉnh sửa hồ sơ, khi ta thay đổi bio không có thông báo thành công hay thất bại: <https://kduong.atlassian.net/browse/SW-5>

A screenshot of a computer

Description automatically generated

Hình 75: Bug Chỉnh sửa bio trên trang cá nhân không có thông báo

f) Chỉnh sửa thông tin cá nhân về cài đặt thông báo

Khi nhấn vào “Push notifications” trong “Setting” thì không hoạt động và không chuyển đến trang cài đặt thông báo: <https://kduong.atlassian.net/browse/SW-6>

A screenshot of a computer

Description automatically generated

Hình 76: Bug Chỉnh sửa thông tin cá nhân về cài đặt thông báo

g) Kiểm thử trường hợp vừa đăng nhập vào nhanh tay nhấn vào tin nhắn

Sau khi vừa đăng nhập vào, ở màn hình home nhanh tay nhấn vào menu tin nhắn và chọn đoạn chat sẽ xuất hiện lỗi: <https://kduong.atlassian.net/browse/SW-7>

A screenshot of a computer

Description automatically generated

Hình 77: Bug Nhanh tay nhấn vào tin nhắn

h) Kiểm thử trường hợp gửi tin nhắn chỉ có ký tự khoảng cách

Khi nhập ký tự khoảng trắng vào ô nhắn tin thì hiển thị nút cho phép gửi tin nhắn sau đó nhấn gửi thì tin nhắn được gửi đi: <https://kduong.atlassian.net/browse/SW-8>

A screenshot of a computer

Description automatically generated

Hình 78: Bug Gửi tin nhắn chỉ có ký tự khoảng cách

i) Kiểm thử chức năng xem duyệt, từ chối yêu cầu kết bạn

Khi duyệt lời mời kết bạn web không cập nhật lại số bạn bè và danh sách bạn bè: <https://kduong.atlassian.net/browse/SW-9>

A screenshot of a computer

Description automatically generated

Hình 79: Bug Cập nhật lại số bạn bè và danh sách bạn bè

k) Thay đổi mật khẩu với mật khẩu mới giống mật khẩu hiện tại

Khi ta thay đổi mật khẩu với mật với mới giống với mật khẩu hiện tại thì không xuất hiện cảnh báo lỗi mà vẫn thực hiện thay đổi mật khẩu và thông có nút ẩn hiện mật khẩu: <https://kduong.atlassian.net/browse/SW-10>

A screenshot of a computer

Description automatically generated

Hình 80: Bug Thay đổi mật khẩu với mật khẩu mới giống mật khẩu hiện tại

l) Thay đổi mật khẩu với mật khẩu mới có ít ký tự

Khi ta thay đổi mật khẩu với mật khẩu mới ít ký tự thì không xuất hiện cảnh báo lỗi mà chỉ hiện lỗi code: <https://kduong.atlassian.net/browse/SW-11>

A screenshot of a computer

Description automatically generated

Hình 81: Bug Thay đổi mật khẩu với mật khẩu mới có ít kí tự

m) Thông báo đến chủ bài viết khi chủ bài viết bình luận bài viết của mình

Thông báo tại mục notifications khi ta tự bình luận vào bài viết của mình: <https://kduong.atlassian.net/browse/SW-12>

A screenshot of a computer

Description automatically generated

Hình 82: Bug Thông báo khi bình luận bài viết của chính mình

# **CHƯƠNG 5: CÀI ĐẶT DEMO**

## **5.1. Các yêu cầu về môi trường cài đặt và phiên bản phần mềm**

* Ngôn ngữ lập trình: JavaScript.
* Hệ thống cơ sở dữ liệu: MongoDB.
* Quản lý mã nguồn: Github.
* Hệ quản trị cơ sở dữ liệu: MongoDB Atlas.
* Thiết kế Usecase và biểu đồ: Draw.io.
* Phiên bản MongoDB: 5.0.14.
* Phiên bản ExpressJS: 4.17.1.
* Phiên bản ReactJS: 18.2.0.
* Phiên bản NodeJS: 20.15.1.

