

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**  
**CƠ SỞ THÀNH PHỐ HỒ CHÍ MINH**  
**KHOA CÔNG NGHỆ THÔNG TIN 2**

∞Ω∞

**BÁO CÁO KIỂM THỬ VÀ CHẤT LƯỢNG**  
**ỨNG DỤNG**

**HỌC PHẦN: NHẬP MÔN CÔNG NGHỆ PHẦN MỀM**  
**ĐỀ TÀI: Web App Quản Lý Chấm Công Nhân Viên**

Nhóm sinh viên:

Nhóm 3

Phụ trách kiểm thử:

Nguyễn Ngọc Gia Hân

Giảng viên hướng dẫn đồ án:

Châu Văn Vân

***TP. HCM 12/2025***

## MỤC LỤC

<b>LỜI MỞ ĐẦU .....</b>	<b>4</b>
<b>I. THÔNG TIN DỰ ÁN .....</b>	<b>5</b>
1. Tổng quan dự án .....	5
2. Yêu cầu chức năng.....	5
<b>II. CÔNG CỤ KIỂM THỬ .....</b>	<b>6</b>
1. Jest – framework kiểm thử đơn vị.....	6
a. Lý do lựa chọn Jest .....	6
b. Jest trong dự án.....	6
c. Vai trò của Jest trong dự án .....	8
2. Selenium – Kiểm thử UI.....	8
a. Lý do lựa chọn Selenium .....	8
b. Selenium trong dự án.....	8
c. Vai trò của Selenium trong dự án .....	9
3. Postman – Kiểm thử API.....	10
a. Lý do lựa chọn Postman .....	10
b. Postman trong dự án.....	10
c. Vai trò của Postman trong dự án .....	10
<b>III. THỰC HIỆN KIỂM THỬ .....</b>	<b>11</b>
1. Kế hoạch kiểm thử.....	11
a. Mục tiêu kiểm thử .....	11
b. Phạm vi kiểm thử .....	11
c. Phương pháp kiểm thử .....	11
d. Môi trường kiểm thử .....	12
e. Lịch trình kiểm thử.....	12
f. Tiêu chí bắt đầu và kết thúc kiểm thử .....	13
2. Test Case .....	13
a. Kiểm thử logic – Jest.....	13
b. Kiểm thử giao diện – Selenium .....	16

c. Kiểm thử API – Postman.....	18
3. Kết quả kiểm thử.....	22
a. Postman.....	22
b. Jest.....	31
c. Selenium.....	39
<b>IV. ĐÁNH GIÁ, CẢI TIẾN .....</b>	<b>45</b>
1. Tổng Quan Chiến Lược Kiểm Thử .....	45
2. Đánh Giá Chi Tiết.....	45
a. Tầng Unit & Integration Test (Backend - Jest).....	45
b. Tầng API Test (Postman).....	46
c. Tầng E2E Test (Frontend - Selenium).....	46
4. Đề Xuất Cải Tiến.....	47
a. Cải thiện quản lý dữ liệu (Data Management).....	47
b. Tăng cường Automation cho Postman.....	47
c. Nâng cấp E2E Test .....	47
d. Bổ sung Performance Test (Nâng cao) .....	47
<b>V. TỔNG KẾT .....</b>	<b>47</b>
1. Tổng Quan Quá Trình Kiểm Thử.....	47
2. Đánh Giá Chất Lượng Dự Án.....	48
<b>LỜI KẾT .....</b>	<b>50</b>
<i>Tài liệu tham khảo .....</i>	<i>50</i>

## LỜI MỞ ĐẦU

Trong quá trình phát triển phần mềm, hoạt động kiểm thử (Testing) đóng vai trò quan trọng nhằm đảm bảo hệ thống vận hành đúng chức năng, ổn định và đáp ứng yêu cầu nghiệp vụ. Đối với dự án Mini App Chấm Công Nhân Viên, việc kiểm thử càng trở nên cần thiết bởi hệ thống liên quan đến dữ liệu thời gian thực, vị trí GPS, thông tin nhân viên và các thao tác mang tính nhạy cảm như Check-in, Check-out. Nếu không được kiểm thử đầy đủ, hệ thống có thể phát sinh sai lệch dữ liệu chấm công, ảnh hưởng trực tiếp đến hiệu suất làm việc và quá trình tính lương của doanh nghiệp.

Mục tiêu của quá trình kiểm thử trong dự án này là:

- Xác minh độ chính xác của các API chấm công.
- Đảm bảo giao diện người dùng hoạt động ổn định, dễ sử dụng và hiển thị đúng dữ liệu.
- Kiểm tra logic xử lý thời gian làm việc, điều kiện check-in/check-out và định vị GPS.
- Phát hiện, ghi nhận và đề xuất hướng khắc phục các lỗi có thể xảy ra trước khi triển khai thực tế.

Để đảm bảo chất lượng toàn diện, báo cáo sử dụng ba công cụ kiểm thử phổ biến:

1. **Postman** – dùng để kiểm thử API (API Testing), đảm bảo các endpoint như Login, Check-in, Check-out và Báo cáo trả về đúng dữ liệu và status code.
2. **Selenium** – dùng cho kiểm thử giao diện (UI Testing), mô phỏng thao tác người dùng trên trình duyệt nhằm kiểm tra độ ổn định, tính tương tác và tính chính xác của các trang chấm công và báo cáo.
3. **Jest** – dùng để kiểm thử đơn vị (Unit Testing), đảm bảo các hàm xử lý logic như tính giờ làm, xác thực điều kiện check-in/check-out và kiểm tra vị trí GPS hoạt động đúng trong mọi trường hợp.

Thông qua việc kết hợp cả ba công cụ trên, quá trình kiểm thử giúp đánh giá toàn diện hệ thống từ backend, frontend đến logic xử lý, góp phần nâng cao chất lượng và độ tin cậy của ứng dụng chấm công.

# I. THÔNG TIN DỰ ÁN

## 1. Tổng quan dự án

**Tên hệ thống:** Hệ thống Chấm công & Quản lý Nhân sự (Timekeeping & HR Management System).

Dự án Mini App Chấm Công Nhân Viên được xây dựng nhằm hỗ trợ doanh nghiệp quản lý thời gian làm việc của nhân viên một cách chính xác và tiện lợi. Hệ thống cho phép nhân viên thực hiện Check-in/Check-out thông qua quét khuôn mặt và vị trí GPS, đồng thời cung cấp giao diện báo cáo để theo dõi thời gian làm việc theo ngày, tuần và tháng. Mục tiêu chính của dự án là tối ưu hóa quá trình chấm công thủ công truyền thống, giảm sai sót và nâng cao tính minh bạch trong việc quản lý nhân sự.

Ứng dụng gồm ba thành phần chính:

- Frontend Web: giao diện người dùng, hỗ trợ thao tác chấm công và xem báo cáo. Chuyển đổi giữa tài khoản người dùng và người quản lý.
- Backend API: cung cấp các dịch vụ xử lý xác thực người dùng, ghi nhận thời điểm check-in/check-out và truy xuất báo cáo.
- Cơ sở dữ liệu: lưu trữ thông tin nhân viên, phiên chấm công và lịch sử làm việc, quản lý ca làm và quản lý hệ thống nhân viên.

Phạm vi của dự án tập trung vào:

- Quét mặt để xác thực nhân viên.
- Ghi nhận tọa độ GPS trong mỗi phiên chấm công.
- Tính toán thời gian làm việc trong ngày.
- Xuất báo cáo theo từng khoảng thời gian trong tháng.

Việc kiểm thử dự án là bước quan trọng nhằm đảm bảo hệ thống hoạt động đúng như yêu cầu, dữ liệu chấm công được ghi nhận chính xác và giao diện đáp ứng tốt nhu cầu sử dụng của nhân viên và quản trị viên.

## 2. Yêu cầu chức năng

- Đăng nhập hệ thống bằng tài khoản nhân viên.
- Check-in:
  - + Xác thực qua camera (Face Scan).
  - + Ghi nhận vị trí GPS tại thời điểm check-in.
- Check-out:
  - + Lưu thời gian kết thúc ca làm.
  - + Kiểm tra điều kiện hợp lệ (ví dụ: không check-out trước khi check-in).

- Xem báo cáo:
  - + Báo cáo theo ngày/tháng.
  - + Thống kê thời gian làm việc
- Quản lý thông tin nhân viên, thông tin ca làm.

## II. CÔNG CỤ KIỂM THỬ

### 1. Jest – framework kiểm thử đơn vị (Unit Test)

- Jest là một framework kiểm thử JavaScript phổ biến, được phát triển bởi Meta (Facebook) và được sử dụng rộng rãi trong các dự án web hiện đại. Jest được thiết kế chuyên biệt cho việc kiểm thử logic xử lý (Unit Testing) và hoạt động hiệu quả với các ứng dụng Node.js cũng như các thư viện frontend như React.
- Trong dự án Mini App Chấm Công Nhân Viên, Jest được sử dụng để kiểm thử các hàm logic chấm công, bao gồm: kiểm tra điều kiện check-in/check-out, tính toán thời gian làm việc, và xác định tính hợp lệ của vị trí GPS.
- Bộ test suite hiện tại bao phủ các chức năng cốt lõi của hệ thống chấm công, bao gồm: xác thực (Auth), phân quyền (RBAC), nghiệp vụ chấm công (Attendance), quản lý ca làm việc (Shifts) và nhận diện khuôn mặt (FaceID).
- Kiểm thử được thực hiện theo mô hình **Mocking, Time Travel** giả lập các tương tác với cơ sở dữ liệu (Firebase Firestore) và các thư viện bên ngoài (jsonwebtoken) để đảm bảo tốc độ và tính cô lập của test case.

