

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG



BÁO CÁO
THỰC TẬP CƠ SỞ

Giảng viên: Kim Ngọc Bách

Sinh viên:

Nguyễn Việt Huy

B21DCCN437

I. Giới thiệu về dự án

- Dự án Instagram Clone là dự án cá nhân về trang web mạng xã hội được lấy ý tưởng dựa trên trang web mạng xã hội được ưa thích Instagram.
- Mạng xã hội Instagram cho phép người dùng có thể đăng tải hình ảnh, video, tương tác với bài viết của nhau bằng cách thích (like), bình luận (comment) và chia sẻ (share). Người dùng có thể tạo hồ sơ cá nhân, theo dõi bạn bè, nhắn tin trực tiếp và khám phá nội dung từ cộng đồng.

II. Thiết lập dự án & Thiết kế cấu trúc

1. **Mục tiêu:** Thiết lập môi trường phát triển, lựa chọn công nghệ phù hợp, lên ý tưởng và thiết kế sơ bộ cơ sở dữ liệu.
2. **Frontend:**
 - o Khởi tạo dự án bằng React.js và TypeScript để đảm bảo khả năng mở rộng và kiểm soát kiểu dữ liệu tốt hơn.
 - o Cài đặt Tailwind CSS để hỗ trợ thiết kế UI nhanh chóng, đồng nhất.
 - o Thiết lập Redux Toolkit nhằm quản lý state tập trung, giúp các component dễ dàng truy xuất và thay đổi dữ liệu.
3. **Backend:**
 - o Tạo server Express cơ bản, thiết lập middleware (CORS, bodyParser, Logging).
 - o Kết nối MongoDB thông qua thư viện Mongoose, xây dựng các schema cơ bản: User, Post.
4. **Thành tựu:** Dự án đã có kiến trúc rõ ràng giữa frontend và backend, phân chia thư mục hợp lý, có thể sẵn sàng tiếp nhận các chức năng phức tạp hơn trong các Sprint tiếp theo.

III. Sprint 2 (25/3 - 31/3): Đăng ký/Đăng nhập & Giao diện cơ bản

1. **Authentication:**
 - Tạo form đăng ký và đăng nhập có xác thực dữ liệu đầu vào.
 - Mã hoá mật khẩu người dùng bằng bcrypt để tăng cường bảo mật.
 - Tạo JWT Access Token và Refresh Token, hỗ trợ cơ chế đăng nhập bảo mật cao.
 - Xử lý tự động refresh token khi token hết hạn.

2. Social Login:

- Tích hợp đăng nhập bằng Facebook
- Backend xác minh token, tự động tạo tài khoản nếu chưa có người dùng.

3. Giao diện:

- Tạo giao diện đăng nhập, đăng ký.
- Bố cục trang chủ với các vùng layout chính như navbar, feed, sidebar.

4. Thành tựu: Người dùng đã có thể đăng ký, đăng nhập, lưu trữ token và điều hướng đến các trang chính của ứng dụng.

IV. Hệ thống Follow & Đăng bài

1. Giao diện:

- Giao diện trang cá nhân được bổ sung thêm nút "Follow/Unfollow" khi truy cập vào trang của người dùng khác.
- Giao diện đăng bài bao gồm khung mô tả nội dung, nút chọn ảnh và nút đăng bài.
- Mỗi bài viết hiển thị ảnh, mô tả, số lượt thích và bình luận, cùng avatar và tên người đăng.
- Feed chính hiển thị danh sách bài viết từ những người mà người dùng đang theo dõi.

2. Logic hoạt động:

- **Follow System:**
 - Khi người dùng bấm nút "Follow", frontend gửi request đến backend với thông tin user cần follow.
 - Backend cập nhật mảng followers của user được theo dõi và followings của người dùng hiện tại.
 - API trả về danh sách cập nhật mới và phản hồi trạng thái để cập nhật UI ngay.
- **Post Management:**
 - Ảnh được chọn trên giao diện frontend sẽ được upload lên Firebase Storage.
 - Sau khi upload thành công, URL ảnh được gửi cùng thông tin mô tả lên backend.

- Backend tạo bản ghi bài viết mới, liên kết với ID người đăng và trả về dữ liệu bài viết vừa tạo để hiển thị lên feed.
- **Feed:**
 - Người dùng chỉ thấy bài viết từ những người họ đang follow. Backend thực hiện truy vấn bằng cách lấy danh sách followings của user, sau đó lọc bài viết tương ứng theo thứ tự thời gian mới nhất.
- **Tương tác:**
 - Khi người dùng bấm like/unlike bài viết, frontend gửi request đến backend để thêm hoặc xóa ID người dùng khỏi danh sách like trong bài viết.
 - Đối với bình luận, người dùng gửi nội dung bình luận cùng ID bài viết. Backend lưu vào mảng comments của bài viết và phản hồi lại với nội dung vừa bình luận, thời gian tạo, ảnh đại diện và tên người dùng.
- 3. **Thành tựu:**
 - Tính năng mạng xã hội cơ bản đã được triển khai hoàn thiện.
 - Người dùng có thể theo dõi nhau, đăng ảnh, tương tác với bài viết.

V. Notifications & Tìm kiếm

1. Giao diện:

- Khi người dùng nhận thông báo mới (like, comment, follow), biểu tượng chuông trên navbar sẽ hiển thị dấu hiệu có thông báo chưa đọc.
- Giao diện hiển thị danh sách thông báo dạng list, mỗi thông báo có avatar, tên người thực hiện hành động và mô tả ngắn.
- Giao diện tìm kiếm cho phép người dùng nhập từ khóa và hiện kết quả user khớp theo tên hoặc username.