## **5.2. Cài đặt Server**

Cài đặt các package dành cho Server: bcryptjs(2.4.3), body-parser(1.20.2), compression(1.7.4), connect-mongo(1.4.6), canvas(2.11.2), cors(2.8.5), express(4.18.2), cookir-parser(1.4.6), csvtojson(2.0.10), express-session(1.18.0), express-validator(7.0.1), face-api.js(0.22.2), helmet(7.1.0), jsonwebtoken(9.0.2), mongoose(7.6.3), mongoose-unique-validator(4.0.0), multer(1.4.5-lts.1), nodemailer(6.9.7), node-stream-zip(1.15.0), node-unrar-js(2.0.2), xlsx(0.18.5)

Tổ chức các thư mục và file:

A screenshot of a computer program

Description automatically generated

Hình 83: Tổ chức thư mục, file Server

Bao gồm:

* controllers: để chứa code các function xử lý cho các routes.
* middleware: chứa các middleware (Middleware là các hàm khác nhau được gọi bởi [Route Express] trước khi các yêu cầu hoàn tất.).
* models: dùng để cung cấp dữ liệu, thực hiện kết nối, trích lọc, chèn, chỉnh sửa dữ liệu trong database, tương tác với file system, network.
* routes: dùng để chuyển hướng các URL đến các hàm xử lý tương ứng.
* env: chứa các tham số cấu hình.
* server.js: tạo kết nối đến express, mongoDB và socket.io.
* socketServer.js: chứa các phương thức của socket.io.

## **5.3. Cài đặt Client**

Cài đặt các package dành cho Client: @babel/plugin-proposal-private-property-in-object(7.14.5), @emotion/react(11.11.1), @emotion/styled(11.11.0), @mui/icons-material (5.14.11), @testing-library/jest-dom (5.11.9), @testing-library/react (11.2.3), @testing-library/user-event(12.6.2), @mui/material(5.14.13), @mui/lab(5.0.0-alpha.169), @mui/x-date-pickers(7.0.0), axios(1.5.1), chart.js(4.4.0), classnames(2.3.2), date-fns(2.30.0), emoji-picker-react(4.5.7), face-api.js(0.22.2), resolve-url-loader(5.0.0), install(0.13.0), moment(2.29.4), npm(10.5.0), react (18.2.0), react-toastify(9.1.3), react-dom (18.2.0), react-moment (1.1.3), react-otp-input(3.1.0), react-router-dom(6.16.0), react-scripts(5.0.1), react-slick(0.30.2), react-toastify (9.1.3), sass(1.68.0), sass-loader(13.3.2), simplebar-react(3.2.4), socket.io-client (4.7.2), web-vitals (2.1.4), slick-carousel(1.8.1), xlsx(0.18.5), xlsx-populate(1.21.0).

Tổ chức các thư mục và file:

A screenshot of a computer

Description automatically generated

Hình 84: Tổ chức thư mục, file Client

Bao gồm:

* components: chứa các thành phần giao diện có thể tái sử dụng.
* config: chứa kết nối firebase.
* routes: chứa các đường dẫn của trang web tới các trang.
* pages: những trang giao diện chính của ứng dụng như trang chủ, trang nhắn tin, ....
* context: chứa các actions và reducers để quản lí các state.
* services: chứa các file gọi api của trang web.
* utils: chứa các file tiện ích hỗ trợ quá trình cấu hình, gửi yêu cầu http, xác thực.
* hook: chứa các file có các hàm quản lý state.
* App.js: file chính chứa các route và dùng để render page.
* socketClient.js: chứa các phương thức của socket.io-client.

## **5.4. Sử dụng Docker để deploy trên môi trường production**

- Bước 1: Xây dựng Docker file.

A screenshot of a computer program

Description automatically generated

Hình 85: Docker Deploy

* Bước 2: Sử dụng lệnh “docker build -t thanhdanh/express-app .”, để xây dựng Docker Image.
* Bước 3: Sử dụng lệnh “docker push thanhdanh/express-app:latest” để đẩy Docker Image lên Docker Hub.
* Bước 4: Sử dụng đường dẫn của Docker image và dùng service cung cấp bởi render.com để hosting BE.