#### a. Lý do lựa chọn Jest

- Dễ cấu hình: cài đặt và sử dụng đơn giản, không yêu cầu nhiều thiết lập
- Hiệu năng cao: Jest chạy test rất nhanh nhờ cơ chế chạy song song.
- Tích hợp tốt với Node.js: phù hợp để kiểm thử các module logic backend.

#### b. Jest trong dự án

- **Tầng Middleware (Bảo mật & Phân quyền)**
- Các file: **auth.middleware.test.js**, **admin.middleware.test.js**, **selfOrAdmin.middleware.test.js**.
- Xác thực (Authentication): Kiểm tra tính hợp lệ của JWT Token (Header Bearer, Query param). Xử lý các trường hợp Token hết hạn, sai chữ ký, hoặc format không đúng. Hàm kiểm tra điều kiện check-out (check-out trước khi check-in...).

- Phân quyền Admin (Authorization): Đảm bảo chỉ các role được cấp phép (admin, System Admin, manager) mới truy cập được. Chặn các role không hợp lệ (user, employee) hoặc user null.
- Quyền truy cập dữ liệu (Data Privacy): Kiểm thử logic selfOrAdmin - cho phép User thường xem dữ liệu của chính mình, nhưng Admin có thể xem dữ liệu của bất kỳ ai.
- **Tầng Service (Nghiệp vụ cốt lõi)**
- Các file: **attendance.service.test.js, shifts.service.test.js, face.service.test.js**

#### **Attendance (Chấm công):**

- Logic Check-in/Check-out: Ngăn chặn check-in 2 lần trong ngày, tính toán giờ làm việc (workSeconds).
- Xử lý lịch sử & thống kê: Tính số ngày công, ngày nghỉ chính xác dựa trên dữ liệu giả lập
- Admin Update: Kiểm tra quyền sửa đổi bản ghi chấm công.

#### **Shifts (Ca làm việc):**

- CRUD: Tạo, sửa, xóa ca làm việc. Tự động sinh mã ca (CA001) và tên ca nếu thiếu.
- Phân ca (Assign): Gán ca cho một hoặc nhiều nhân viên, xử lý định dạng ngày tháng
- Validation: Bắt lỗi thiếu giờ bắt đầu/kết thúc, trùng lặp nhân viên trong ca.

#### **FaceID (Nhận diện):**

- Thuật toán: Tính khoảng cách Euclidean giữa các vector khuôn mặt (Embedding 128 chiều).
- GPS: Tính khoảng cách địa lý (Haversine) để đảm bảo nhân viên chấm công trong bán kính cho phép.
- Cấu hình: Kiểm tra việc đọc FACE\_THRESHOLD từ biến môi trường.
- **Tầng Controller (Xử lý Request/Response)**
- Các file: **attendance.controllers.test.js, face.controller.test.js**
- Đảm bảo Controller gọi đúng Service tương ứng.
- Kiểm tra định dạng phản hồi (Response format): Trả về JSON chuẩn { success: true, data: ... }
- Mã lỗi HTTP: Kiểm tra việc trả về đúng status code:
  - + 200: Thành công.
  - + 400: Lỗi dữ liệu đầu vào (thiếu param, sai format).
  - + 404: Không tìm thấy tài nguyên.

+ 500: Lỗi Server/Service.

- Điểm nổi bật: Test kỹ việc validate dữ liệu đầu vào tại controller (ví dụ: embedding phải là array, thiếu month trong query param).

### c. Vai trò của Jest trong dự án

- Đảm bảo độ chính xác của nghiệp vụ chấm công, tránh sai lệch giờ làm.
- Ngăn lỗi logic phát sinh khi cập nhật tính năng mới
- Tăng độ tin cậy của hệ thống vì logic cốt lõi đã được kiểm tra đầy đủ.
- Hỗ trợ Debug nhanh khi phát hiện lỗi liên quan đến quy trình xử lý dữ liệu.

## 2. Selenium – Kiểm thử UI

- Selenium là bộ công cụ tự động hóa kiểm thử giao diện người dùng (UI Testing) trên trình duyệt. Selenium mô phỏng các thao tác của người dùng như click, nhập liệu, chuyển trang và kiểm tra nội dung hiển thị trên web.
- Trong dự án Mini App Chấm Công Nhân Viên, Selenium được sử dụng để kiểm thử hai giao diện chính: **Giao diện chấm công** và **Giao diện báo cáo**. Đây là những phần người dùng thao tác trực tiếp nên việc kiểm thử UI đóng vai trò quan trọng để đảm bảo trải nghiệm mượt mà và không lỗi.
- Bộ test suite này thực hiện kiểm thử đầu cuối (End-to-End Testing), mô phỏng hành vi thực tế của người dùng trên trình duyệt để xác minh các luồng nghiệp vụ quan trọng nhất: Chấm công vào (Check-in), Chấm công ra (Check-out) và Xem báo cáo thống kê (Manager Report).

### a. Lý do lựa chọn Selenium

- Tự động hóa kiểm thử UI trên nhiều trình duyệt khác nhau.
- Mô phỏng thao tác thực của người dùng rất chính xác.
- Hỗ trợ đa ngôn ngữ lập trình như JavaScript, Java, Python,...
- Tích hợp tốt với quy trình CI/CD.
- Phù hợp để kiểm thử các luồng thao tác lặp đi lặp lại như chấm công hằng ngày.

### b. Selenium trong dự án

#### • Test checkin.test.js

- Mục tiêu: Xác minh quy trình chấm công khuôn mặt từ lúc đăng nhập đến khi dữ liệu được ghi nhận vào lịch sử.
- Kiểm thử các tác vụ:
  - + Xử lý quyền Camera: Chờ và xử lý trạng thái sẵn sàng của giao diện trước khi gọi Camera.



- + Kích hoạt FaceID: Nhấn nút Check-in -> Chờ Modal xuất hiện -> Kiểm tra Video stream hiển thị.
- + Xác nhận: Đóng Modal (giả lập nhận diện thành công) -> Chờ hệ thống đồng bộ dữ liệu.+ Kiểm tra dữ liệu hiển thị trên trang báo cáo.
- Điểm kỹ thuật: Sử dụng kỹ thuật đếm dòng (rows.length) để verify thay vì chỉ kiểm tra thông báo UI, giúp đảm bảo dữ liệu thực sự được lưu xuống Database.
- **Test checkout.test.js**
  - Mục tiêu: Xác minh quy trình chấm công ra và điều hướng trang.
  - Kiểm thử các tác vụ:
    - + Kích hoạt FaceID: Nhấn nút Check-out -> Kiểm tra Modal và Video stream hoạt động.
    - + Xử lý Modal: Giả lập thao tác đóng modal sau khi nhận diện xong -> Kiểm tra Modal đã biến mất (stalenessOf).
    - + Kiểm chứng UI: Tự động điều hướng sang Tab Lịch sử -> Xác minh bảng lịch sử chấm công (Table) có hiển thị và chứa dữ liệu.
  - Điểm kỹ thuật: Kiểm tra kỹ trạng thái của Modal (Visible/Invisible) để đảm bảo trải nghiệm người dùng không bị treo (freeze).
- **Test report.test.js**
  - Mục tiêu: Kiểm tra chức năng xem và trích xuất báo cáo dành cho Admin/Manager.
  - Kiểm thử các tác vụ:
    - + Đăng nhập Admin: Sử dụng tài khoản quản lý ([admin@timekeeping.com](mailto:admin@timekeeping.com)).
    - + Bộ lọc dữ liệu: Truy cập trang Báo cáo -> Nhập tháng cần xem (2025-01) -> Nhấn nạp dữ liệu.
    - + Kiểm tra chi tiết (Detailed View): Mở xem chi tiết ca làm việc của nhân viên.
    - + Kiểm tra thống kê (Aggregated Stats):
    - + Chức năng Xuất (Export): Kiểm tra nút "Xuất báo cáo" có hiển thị và cho phép click.
  - Điểm kỹ thuật: Sử dụng Wait thông minh để chờ dữ liệu từ API tải xong trước khi thực hiện các phép Assert trên các ô thống kê.
- c. **Vai trò của Selenium trong dự án**
  - Xác thực toàn bộ trải nghiệm UI của người dùng.
  - Đảm bảo giao diện không gặp lỗi hiển thị hay lỗi thao tác.
  - Kiểm tra sự tương thích giao diện trên nhiều kích thước màn hình.
  - Giảm thời gian kiểm thử thủ công trong các tác vụ lặp lại.

