**TRANG BÌA CHÍNH**

|  |
| --- |
| BỘ GIÁO DỤC VÀ ĐÀO TẠO  **TRƯỜNG ĐẠI HỌC PHENIKAA**  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  A blue text on a black background  AI-generated content may be incorrect.  **BÁO CÁO TỔNG KẾT**    **TÊN ĐỀ TÀI: ỨNG DỤNG NHẮN TIN-**  **CLONE MESSENGER**  Lĩnh vực:  Chuyên ngành: Công nghệ thông tin  **Nhóm: 2 người**   * Sinh viên 1: Thái Trung Thiện – MSSV: 23010212   Sinh viên 2: Vũ Quốc Toản – MSSV: 23010003   * Giảng viên hướng dẫn: Trương Anh Hoàng   BỘ GIÁO DỤC VÀ ĐÀO TẠO  **TRƯỜNG ĐẠI HỌC PHENIKAA**  **BÁO CÁO TỔNG KẾT**  **TÊN ĐỀ TÀI:**  **ỨNG DỤNG NHẮN TIN-**  **CLONE MESSENGER**  Lĩnh vực:  Chuyên ngành: Công nghệ thông tin  **Nhóm: 2 người**   * Sinh viên 1: Thái Trung Thiện – MSSV: 23010212   Sinh viên 2: Vũ Quốc Toản – MSSV: 23010003   * Giảng viên hướng dẫn: Trương Anh Hoàng   **Hà Nội, tháng năm 2022** |

**TRANG BÌA PHỤ**

**MỤC LỤC**

1. Mục tiêu đề tài
2. Phân tích yêu cầu hệ thống
3. Thiết kế hệ thống
4. Cài đặt
5. Kiểm thử
6. Đánh giá – Kết luận
7. Hướng phát triển
8. Tài liệu tham khảo

**1. MỤC TIÊU ĐỀ TÀI**

Clone Messenger là một ứng dụng nhắn tin cơ bản mô phỏng một phần tính năng của Facebook Messenger. Đề tài hướng đến việc giúp sinh viên hiểu quy trình phát triển phần mềm theo mô hình Agile và áp dụng kỹ năng lập trình, thiết kế UI/UX, kết nối cơ sở dữ liệu và sử dụng Socket hoặc API.

Chức năng chính:

* Đăng ký/đăng nhập người dùng
* Gửi/nhận tin nhắn thời gian thực
* Hiển thị danh sách bạn bè
* Tìm kiếm người dùng
* Giao diện thân thiện, dễ sử dụng

**2. PHÂN TÍCH YÊU CẦU HỆ THỐNG**

**2.1. Yêu cầu chức năng:**

* Đăng ký tài khoản mới
* Đăng nhập với tài khoản đã có
* Gửi và nhận tin nhắn văn bản
* Hiển thị danh sách bạn bè
* Hiển thị thời gian gửi tin nhắn
* Trạng thái online/offline của người dùng

**2.2. Yêu cầu phi chức năng:**

* Giao diện đơn giản, dễ sử dụng
* Phản hồi nhanh khi gửi/nhận tin nhắn
* Bảo mật thông tin người dùng (mã hóa mật khẩu)
* Khả năng mở rộng thêm tính năng

**3. THIẾT KẾ HỆ THỐNG**

**3.1 Kiến trúc hệ thống:**

* **Frontend**: ReactJS / Flutter
* **Backend**: NodeJS + Express
* **Realtime**: Socket.IO
* **Cơ sở dữ liệu**: MongoDB

**3.2 Sơ đồ ca sử dụng (Use Case Diagram):**

* Người dùng có thể:
  + Đăng ký, đăng nhập
  + Xem danh sách bạn
  + Chọn người để chat
  + Gửi tin nhắn
  + Nhận tin nhắn

**3.3 Giao diện người dùng:**

* Trang đăng nhập/đăng ký
* Trang danh sách bạn bè
* Giao diện chat thời gian thực

**4. CÀI ĐẶT**

**4.1 Công nghệ sử dụng:**

* **ReactJS (Frontend)**
* **NodeJS + Express (Backend)**
* **MongoDB (Database)**
* **Socket.IO (Realtime communication)**
* **JWT (Xác thực người dùng)**

**4.2 Cấu trúc dự án:**

bash

Sao chépChỉnh sửa

/client – React app

/server – NodeJS + Socket.IO

/database – MongoDB Atlas

**4.3 Một số đoạn mã tiêu biểu:**

**Gửi tin nhắn qua Socket:**

javascript

Sao chépChỉnh sửa

socket.emit("sendMessage", {

senderId: currentUser.\_id,

receiverId: selectedUser.\_id,

text: message,

});

**Backend nhận tin nhắn:**

javascript

Sao chépChỉnh sửa

socket.on("sendMessage", ({ senderId, receiverId, text }) => {

io.to(users[receiverId]).emit("getMessage", { senderId, text });

});

**5. KIỂM THỬ**

**5.1 Kiểm thử chức năng:**

* Đăng nhập: ✅
* Gửi tin nhắn: ✅
* Nhận tin nhắn real-time: ✅
* Hiển thị danh sách bạn: ✅

**5.2 Kiểm thử phi chức năng:**

* Thời gian phản hồi: < 300ms
* Dung lượng gửi: nhỏ hơn 10KB/tin
* Tốc độ tải trang: ~1 giây

**6. ĐÁNH GIÁ – KẾT LUẬN**

**6.1 Kết quả đạt được:**

* Clone được giao diện và chức năng cơ bản của ứng dụng Messenger
* Hiểu rõ mô hình client-server và kỹ thuật giao tiếp real-time
* Thực hành chia nhóm, phân công công việc, quản lý tiến độ

**6.2 Hạn chế:**

* Chưa hỗ trợ gửi hình ảnh
* Chưa có thông báo khi có tin nhắn mới từ người khác
* Giao diện còn đơn giản, chưa có hiệu ứng

**7. HƯỚNG PHÁT TRIỂN**

* Thêm tính năng gửi ảnh, biểu tượng cảm xúc
* Tích hợp thông báo đẩy (push notification)
* Lưu trữ tin nhắn vĩnh viễn trong cơ sở dữ liệu
* Xây dựng phiên bản mobile với React Native hoặc Flutter