2. Logic hoạt động:

- **Notifications:**
 - Khi có sự kiện like, comment hoặc follow, client gửi sự kiện lên server thông qua Socket.IO.
 - Server nhận sự kiện, tạo thông báo mới trong MongoDB với thông tin: user nhận, user gửi, loại thông báo và thời

gian.

- Nếu người nhận đang online, server gửi socket event ngay lập tức để client hiển thị realtime.
- Nếu không online, client sẽ nhận được thông báo khi đăng nhập lại.

- **Search:**

- Người dùng nhập từ khóa trong ô tìm kiếm (ví dụ: "huynh"), frontend gửi request đến API `/search/users?q=huynh`.
- Backend sử dụng MongoDB Text Index để tìm các user có tên hoặc username khớp từ khóa.
- Kết quả trả về gồm danh sách user rút gọn (avatar, username, full name) để hiển thị ngay trong dropdown.

3. Thành tựu:

- Người dùng đã có thể nhận thông báo theo thời gian thực và tìm kiếm user nhanh chóng.
- Tăng tính tương tác, tạo cảm giác kết nối giữa các tài khoản trong hệ thống.

VI. Realtime Chat

1. Giao diện:

- Giao diện chat giống ứng dụng Messenger: chia làm hai phần chính gồm sidebar danh sách người đã theo dõi và khung hội thoại.
- Khung tin nhắn hiển thị thời gian, người gửi và nội dung rõ ràng, tin nhắn mới sẽ được tự động cuộn xuống dưới.
- Cho phép gửi tin nhắn bằng cách bấm nút gửi hoặc nhấn phím Enter.

2. Logic hoạt động:

- **Kết nối Socket.IO:**

- Khi người dùng đăng nhập thành công, frontend gửi sự kiện "addUser" chứa userId để server lưu socket.id tương ứng vào Map onlineUsers.
- Khi một user gửi tin nhắn, socket phát sự kiện sendMessage với senderId, receiverId và nội dung tin nhắn (content).
- Backend nhận được tin nhắn sẽ lưu vào MongoDB và kiểm tra xem người nhận có đang online không:

- Nếu **online**, server gửi trực tiếp tin nhắn qua socket receiverId.
- Nếu **offline**, tin nhắn vẫn được lưu vào DB và sẽ được lấy lại khi user mở lại trang chat.

- **Database:**

- Tin nhắn được lưu trong collection messages với các trường: senderId, receiverId, content, createdAt.
- Lịch sử chat giữa hai người được truy vấn bằng cách tìm những tin nhắn có cặp {senderId, receiverId} và sắp xếp theo createdAt tăng dần.

3. **Thành tựu:** Hệ thống nhắn tin thời gian thực hoạt động ổn định, giúp người dùng giao tiếp thuận tiện, phản hồi tức thì như các ứng dụng trò chuyện hiện đại.

VII. Trang cá nhân & Tối ưu hiệu năng

1. Giao diện:

- Trang cá nhân có giao diện rõ ràng: hiển thị avatar, tên người dùng, bio, số lượng bài viết, followers và followings.
- Nút chỉnh sửa hồ sơ cho phép người dùng cập nhật ảnh đại diện, mô tả ngắn (bio).
- Các bài viết được hiển thị dưới dạng lưới (grid), hỗ trợ xem chi tiết bài viết khi bấm vào.

2. Logic hoạt động:

- Trang cá nhân được lấy thông qua API theo userId với các thông tin: avatar, fullName, bio, followers, following, posts.
- Khi chỉnh sửa thông tin, người dùng chọn ảnh mới và nhập bio, frontend gửi form-data tới backend.
- Backend cập nhật thông tin và phản hồi để cập nhật lại UI.

3. Tối ưu hiệu năng:

- Cài đặt phân trang (pagination) cho feed bài viết và lịch sử tin nhắn.

4. **Thành tựu:** Trang cá nhân hoàn chỉnh.

VIII. Testing & Deploy

1. Testing:

- Hiện chưa có đủ thời gian để thực hiện các testing

2. Deploy:

- Tối ưu hóa cấu hình Vercel để deploy frontend.
- Sử dụng Heroku hoặc EC2 để triển khai backend API.
- MongoDB Atlas dùng làm cơ sở dữ liệu đám mây.
- Cấu hình HTTPS, CORS và domain để đảm bảo bảo mật khi deploy.

3. Thành tựu: Dự án hoạt động ổn định ở môi trường thật.

IX. Tổng kết

- Dự án Instagram Clone là thành quả của cả học kì học hỏi và làm việc liên tục theo mô hình Agile. Qua từng Sprint, các chức năng ngày càng hoàn thiện, giao diện người dùng được cải tiến, backend được tối ưu hóa và hỗ trợ các tính năng thời gian thực. Đây là một nền tảng hoàn chỉnh có thể phát triển thành ứng dụng thực tế.
- Tuy nhiên hiện tại ứng dụng vẫn còn nhiều điểm còn hạn chế và nhiều năng vẫn còn để phát triển thêm trong tương lai
- **Những điểm nổi bật:**
 - Áp dụng công nghệ hiện đại: React, TypeScript, Redux Toolkit, Express, MongoDB, Socket.IO.
 - Đảm bảo bảo mật qua JWT, hash mật khẩu.
 - Tính năng realtime: chat, thông báo.
- **Định hướng tương lai:**
 - Tối ưu các chức năng hiện đang có.
 - Phát triển responsive mobile.
 - Các tính năng còn thiếu của Instagram gốc: video call, đăng tải story, ...
- **Kết luận:** Dự án đạt mục tiêu về chức năng, kỹ thuật và trải nghiệm người dùng. Đây là sản phẩm bước đệm lý tưởng để học hỏi, thực hành kỹ năng fullstack và có thể triển khai thực tế trong các dự án lớn hơn.