### 3. Postman – Công cụ kiểm thử API

- Postman là một công cụ mạnh mẽ và phổ biến dùng để kiểm thử API trong quá trình phát triển phần mềm. Công cụ cho phép gửi request trực tiếp đến server, kiểm tra dữ liệu trả về và mô phỏng toàn bộ quy trình hoạt động của các API mà không cần xây dựng giao diện người dùng.
- Trong dự án Mini App Chấm Công Nhân Viên, Postman được sử dụng để kiểm thử các API quan trọng như **Check-in, Check-out**. Những API này đóng vai trò nền tảng trong việc ghi nhận và xử lý dữ liệu chấm công, vì vậy việc kiểm thử bằng Postman giúp đảm bảo tính ổn định và độ chính xác của backend.

#### a. Lý do lựa chọn Postman

- Giao diện thân thiện, dễ sử dụng cho cả tester và developer.
- Hỗ trợ nhiều loại request (GET, POST, PUT, DELETE,...)
- Có thể viết test script (JavaScript) để kiểm tra tự động response.
- Hỗ trợ chạy Collection theo bộ giúp tự động hóa và tăng tốc kiểm thử.
- Cho phép mô phỏng nhiều môi trường (local, staging, production).

#### b. Postman trong dự án

- Module Xác Thực (Authentication)
  - **API:** POST /api/auth/login
  - Đánh giá: API xử lý tốt việc cấp phát Token JWT và bảo mật thông tin (password trả về dạng hash hoặc ẩn đi - trong log thấy passwordHash được trả về, lưu ý bảo mật nên ẩn field này ở response nếu không cần thiết)
- Module Đăng Ký (Enrollment)
  - **API:** POST /api/face/enroll
  - Đánh giá: API yêu cầu xác thực (Authorization: Bearer Token) trước khi cho phép đăng ký, đảm bảo tính bảo mật.
- Module Chấm Công (Check-in / Check-out)
  - **API:** POST /api/face/check-in & POST /api/face/check-out
  - Đánh giá: Đây là phần logic nghiệp vụ phức tạp nhất, được kiểm thử rất kỹ với nhiều trường hợp ngoại lệ.

#### c. Vai trò của Postman trong dự án

- Phát hiện lỗi nghiệp vụ trước khi đưa lên giao diện.
- Đảm bảo các API hoạt động ổn định và chính xác.

- Kiểm tra tính an toàn của API thông qua token và header.
- Đảm bảo dữ liệu chấm công được ghi nhận đúng và nhất quán.

### III. THỰC HIỆN KIỂM THỬ

#### 1. Kế hoạch kiểm thử

##### a. Mục tiêu kiểm thử

Kế hoạch kiểm thử được xây dựng nhằm đảm bảo rằng toàn bộ các chức năng của hệ thống Mini App Chấm Công Nhân Viên hoạt động chính xác theo yêu cầu đề ra. Cụ thể, mục tiêu của quá trình kiểm thử là:

- Xác minh tính đúng đắn của API chấm công (check-in, check-out, báo cáo).
- Đảm bảo giao diện người dùng UI hoạt động ổn định và không phát sinh lỗi trong quá trình thao tác.
- Kiểm thử các logic xử lý cốt lõi của hệ thống như kiểm tra điều kiện check-in/chế độ GPS và tính toán thời gian làm việc.
- Phát hiện lỗi sớm, giảm rủi ro khi triển khai và đảm bảo chất lượng hệ thống trước khi đưa vào sử dụng thực tế.

##### b. Phạm vi kiểm thử

###### • Chức năng được kiểm thử

- API Backend: Login, Check-in, Check-out.
- Giao diện Check-in/Check-out: nút chức năng, camera quét mặt, thông báo.
- Giao diện báo cáo: bảng dữ liệu, lọc theo tháng, chi tiết ngày công nhân viên.
- Logic xử lý: điều kiện chấm công, tính giờ làm, xác thực GPS.
- Chức năng phân quyền nâng cao (nếu chưa phát triển), quản lý ca làm.

##### c. Phương pháp kiểm thử

###### • Kiểm thử đơn vị (Unit Test) – Jest

- Kiểm tra logic check-in/check-out
- Kiểm tra xử lý tính toán giờ làm.
- Kiểm tra tính hợp lệ GPS.
- Kiểm tra luồng phân quyền, xác thực
- Kiểm tra quản lý ca làm.

- **Kiểm thử API – Postman**

- Gửi request và đánh giá response Module Check-in/Check-out, Login.
- Kiểm tra token và xác thực.
- Kiểm tra error handling và status code.

- **Kiểm thử giao diện – Selenium**

- Mô phỏng thao tác người dùng thật.
- Kiểm tra hiển thị bảng báo cáo, chi tiết chấm công.
- Kiểm tra popup camera và các hành động click.

**d. Môi trường kiểm thử**

- **Phần mềm**

- Trình duyệt: Microsoft Edge 143v
- Node.js v24
- Postman v11
- Selenium WebDriver
- Jest v30

- **Phần cứng**

- Máy tính hỗ trợ camera.
- GPS mô phỏng qua trình duyệt hoặc thiết bị di động thực.

- **Dữ liệu kiểm thử**

- Tài khoản nhân viên mẫu, quản lý mẫu.
- Dữ liệu GPS hợp lệ / không hợp lệ.
- Các trường hợp chấm công hợp lệ và bất thường.

**e. Lịch trình kiểm thử**

Giai đoạn	Nội dung	Thời gian
Chuẩn bị	Cấu hình môi trường, tạo dữ liệu kiểm thử	1 ngày
Kiểm thử API	Test Login, Check-in, Check-out	1 ngày
Kiểm thử UI	Test giao diện chấm công + báo cáo	2 ngày
Unit test với Jest	Kiểm thử logic xử lý	2 ngày
Tổng hợp kết quả	Ghi lỗi, báo cáo, retest	1 ngày

#### f. Tiêu chí bắt đầu và kết thúc kiểm thử

##### • Tiêu chí bắt đầu

- Môi trường đã được cấu hình hoàn chỉnh.
- Backend và frontend chạy ổn định.
- Tất cả test case đã được chuẩn bị.

##### • Tiêu chí kết thúc

- Tất cả test case đã được thực thi 100%
- Tỷ lệ pass  $\geq 90\%$ .
- Không còn lỗi nghiêm trọng (critical/blocker).
- Các lỗi mức trung bình đã được ghi lại và xác nhận.

## 2. Test Case

### a. Kiểm thử logic – Jest

ID	Chức năng	Kịch bản kiểm thử (Test Scenario)	Input / Điều kiện	Kết quả mong đợi (Expected)
<b>Test Case phân hệ Xác thực &amp; Phân quyền:</b> <code>auth.middleware.test.js</code> , <code>admin.middleware.test.js</code> và <code>selfOrAdmin.middleware.test.js</code>				
AUTH _01	JWT Auth	Token hợp lệ trong Header	Header: Bearer valid_token	next() được gọi, req.user có dữ liệu
AUTH _02	JWT Auth	Token hợp lệ trong Query Param	Query: ?token=valid_token	next() được gọi

AUTH_03	JWT Auth	Token hết hạn hoặc sai chữ ký	Token hết hạn / sai secret key	Trả về 401 Invalid token
AUTH_04	JWT Auth	Không có token	Header/Query trống	Trả về 401 Missing token
RBAC_01	Admin Role	Người dùng có quyền Admin truy cập	Role: admin, System Admin, manager	next() được gọi
RBAC_02	Admin Role	Người dùng thường truy cập trang Admin	Role: user, employee	Trả về 403 Admin only
SOA_01	Self/Admin	User xem dữ liệu của chính mình	req.user.id trùng req.params.userId	next() được gọi
SOA_02	Self/Admin	Admin xem dữ liệu người khác	User role: admin, Params: ID người khác	next() được gọi
SOA_03	Self/Admin	User thường xem dữ liệu người khác	User role: employee, Params: ID người khác	Trả về 403 Access denied
<b>Test Case phân hệ Chấm công: <span style="color: red;">attendance.controllers.test.js</span> và <span style="color: red;">attendance.service.test.js</span></b>				
ATT_01	Check-in	Check-in lần đầu trong ngày	Chưa có bản ghi Check-in hôm nay	Success: True, tạo docId mới
ATT_02	Check-in	Check-in lại (Duplicate)	Đã tồn tại bản ghi Check-in hôm nay	Success: False, báo lỗi đã check-in
ATT_03	Check-out	Check-out thành công	Đã Check-in, chưa Check-out	Success: True, cập nhật checkOutAt
ATT_04	Check-out	Check-out khi chưa Check-in	Không tìm thấy bản ghi Check-in	Success: False, báo lỗi chưa check-in
ATT_05	History	Lấy lịch sử chấm công	User ID hợp lệ	Trả về danh sách, sắp xếp theo ngày giảm dần