Truy cập: <https://nestme-ins.onrender.com/> để sử dụng website.

# **PHẦN 3: KẾT LUẬN**

## **Kết quả đạt được**

Đề tài đã đạt được kết quả theo mục tiêu đã đề ra, học thêm những kiến thức, những kỹ năng mới, công nghệ mới và áp dụng những điều đấy để xây dựng nên website hỗ trợ thanh tra trong các kỳ thi:

* + Nâng cao khả năng làm việc nhóm
  + Học thêm các công nghệ mới: Reactjs, Nodejs
  + Học cách quản lý source control và bug
  + Nhận diện khuôn mặt với thư viện face-api.js
  + Thao tác với file ảnh, excel, ...
  + Sử dụng jwt, bcrypt để bảo mật
  + Gửi mail xác nhận

1. **Danh sách các chức năng đã làm được**
   * Chức năng dành cho admin:

**+** Đăng nhập, đăng xuất.

+ Thống kê (người đóng góp nhiều nhất theo dạng bảng, số người dùng và bài viết so với ngày hôm qua theo dạng biểu đồ)

+ Quản lý người dùng (xem thông tin người dùng, xem báo cáo và ban user)

+ Quản lý bài viết (xem thông tin bài viết, xem báo cáo và ẩn bài viết)

* + Chức năng dành cho thanh tra:

**+** Đăng nhập và đăng xuất.

+ Tìm kiếm và xem thông tin sinh viên

+ Tạo, chỉnh sửa và xem thông tin báo cáo

+ Xem thông tin phòng thi

+ Cảnh báo sinh viên nghi vấn

+ Điểm danh khuôn mặt sinh viên bằng camera hoặc hình ảnh

+ Xuất file excel thông tin của một phòng thi hoặc các phòng trong một ngày thi

## **Ưu điểm và khuyết điểm**

**3.1. Ưu điểm**

* + Tìm hiểu và sử dụng thành công các công nghệ vào đề tài
  + Thiết kế giao diện dễ nhìn, thân thiện với người dùng
  + Tương đối đầy đủ các chức năng dùng để hỗ trợ cho thanh tra
  + Chức năng hoạt động tương đối ổn định
  + Bảo mật tương đối tốt khi dùng jwt và email để xác thực và mã hóa mật khẩu
  + Có chức năng xử lý file, nhận diện điểm danh nhanh chóng

**3.2. Khuyết điểm**

* + Chức năng train hình ảnh còn khá chậm
  + Việc xử lý logic và hiệu suất vẫn chưa tối ưu
  + Nghiệp vụ quản lý chưa hoàn thiện

## **Hướng phát triển**

Một số chức năng vẫn chưa được tối ưu nên cần xử lý logic các chức năng để website hoạt động tốt và nhanh nhất có thể. Bên cạnh đó công nghệ hỗ trợ tải của website vẫn còn yếu, cần một phương pháp có thể giải quyết vấn đề đó. Từ đó chúng em mong muốn tìm hiểu phương pháp giúp hệ thống mạnh mẽ, ổn định hơn với đầy đủ các chức năng để có thể hỗ trợ thanh tra trong các kỳ thi một cách tốt nhất.

# **DANH MỤC TÀI LIỆU THAM KHẢO**

* + - 1. <https://topdev.vn/blog/express-js-la-gi/>
      2. <https://tutsmodel.blogspot.com/2015/04/gioi-thieu-ve-node-js.html>
      3. <https://topdev.vn/blog/node-js-la-gi/>
      4. <https://t3h.com.vn/tin-tuc/mern-stack-la-gi>
      5. <https://topdev.vn/blog/mongodb-la-gi/>
      6. <https://viblo.asia/p/reactjs-uu-diem-va-nhuoc-diem-V3m5WzexlO7>
      7. <https://chat.openai.com/>