ATT_06	Summary	Tính tổng ngày công	Có dữ liệu shifts và attendance	Trả về đúng số daysWorked và daysOff
ATT_07	Calendar	Lấy lịch làm việc tháng	Month: YYYY-MM	Trả về object map theo ngày với status (checked-full, absent, v.v.)
ROUT_E_01	API Route	Gọi API sai phương thức (Method)	Gọi POST vào endpoint chỉ nhận GET	Trả về 404
ROUT_E_02	Admin API	Admin cập nhật chấm công	Body chứa workSeconds, note	Gọi Controller update thành công (200)
<b>Test Case phân hệ Quản lý Ca làm: <span style="color: red;">shifts.service.test.js</span></b>				
SFT_01	Tạo ca	Tạo ca với đầy đủ thông tin	Date, Name, Start, End hợp lệ	Tạo thành công, trả về Shift Code (VD: CA001)
SFT_02	Tạo ca	Tạo ca thiếu giờ	Thiếu startTime hoặc endTime	Ném lỗi (Throw Error)
SFT_03	Tạo ca	Tự động sinh tên ca	Không gửi name	Tên tự động: Ca ngày... (Giờ-Giờ)
SFT_04	Gán ca	Gán nhiều NV vào nhiều ca	userIds [], shiftIds []	Tạo đủ số lượng bản ghi trong user_shifts
SFT_05	Gán ca	Format ngày tháng	Input 28/11/2025	Lưu vào DB dạng chuẩn 2025-11-28
SFT_06	Xoá NV	Xoá nhân viên khỏi ca	ShiftId và UserId hợp lệ	Xoá bản ghi tương ứng trong DB
SFT_07	Xoá NV	Xoá nhân viên không có trong ca	ShiftId/UserId không khớp	Ném lỗi Nhân viên không tồn tại trong ca
<b>Test Case phân hệ Nhận diện Khuôn mặt &amp; GPS: <span style="color: red;">face.service.test.js</span> và <span style="color: red;">face.controller.test.js</span></b>				
FACE_01	Enroll	Đăng ký khuôn mặt mới	Embedding (vector 128 số) hợp lệ	Lưu thành công vào Firestore

FACE _02	Validate	Xác thực khuôn mặt + GPS hợp lệ	Face Distance < 0.6 VÀ GPS Distance < Radius	Success: True
FACE _03	Validate	Khuôn mặt không khớp	Face Distance > Threshold (0.6)	Success: False, message "Khuôn mặt không khớp"
FACE _04	Validate	Sai vị trí địa lý	GPS Distance > Radius (Company Settings)	Success: False, message "Ngoài khu vực làm việc"
FACE _05	Validate	Môi trường biến thiên (Env Var)	Chỉnh process.env.FACE_THRE SHOLD thấp xuống	Logic so sánh phải dùng giá trị mới từ Env
CTR_ 01	Check-in API	Check-in bằng FaceID thành công	Service Face trả về True + Service Checkin trả về True	Trả về 200 + Data checkin
CTR_ 02	Check-in API	Check-in thất bại do logic Face	Service Face trả về False	Trả về 400 + Message lỗi từ Service

## b. Kiểm thử giao diện – Selenium

ID	File Nguồn	Tên Kịch Bản (Scenario)	Các Bước Thực Hiện (Test Steps)	Dữ Liệu Kiểm Thử (Test Data)	Kết Quả Mong Đợi (Expected Result)
TC- 01	<b>checkin. test.js</b>	Check-in thành công và ghi nhận lịch sử	1. Đăng nhập tài khoản nhân viên.	Email: nv2@examp le.com	- Modal FaceID bật lên.
			2. Vào tab Home, nhấn nút "Face Check-in".	Pass: 12345678	- Video stream hiển thị.
			3. Chờ Modal và Video hiển thị.		- Số lượng bản ghi trong History tăng lên 1 dòng so với trước khi check-in.
			4. Đóng Modal (giả lập nhận diện xong).		
			5. Chuyển sang tab History.		



			6. Đếm số dòng dữ liệu.		
TC-02	<b>checkin.test.js</b>	Kiểm tra hiển thị giao diện Camera Check-in	1. Đăng nhập.	N/A	- Phần tử #faceModal có thuộc tính open.
			2. Nhấn nút "Face Check-in".		- Phần tử #faceVideo hiển thị (isDisplayed = true).
			3. Kiểm tra phần tử Modal và Video.		
TC-03	<b>checkout.test.js</b>	Check-out thành công và điều hướng	1. Đăng nhập tài khoản nhân viên.	Email: nv2@example.com	- Modal Check-out hiển thị và đóng lại được.
			2. Điều hướng về Tab Home.	Pass: 12345678	- Điều hướng thành công sang trang History.
			3. Nhấn nút "Face Check-out".		- Bảng #histTable hiển thị.
			4. Chờ Modal hiển thị, sau đó đóng Modal.		- Có dữ liệu dòng (tr) trong bảng.
			5. Chờ Modal biến mất hoàn toàn.		
			6. Điều hướng sang Tab History.		
TC-04	<b>report.test.js</b>	Xem báo cáo chấm công theo tháng (Manager)	1. Đăng nhập tài khoản Quản lý (Admin).	Email: admin@...	- Nút Summary (#btnLoadSummary) chuyển trạng thái enabled.
			2. Vào tab "Statistics & Report".	Pass: admin123	- Bảng chi tiết nhân viên (#employeeSummaryTable) có dữ liệu (innerHTML không rỗng).
			3. Nhập tháng vào ô chọn tháng.	Tháng: 2025-01	
			4. Nhấn nút "Nạp dữ liệu" (Reload).		
			5. Nhấn nút "Xem chi tiết" (Summary).		
TC-05	<b>report.test.js</b>	Kiểm tra hiển thị các	1. Thực hiện các bước nạp dữ liệu như TC-04.	N/A	Các ô chỉ số sau không được rỗng:

		chỉ số thống kê tổng hợp	2. Đọc giá trị các ô thống kê: Tổng ca, Có mặt, Vắng mặt, Tỷ lệ.		- Tổng ca (#sumTotal)
					- Có mặt (#sumPresent)
					- Vắng mặt (#sumAbsent)
					- Tỷ lệ (#sumRate)
TC-06	report.test.js	Kiểm tra trạng thái nút Xuất báo cáo	1. Thực hiện nạp dữ liệu thành công.	N/A	- Nút xuất báo cáo phải hiển thị (isDisplayed).
			2. Kiểm tra trạng thái nút Xuất báo cáo (nút Summary).		- Nút phải cho phép bấm (isEnabled).

### c. Kiểm thử API – Postman

TC ID	Test Scenario	Test Data	Header & Body	Output
LOG 01	Login thành công	<b>HTTP method:</b> POST <b>Path:</b> <a href="http://localhost:5000/api/auth/login">http://localhost:5000/api/auth/login</a>	<b>Content-Type:</b> application/json <b>Body:</b> { "email": "nv1@example.com", "password": "123" }	{ "success": true, "data": { "token": "eyJhbGciOiJIUz...", "refreshToken": "eyJhbGciOiJIUz...", "user": { "id": "oHnlk8LIMMdSEHgxDoP", "employeeCode": "NV013", "name": "Nhân viên 3", "email": "nv3@example.com", "passwordHash": "\$2b\$10\$BqW9ygcwFG/fV7vQ koWTcuIvd3HCcIKZw7JVoW0 q/OT1sZB/mImGu", "role": "user", "position": "employee", "dept": "", "status": "active",

				<pre> "workStatus": "dang_lam", "createdAt": "2025-12-04T19:40:58.403Z", "updatedAt": "2025-12-04T19:40:58.403Z" } } } </pre>
LOG 02	Sai mật khẩu		<b>Content-Type:</b> application/json <b>Body:</b> <pre> {   "embedding": [     0.121, 0.234, 0.556, -0.223,     0.112, 0.998, -0.445, 0.002   ] } </pre>	<pre> {   "success": false,   "message": "Email hoặc mật khẩu không đúng" } </pre>
ER 01	Enroll thành công	<b>HTTP method:</b> POST <b>Path:</b> <a href="http://localhost:5000/api/face/enroll">http://localhost:5000/api/face/enroll</a>	<b>Authorization:</b> Bearer eyJhbGciOiJI...	<pre> {   "success": true,   "message": "Đăng ký FaceID thành công" } </pre>
CI 01	Check-in thành công	<b>HTTP method:</b> POST <b>Path:</b> <a href="http://localhost:5000/api/face/check-in">http://localhost:5000/api/face/check-in</a>	<b>Authorization:</b> Bearer eyJhbGciOiJI...	<pre> {   "success": true,   "message": "Check-in FaceID thành công",   "data": {     "docId": "oHnlk8LIMMdSEHgxeDoP_2025-12-07",     "faceDist": 0,     "gpsDist": 0   } } </pre>
CI 02	Chưa enroll mặt		<b>Authorization:</b> Bearer eyJhbGciOiJI...	<pre> {   "success": false,   "message": "Chưa đăng ký khuôn mặt" } </pre>

			"lat": 10.123456, "lng": 106.123456 }	
CI 03	Sai mặt		<b>Authorization:</b> Bearer eyJhbGciOiJI... <b>Content-Type:</b> application/json <b>Body:</b> { "embedding": [ -0.881, 0.554, 0.221, 0.992, - 0.337, 0.998, -0.445, 0.002 ], "lat": 10.123456, "lng": 106.123456 }	{ "success": false, "message": "Khuôn mặt không khớp" }
CI 04	Thiếu GPS		<b>Authorization:</b> Bearer eyJhbGciOiJI... <b>Content-Type:</b> application/json <b>Body:</b> { "embedding": [ 0.121, 0.234, 0.556, -0.223, 0.112, 0.998, -0.445, 0.002 ] }	{ "success": false, "message": "Thiếu tọa độ GPS" }
CI 05	Ngoài vùng GPS		<b>Authorization:</b> Bearer eyJhbGciOiJI... <b>Content-Type:</b> application/json <b>Body:</b> { "embedding": [ 0.121, 0.234, 0.556, -0.223, 0.112, 0.998, -0.445, 0.002 ], "lat": 11.000000, "lng": 107.000000 }	{ "success": false, "message": "Bạn đang ngoài khu vực làm việc (cách 136676m, giới hạn 100m)." }
CI 06	Check-in lần 2		<b>Authorization:</b> Bearer eyJhbGciOiJI... <b>Content-Type:</b> application/json <b>Body:</b> { "embedding": [ 0.121, 0.234, 0.556, -0.223, 0.112, 0.998, -0.445, 0.002 ] }	{ "success": false, "message": "Bạn đã check-in rồi." }

			<pre>], "lat": 10.123456, "lng": 106.123456 }</pre>	
CO 01	Check-out thành công	<b>HTTP method:</b> POST <b>Path:</b> <a href="http://localhost:5000/api/face/check-out">http://localhost:5000/api/face/check-out</a>	<b>Authorization:</b> Bearer eyJhbGciOiJI... <b>Content-Type:</b> application/json <b>Body:</b> <pre>{   "embedding": [     0.121, 0.234, 0.556, -0.223,     0.112, 0.998, -0.445, 0.002   ], "lat": 10.123456,   "lng": 106.123456 }</pre>	<pre>{   "success": true,   "message": "Check-out FaceID thành công",   "data": {     "docId": "oHnIk8LIMMdSEHgxeDoP_2025-12-07",     "faceDist": 0,     "gpsDist": 0   } }</pre>
CO 02	Sai mặt		<b>Authorization:</b> Bearer eyJhbGciOiJI... <b>Content-Type:</b> application/json <b>Body:</b> <pre>{   "embedding": [     -0.991, 0.442, 0.330, -0.828   ], "lat": 10.123456,   "lng": 106.123456 }</pre>	<pre>{   "success": false,   "message": "Khuôn mặt không khớp" }</pre>
CO 03	Thiếu GPS		<b>Authorization:</b> Bearer eyJhbGciOiJI... <b>Content-Type:</b> application/json <b>Body:</b> <pre>{   "embedding": [     0.121, 0.234, 0.556, -0.223,     0.112, 0.998, -0.445, 0.002 ] }</pre>	<pre>{   "success": false,   "message": "Thiếu tọa độ GPS" }</pre>
CO 04	Ngoài vùng GPS		<b>Authorization:</b> Bearer eyJhbGciOiJI... <b>Content-Type:</b> application/json <b>Body:</b> <pre>{   "embedding": [     0.121, 0.234, 0.556, -0.223,     0.112, 0.998, -0.445, 0.002   ], "lat": 11.000000,   "lng": 107.000000 }</pre>	<pre>{   "success": false,   "message": "Bạn đang ngoài khu vực làm việc (cách 136676m, giới hạn 100m)." }</pre>

CO 05	Check-out khi chưa check-in		<b>Authorization:</b> Bearer eyJhbGciOiJI... <b>Content-Type:</b> application/json <b>Body:</b> <pre>{   "embedding": [     0.121, 0.234, 0.556, -0.223,     0.112, 0.998, -0.445, 0.002 ],   "lat": 10.123456,   "lng": 106.123456 }</pre>	<pre>{   "success": false,   "message": "Bạn chưa check- in hôm nay)." }</pre>
CO 06	Check-out lần 2		<b>Authorization:</b> Bearer eyJhbGciOiJI... <b>Content-Type:</b> application/json <b>Body:</b> <pre>{   "embedding": [     0.121, 0.234, 0.556, -0.223,     0.112, 0.998, -0.445, 0.002   ], "lat": 10.123456,   "lng": 106.123456 }</pre>	<pre>{   "success": false,   "message": "Bạn đã check-out rồi." }</pre>

### 3. Kết quả kiểm thử

#### a. Postman

- LOG-01 (Success): Đăng nhập đúng thông tin -> Trả về HTTP 200 kèm token, refreshToken và thông tin user đầy đủ
- LOG-02 (Fail): Đăng nhập sai password -> Trả về HTTP 400 và thông báo lỗi chính xác.
- ER-01: (Success): Gửi vector đặc trưng khuôn mặt (embedding array) kèm Token hợp lệ -> Trả về HTTP 200 "Đăng ký FaceID thành công".
- **Happy Path (Luồng thành công):**
  - CI-01, CO-01: Input hợp lệ (Face Embedding khớp, GPS trong vùng cho phép) -> Trả về HTTP 200, ghi nhận checkInAt/checkOutAt, trả về docId
- **Validation & Error Handling (Luồng lỗi):**
  - Chưa đăng ký (Unenrolled): CI-02 -> Chặn người dùng chưa có dữ liệu khuôn mặt (HTTP 400).
  - Nhận diện sai (Face Mismatch): CI-03, CO-03 -> Embedding gửi lên không khớp với Database -> Trả về lỗi "Khuôn mặt không khớp" (HTTP 400).
  - Ràng buộc GPS:

CI-05, CO-05 (Ngoài vùng): Tọa độ gửi lên nằm ngoài bán kính cho phép -> Trả về lỗi (HTTP 400).

+ CI-06: Check-in lần 2 trong ngày -> Báo lỗi "Đã check-in hôm nay".

+ CO-07: Check-out lần 2 -> Báo lỗi "Đã check-out rồi"

[illegible]

23

HTTP Login / Sai mật khẩu

POST http://localhost:5000/api/auth/login

Body

```
1 {
2   "email": "nv1@example.com",
3   "password": "123"
4 }
```

400 Bad Request • 211 ms • 367 B

Body

```
1 {
2   "success": false,
3   "message": "Email hoặc mật khẩu không đúng"
4 }
```

## LOG 02 – Sai mật khẩu

HTTP Error / Enroll thành công

POST http://localhost:5000/api/face/enroll

Body

```
1 {
2   "embedding": [
3     0.121, 0.234, 0.556, -0.223, 0.112, 0.998, -0.445, 0.002
4   ]
5 }
```

200 OK • 113 ms • 348 B

Body

```
1 {
2   "success": true,
3   "message": "Đăng ký FaceID thành công"
4 }
```

## ER 01 – Enroll thành công



HTTP Check-in / **Check-in thành công** Save Share

**POST** http://localhost:5000/api/face/check-in Send

Overview Params Authorization Headers (10) **Body** Scripts Settings Cookies

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL **JSON** Schema Beautify

```
1 {
2   "embedding": [
3     0.121, 0.234, 0.556, -0.223, 0.112, 0.998, -0.445, 0.002
4   ],
5   "lat": 10.123456,
6   "lng": 106.123456
7 }
```

**Body** Cookies Headers (9) Test Results 200 OK • 462 ms • 423 B • Save Response

**{}** JSON Preview Visualize

```
1 {
2   "success": true,
3   "message": "Check-in FaceID thành công",
4   "data": {
5     "docId": "aIUwc6CU3hV0W233kDQ_2025-12-07",
6     "faceDist": 0,
7     "gpsDist": 0
8   }
9 }
```

### CI 01 – Check-in thành công

HTTP Check-in / **Chưa enroll khuôn mặt** Save Share

**POST** http://localhost:5000/api/face/check-in Send

Overview Params Authorization Headers (10) **Body** Scripts Settings Cookies

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL **JSON** Schema Beautify

```
1 {
2   "embedding": [
3     0.121, 0.234, 0.556, -0.223, 0.112, 0.998, -0.445, 0.002
4   ],
5   "lat": 10.123456,
6   "lng": 106.123456
7 }
```

**Body** Cookies Headers (9) Test Results 400 Bad Request • 97 ms • 357 B • Save Response

**{}** JSON Preview Debug with AI

```
1 {
2   "success": false,
3   "message": "Chưa đăng ký khuôn mặt"
4 }
```

### CI 02 – Chưa enroll mặt

HTTP Check-in / **Khuôn mặt không khớp** Save Share

**POST** ▼ http://localhost:5000/api/face/check-in Send ▼

Docs Params Authorization Headers (10) Body Scripts Settings Cookies

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL **JSON** ▼ Schema Beautify

```
1 {
2   "embedding": [
3     -0.881, 0.554, 0.221, 0.992, -0.337
4   ],
5   "lat": 10.123456,
6   "lng": 106.123456
7 }
```

**Body** Cookies Headers (9) Test Results ↺ 400 Bad Request • 83 ms • 354 B • Save Response ⋮

{} **JSON** ▼ Preview Debug with AI ▼ ⌵ ⌵ 🔍 📄 🔗

```
1 {
2   "success": false,
3   "message": "Khuôn mặt không khớp"
4 }
```

### CI 03 – Khuôn mặt không khớp

HTTP Check-in / **Thiếu GPS** Save Share

**POST** ▼ http://localhost:5000/api/face/check-in Send ▼

Overview Params Authorization Headers (10) Body Scripts Settings Cookies

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL **JSON** ▼ Schema Beautify

```
1 {
2   "embedding": [
3     0.121, 0.234, 0.556, -0.223, 0.112, 0.998, -0.445, 0.002
4   ]
5 }
```

**Body** Cookies Headers (9) Test Results 400 Bad Request • 102 ms • 351 B • Save Response ⋮

{} **JSON** ▼ Preview Debug with AI ▼ ⌵ ⌵ 🔍 📄 🔗

```
1 {
2   "success": false,
3   "message": "Thiếu tọa độ GPS"
4 }
```

### CI 04 – Thiếu GPS

HTTP Check-in / **Ngoài vùng GPS** Save Share

**POST** ▼ http://localhost:5000/api/face/check-in Send

Overview Params Authorization Headers (10) **Body** • Scripts Settings Cookies

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL **JSON** ▼ Schema Beautify

```

1 {
2   "embedding": [
3     0.121, 0.234, 0.556, -0.223, 0.112, 0.998, -0.445, 0.002
4   ],
5   "lat": 11.000000,
6   "lng": 107.000000
7 }

```

Body Cookies Headers (9) Test Results 400 Bad Request • 178 ms • 405 B • Save Response

**{}** JSON ▼ Preview Debug with AI ▼ Schema Beautify

```

1 {
2   "success": false,
3   "message": "Bạn đang ngoài khu vực làm việc (cách 136676m, giới hạn 100m)."
4 }

```

### CI 05 – Ngoài vùng GPS

HTTP Check-in / **Check-in lần 2** Save Share

**POST** ▼ http://localhost:5000/api/face/check-in Send

Docs Params Authorization Headers (10) **Body** • Scripts Settings Cookies

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL **JSON** ▼ Schema Beautify

```

1 {
2   "embedding": [
3     0.121, 0.234, 0.556, -0.223, 0.112, 0.998, -0.445, 0.002
4   ],
5   "lat": 10.123456,
6   "lng": 106.123456
7 }

```

Body Cookies Headers (9) Test Results 400 Bad Request • 568 ms • 363 B • Save Response

**{}** JSON ▼ Preview Debug with AI ▼ Schema Beautify

```

1 {
2   "success": false,
3   "message": "Bạn đã check-in hôm nay rồi."
4 }

```

### CI 06 – Check-in lần 2

HTTP Check-out / **Check-out thành công** Save Share

**POST** http://localhost:5000/api/face/check-out Send

Docs Params Authorization Headers (10) **Body** Scripts Settings Cookies

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL **JSON** Schema Beautify

```
1 {
2   "embedding": [
3     0.121, 0.234, 0.556, -0.223, 0.112, 0.998, -0.445, 0.002
4   ],
5   "lat": 10.123456,
6   "lng": 106.123456
7 }
```

Body Cookies Headers (9) Test Results 200 OK • 375 ms • 424 B Save Response

**{}** JSON Preview Visualize

```
1 {
2   "success": true,
3   "message": "Check-out FaceID thành công",
4   "data": {
5     "docId": "FDFA4ZaPQgiWNdaQBypA_2025-12-07",
6     "faceDist": 0,
7     "gpsDist": 0
8   }
9 }
```

### CO 01 – Check-out thành công

HTTP Check-out / **Khuôn mặt không khớp** Save Share

**POST** http://localhost:5000/api/face/check-out Send

Docs Params Authorization Headers (10) **Body** Scripts Settings Cookies

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL **JSON** Schema Beautify

```
1 {
2   "embedding": [
3     -0.991, 0.442, 0.330, -0.828
4   ],
5   "lat": 10.123456,
6   "lng": 106.123456
7 }
```

Body Cookies Headers (9) Test Results 400 Bad Request • 80 ms • 354 B Save Response

**{}** JSON Preview Debug with AI

```
1 {
2   "success": false,
3   "message": "Khuôn mặt không khớp"
4 }
```

### CO 02 – Khuôn mặt không khớp

HTTP Check-out / **Thiếu GPS** Save Share

**POST** http://localhost:5000/api/face/check-out Send

Overview Params Authorization Headers (10) **Body** Scripts Settings Cookies

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL **JSON** Schema Beautify

```

1 {
2   "embedding": [
3     0.121, 0.234, 0.556, -0.223, 0.112, 0.998, -0.445, 0.002
4   ]
5 }

```

**Body** Cookies Headers (9) Test Results 400 Bad Request • 115 ms • 351 B • Save Response ...

**{}** **JSON** Preview Debug with AI ...

```

1 {
2   "success": false,
3   "message": "Thiếu tọa độ GPS"
4 }

```

### CO 03 – Thiếu GPS

HTTP Check-out / **Ngoài vùng GPS** Save Share

**POST** http://localhost:5000/api/face/check-out Send

Overview Params Authorization Headers (10) **Body** Scripts Settings Cookies

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL **JSON** Schema Beautify

```

1 {
2   "embedding": [
3     0.121, 0.234, 0.556, -0.223, 0.112, 0.998, -0.445, 0.002
4   ],
5   "lat": 11.000000,
6   "lng": 107.000000
7 }

```

**Body** Cookies Headers (9) Test Results 400 Bad Request • 148 ms • 405 B • Save Response ...

**{}** **JSON** Preview Debug with AI ...

```

1 {
2   "success": false,
3   "message": "Bạn đang ngoài khu vực làm việc (cách 136676m, giới hạn 100m)."
4 }

```

### CO 04 – Ngoài vùng GPS

HTTP Check-out / **Check-out lần 2** Save Share

**POST** ▼ http://localhost:5000/api/face/check-out Send ▼

Docs Params Authorization Headers (10) **Body** • Scripts Settings Cookies

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL **JSON** ▼ Schema Beautify

```

1  {
2    "embedding": [
3      0.121, 0.234, 0.556, -0.223, 0.112, 0.998, -0.445, 0.002
4    ],
5    "lat": 10.123456,
6    "lng": 106.123456
7  }

```

Body Cookies Headers (9) Test Results | 400 Bad Request • 264 ms • 355 B • Save Response ...

**{}** **JSON** ▼ Preview Debug with AI ▼ Schema Beautify

```

1  {
2    "success": false,
3    "message": "Bạn đã check-out rồi."
4  }

```

### CO 05 – Check-out lần 2

HTTP Check-out / **Check-out khi chưa Check-in** Save Share

**POST** ▼ http://localhost:5000/api/face/check-out Send ▼

Overview Params Authorization Headers (10) **Body** • Scripts Settings Cookies

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL **JSON** ▼ Schema Beautify

```

1  {
2    "embedding": [
3      0.121, 0.234, 0.556, -0.223, 0.112, 0.998, -0.445, 0.002
4    ],
5    "lat": 10.123456,
6    "lng": 106.123456
7  }

```

Body Cookies Headers (9) Test Results 400 Bad Request • 251 ms • 358 B • Save Response ...

**{}** **JSON** ▼ Preview Debug with AI ▼ Schema Beautify

```

1  {
2    "success": false,
3    "message": "Bạn chưa check-in hôm nay."
4  }

```

### CO 06 – Check-out khi chưa Check-in

## b. Jest

- **attendance.service.test.js**

- **Chức năng:** Kiểm thử logic nghiệp vụ chấm công (Check-in/Check-out) và thống kê.
- **Đối tượng test:** handleCheckInService, handleCheckOutService, fetchSummaryService, v.v.
- **Phân tích kịch bản:**
  - + Trạng thái chấm công: Ngăn chặn check-in 2 lần trong ngày, hoặc check-out khi chưa check-in.
  - + Tính toán: Logic tính daysWorked (ngày làm) và daysOff (ngày nghỉ) dựa trên dữ liệu giả lập từ Firebase
  - + Lịch làm việc: Logic phức tạp để tổng hợp dữ liệu từ 4 nguồn (Ca làm, Chấm công, Yêu cầu, Nghỉ phép) để tạo ra trạng thái hiển thị trên lịch (ví dụ: checked-full, absent, pending-request).
- **Kết quả:**

```
backend > _tests_ > JS attendance.service.test.js > describe("Attendance Service Logic") callback > describe("3. History & Summary Logic") c
18 describe('Attendance Service Logic', () => {
195 describe('3. History & Summary Logic', () => {
198 it('fetchSummaryService: Should count daysWorked/daysOff correctly', async () => {
222 });
223
224 it('fetchHistoryService: Should calculate workMinutes from workSeconds', async () => {
225 const mockDocs = [{
226   data: () => ({
227     date: '2024-03-15',
228     checkInAt: '...',
229     checkOutAt: '...',
230     workSeconds: 32400, // 9 hours
231     note: ''
232   })
233 }];
234
235 // ...
236
237 // ...
238
239 // ...
240
241 // ...
242
243 // ...
244
245 // ...
246
247 // ...
248
249 // ...
250
251 // ...
252
253 // ...
254
255 // ...
256
257 // ...
258
259 // ...
260
261 // ...
262
263 // ...
264
265 // ...
266
267 // ...
268
269 // ...
270
271 // ...
272
273 // ...
274
275 // ...
276
277 // ...
278
279 // ...
280
281 // ...
282
283 // ...
284
285 // ...
286
287 // ...
288
289 // ...
290
291 // ...
292
293 // ...
294
295 // ...
296
297 // ...
298
299 // ...
300
301 // ...
302
303 // ...
304
305 // ...
306
307 // ...
308
309 // ...
310
311 // ...
312
313 // ...
314
315 // ...
316
317 // ...
318
319 // ...
320
321 // ...
322
323 // ...
324
325 // ...
326
327 // ...
328
329 // ...
330
331 // ...
332
333 // ...
334
335 // ...
336
337 // ...
338
339 // ...
340
341 // ...
342
343 // ...
344
345 // ...
346
347 // ...
348
349 // ...
350
351 // ...
352
353 // ...
354
355 // ...
356
357 // ...
358
359 // ...
360
361 // ...
362
363 // ...
364
365 // ...
366
367 // ...
368
369 // ...
370
371 // ...
372
373 // ...
374
375 // ...
376
377 // ...
378
379 // ...
380
381 // ...
382
383 // ...
384
385 // ...
386
387 // ...
388
389 // ...
390
391 // ...
392
393 // ...
394
395 // ...
396
397 // ...
398
399 // ...
400
401 // ...
402
403 // ...
404
405 // ...
406
407 // ...
408
409 // ...
410
411 // ...
412
413 // ...
414
415 // ...
416
417 // ...
418
419 // ...
420
421 // ...
422
423 // ...
424
425 // ...
426
427 // ...
428
429 // ...
430
431 // ...
432
433 // ...
434
435 // ...
436
437 // ...
438
439 // ...
440
441 // ...
442
443 // ...
444
445 // ...
446
447 // ...
448
449 // ...
450
451 // ...
452
453 // ...
454
455 // ...
456
457 // ...
458
459 // ...
460
461 // ...
462
463 // ...
464
465 // ...
466
467 // ...
468
469 // ...
470
471 // ...
472
473 // ...
474
475 // ...
476
477 // ...
478
479 // ...
480
481 // ...
482
483 // ...
484
485 // ...
486
487 // ...
488
489 // ...
490
491 // ...
492
493 // ...
494
495 // ...
496
497 // ...
498
499 // ...
500
501 // ...
502
503 // ...
504
505 // ...
506
507 // ...
508
509 // ...
510
511 // ...
512
513 // ...
514
515 // ...
516
517 // ...
518
519 // ...
520
521 // ...
522
523 // ...
524
525 // ...
526
527 // ...
528
529 // ...
530
531 // ...
532
533 // ...
534
535 // ...
536
537 // ...
538
539 // ...
540
541 // ...
542
543 // ...
544
545 // ...
546
547 // ...
548
549 // ...
550
551 // ...
552
553 // ...
554
555 // ...
556
557 // ...
558
559 // ...
560
561 // ...
562
563 // ...
564
565 // ...
566
567 // ...
568
569 // ...
570
571 // ...
572
573 // ...
574
575 // ...
576
577 // ...
578
579 // ...
580
581 // ...
582
583 // ...
584
585 // ...
586
587 // ...
588
589 // ...
590
591 // ...
592
593 // ...
594
595 // ...
596
597 // ...
598
599 // ...
600
601 // ...
602
603 // ...
604
605 // ...
606
607 // ...
608
609 // ...
610
611 // ...
612
613 // ...
614
615 // ...
616
617 // ...
618
619 // ...
620
621 // ...
622
623 // ...
624
625 // ...
626
627 // ...
628
629 // ...
630
631 // ...
632
633 // ...
634
635 // ...
636
637 // ...
638
639 // ...
640
641 // ...
642
643 // ...
644
645 // ...
646
647 // ...
648
649 // ...
650
651 // ...
652
653 // ...
654
655 // ...
656
657 // ...
658
659 // ...
660
661 // ...
662
663 // ...
664
665 // ...
666
667 // ...
668
669 // ...
670
671 // ...
672
673 // ...
674
675 // ...
676
677 // ...
678
679 // ...
680
681 // ...
682
683 // ...
684
685 // ...
686
687 // ...
688
689 // ...
690
691 // ...
692
693 // ...
694
695 // ...
696
697 // ...
698
699 // ...
700
701 // ...
702
703 // ...
704
705 // ...
706
707 // ...
708
709 // ...
710
711 // ...
712
713 // ...
714
715 // ...
716
717 // ...
718
719 // ...
720
721 // ...
722
723 // ...
724
725 // ...
726
727 // ...
728
729 // ...
730
731 // ...
732
733 // ...
734
735 // ...
736
737 // ...
738
739 // ...
740
741 // ...
742
743 // ...
744
745 // ...
746
747 // ...
748
749 // ...
750
751 // ...
752
753 // ...
754
755 // ...
756
757 // ...
758
759 // ...
760
761 // ...
762
763 // ...
764
765 // ...
766
767 // ...
768
769 // ...
770
771 // ...
772
773 // ...
774
775 // ...
776
777 // ...
778
779 // ...
780
781 // ...
782
783 // ...
784
785 // ...
786
787 // ...
788
789 // ...
790
791 // ...
792
793 // ...
794
795 // ...
796
797 // ...
798
799 // ...
800
801 // ...
802
803 // ...
804
805 // ...
806
807 // ...
808
809 // ...
810
811 // ...
812
813 // ...
814
815 // ...
816
817 // ...
818
819 // ...
820
821 // ...
822
823 // ...
824
825 // ...
826
827 // ...
828
829 // ...
830
831 // ...
832
833 // ...
834
835 // ...
836
837 // ...
838
839 // ...
840
841 // ...
842
843 // ...
844
845 // ...
846
847 // ...
848
849 // ...
850
851 // ...
852
853 // ...
854
855 // ...
856
857 // ...
858
859 // ...
860
861 // ...
862
863 // ...
864
865 // ...
866
867 // ...
868
869 // ...
870
871 // ...
872
873 // ...
874
875 // ...
876
877 // ...
878
879 // ...
880
881 // ...
882
883 // ...
884
885 // ...
886
887 // ...
888
889 // ...
890
891 // ...
892
893 // ...
894
895 // ...
896
897 // ...
898
899 // ...
900
901 // ...
902
903 // ...
904
905 // ...
906
907 // ...
908
909 // ...
910
911 // ...
912
913 // ...
914
915 // ...
916
917 // ...
918
919 // ...
920
921 // ...
922
923 // ...
924
925 // ...
926
927 // ...
928
929 // ...
930
931 // ...
932
933 // ...
934
935 // ...
936
937 // ...
938
939 // ...
940
941 // ...
942
943 // ...
944
945 // ...
946
947 // ...
948
949 // ...
950
951 // ...
952
953 // ...
954
955 // ...
956
957 // ...
958
959 // ...
960
961 // ...
962
963 // ...
964
965 // ...
966
967 // ...
968
969 // ...
970
971 // ...
972
973 // ...
974
975 // ...
976
977 // ...
978
979 // ...
980
981 // ...
982
983 // ...
984
985 // ...
986
987 // ...
988
989 // ...
990
991 // ...
992
993 // ...
994
995 // ...
996
997 // ...
998
999 // ...
1000
1001 // ...
1002
1003 // ...
1004
1005 // ...
1006
1007 // ...
1008
1009 // ...
1010
1011 // ...
1012
1013 // ...
1014
1015 // ...
1016
1017 // ...
1018
1019 // ...
1020
1021 // ...
1022
1023 // ...
1024
1025 // ...
1026
1027 // ...
1028
1029 // ...
1030
1031 // ...
1032
1033 // ...
1034
1035 // ...
1036
1037 // ...
1038
1039 // ...
1040
1041 // ...
1042
1043 // ...
1044
1045 // ...
1046
1047 // ...
1048
1049 // ...
1050
1051 // ...
1052
1053 // ...
1054
1055 // ...
1056
1057 // ...
1058
1059 // ...
1060
1061 // ...
1062
1063 // ...
1064
1065 // ...
1066
1067 // ...
1068
1069 // ...
1070
1071 // ...
1072
1073 // ...
1074
1075 // ...
1076
1077 // ...
1078
1079 // ...
1080
1081 // ...
1082
1083 // ...
1084
1085 // ...
1086
1087 // ...
1088
1089 // ...
1090
1091 // ...
1092
1093 // ...
1094
1095 // ...
1096
1097 // ...
1098
1099 // ...
1100
1101 // ...
1102
1103 // ...
1104
1105 // ...
1106
1107 // ...
1108
1109 // ...
1110
1111 // ...
1112
1113 // ...
1114
1115 // ...
1116
1117 // ...
1118
1119 // ...
1120
1121 // ...
1122
1123 // ...
1124
1125 // ...
1126
1127 // ...
1128
1129 // ...
1130
1131 // ...
1132
1133 // ...
1134
1135 // ...
1136
1137 // ...
1138
1139 // ...
1140
1141 // ...
1142
1143 // ...
1144
1145 // ...
1146
1147 // ...
1148
1149 // ...
1150
1151 // ...
1152
1153 // ...
1154
1155 // ...
1156
1157 // ...
1158
1159 // ...
1160
1161 // ...
1162
1163 // ...
1164
1165 // ...
1166
1167 // ...
1168
1169 // ...
1170
1171 // ...
1172
1173 // ...
1174
1175 // ...
1176
1177 // ...
1178
1179 // ...
1180
1181 // ...
1182
1183 // ...
1184
1185 // ...
1186
1187 // ...
1188
1189 // ...
1190
1191 // ...
1192
1193 // ...
1194
1195 // ...
1196
1197 // ...
1198
1199 // ...
1200
1201 // ...
1202
1203 // ...
1204
1205 // ...
1206
1207 // ...
1208
1209 // ...
1210
1211 // ...
1212
1213 // ...
1214
1215 // ...
1216
1217 // ...
1218
1219 // ...
1220
1221 // ...
1222
1223 // ...
1224
1225 // ...
1226
1227 // ...
1228
1229 // ...
1230
1231 // ...
1232
1233 // ...
1234
1235 // ...
1236
1237 // ...
1238
1239 // ...
1240
1241 // ...
1242
1243 // ...
1244
1245 // ...
1246
1247 // ...
1248
1249 // ...
1250
1251 // ...
1252
1253 // ...
1254
1255 // ...
1256
1257 // ...
1258
1259 // ...
1260
1261 // ...
1262
1263 // ...
1264
1265 // ...
1266
1267 // ...
1268
1269 // ...
1270
1271 // ...
1272
1273 // ...
1274
1275 // ...
1276
1277 // ...
1278
1279 // ...
1280
1281 // ...
1282
1283 // ...
1284
1285 // ...
1286
1287 // ...
1288
1289 // ...
1290
1291 // ...
1292
1293 // ...
1294
1295 // ...
1296
1297 // ...
1298
1299 // ...
1300
1301 // ...
1302
1303 // ...
1304
1305 // ...
1306
1307 // ...
1308
1309 // ...
1310
1311 // ...
1312
1313 // ...
1314
1315 // ...
1316
1317 // ...
1318
1319 // ...
1320
1321 // ...
1322
1323 // ...
1324
1325 // ...
1326
1327 // ...
1328
1329 // ...
1330
1331 // ...
1332
1333 // ...
1334
1335 // ...
1336
1337 // ...
1338
1339 // ...
1340
1341 // ...
1342
1343 // ...
1344
1345 // ...
1346
1347 // ...
1348
1349 // ...
1350
1351 // ...
1352
1353 // ...
1354
1355 // ...
1356
1357 // ...
1358
1359 // ...
1360
1361 // ...
1362
1363 // ...
1364
1365 // ...
1366
1367 // ...
1368
1369 // ...
1370
1371 // ...
1372
1373 // ...
1374
1375 // ...
1376
1377 // ...
1378
1379 // ...
1380
1381 // ...
1382
1383 // ...
1384
1385 // ...
1386
1387 // ...
1388
1389 // ...
1390
1391 // ...
1392
1393 // ...
1394
1395 // ...
1396
1397 // ...
1398
1399 // ...
1400
1401 // ...
1402
1403 // ...
1404
1405 // ...
1406
1407 // ...
1408
1409 // ...
1410
1411 // ...
1412
1413 // ...
1414
1415 // ...
1416
1417 // ...
1418
1419 // ...
1420
1421 // ...
1422
1423 // ...
1424
1425 // ...
1426
1427 // ...
1428
1429 // ...
1430
1431 // ...
1432
1433 // ...
1434
1435 // ...
1436
1437 // ...
1438
1439 // ...
1440
1441 // ...
1442
1443 // ...
1444
1445 // ...
1446
1447 // ...
1448
1449 // ...
1450
1451 // ...
1452
1453 // ...
1454
1455 // ...
1456
1457 // ...
1458
1459 // ...
1460
1461 // ...
1462
1463 // ...
1464
1465 // ...
1466
1467 // ...
1468
1469 // ...
1470
1471 // ...
1472
1473 // ...
1474
1475 // ...
1476
1477 // ...
1478
1479 // ...
1480
1481 // ...
1482
1483 // ...
1484
1485 // ...
1486
1487 // ...
1488
1489 // ...
1490
1491 // ...
1492
1493 // ...
1494
1495 // ...
1496
1497 // ...
1498
1499 // ...
1500
1501 // ...
1502
1503 // ...
1504
1505 // ...
1506
1507 // ...
1508
1509 // ...
1510
1511 // ...
1512
1513 // ...
1514
1515 // ...
1516
1517 // ...
1518
1519 // ...
1520
1521 // ...
1522
1523 // ...
1524
1525 // ...
1526
1527 // ...
1528
1529 // ...
1530
1531 // ...
1532
1533 // ...
1534
1535 // ...
1536
1537 // ...
1538
1539 // ...
1540
1541 // ...
1542
1543 // ...
1544
1545 // ...
1546
1547 // ...
1548
1549 // ...
1550
1551 // ...
1552
1553 // ...
1554
1555 // ...
1556
1557 // ...
1558
1559 // ...
1560
1561 // ...
1562
1563 // ...
1564
1565 // ...
1566
1567 // ...
1568
1569 // ...
1570
1571 // ...
1572
1573 // ...
1574
1575 // ...
1576
1577 // ...
1578
1579 // ...
1580
1581 // ...
1582
1583 // ...
1584
1585 // ...
1586
1587 // ...
1588
1589 // ...
1590
1591 // ...
1592
1593 // ...
1594
1595 // ...
1596
1597 // ...
1598
1599 // ...
1600
1601 // ...
1602
1603 // ...
1604
1605 // ...
1606
1607 // ...
1608
1609 // ...
1610
1611 // ...
1612
1613 // ...
1614
1615 // ...
1616
1617 // ...
1618
1619 // ...
1620
1621 // ...
1622
1623 // ...
1624
1625 // ...
1626
1627 // ...
1628
1629 // ...
1630
1631 // ...
1632
1633 // ...
1634
1635 // ...
1636
1637 // ...
1638
1639 // ...
1640
1641 // ...
1642
1643 // ...
1644
1645 // ...
1646
1647 // ...
1648
1649 // ...
1650
1651 // ...
1652
1653 // ...
1654
1655 // ...
1656
1657 // ...
1658
1659 // ...
1660
1661 // ...
1662
1663 // ...
1664
1665 // ...
1666
1667 // ...
1668
1669 // ...
1670
1671 // ...
1672
1673 // ...
1674
1675 // ...
1676
1677 // ...
1678
1679 // ...
1680
1681 // ...
1682
1683 // ...
1684
1685 // ...
1686
1687 // ...
1688
1689 // ...
1690
1691 // ...
1692
1693 // ...
1694
1695 // ...
1696
1697 // ...
1698
1699 // ...
1700
1701 // ...
1702
1703 // ...
1704
1705 // ...
1706
1707 // ...
1708
1709 // ...
1710
1711 // ...
1712
1713 // ...
1714
1715 // ...
1716
1717 // ...
1718
1719 // ...
1720
1721 // ...
1722
1723 // ...
1724
1725 // ...
1726
1727 // ...
1728
1729 // ...
1730
1731 // ...
1732
1733 // ...
1734
1735 // ...
1736
1737 // ...
1738
1739 // ...
1740
1741 // ...
1742
1743 // ...
1744
1745 // ...
1746
1747 // ...
1748
1749 // ...
1750
1751 // ...
1752
1753 // ...
1754
1755 // ...
1756
1757 // ...
1758
1759 // ...
1760
1761 // ...
1762
1763 // ...
1764
1765 // ...
1766
1767 // ...
1768
1769 // ...
1770
1771 // ...
1772
1773 // ...
1774
1775 // ...
1776
1777 // ...
1778
1779 // ...
1780
1781 // ...
1782
1783 // ...
1784
1785 // ...
1786
1787 // ...
1788
1789 // ...
1790
1791 // ...
1792
1793 // ...
1794
1795 // ...
1796
1797 // ...
1798
1799 // ...
1800
1801 // ...
1802
1803 // ...
1804
1805 // ...
1806
1807 // ...
1808
1809 // ...
1810
1811 // ...
1812
1813 // ...
1814
1815 // ...
1816
1817 // ...
1818
1819 // ...
1820
1821 // ...
1822
1823 // ...
1824
1825 // ...
1826
1827 // ...
1828
1829 // ...
1830
1831 // ...
1832
1833 // ...
1834
1835 // ...
1836
1837 // ...
1838
1839 // ...
1840
1841 // ...
1842
1843 // ...
1844
1845 // ...
1846
1847 // ...
1848
1849 // ...
1850
1851 // ...
1852
1853 // ...
1854
1855 // ...
1856
1857 // ...
1858
1859 // ...
1860
1861 // ...
1862
1863 // ...
1864
1865 // ...
1866
1867 // ...
1868
1869 // ...
1870
1871 // ...
1872
1873 // ...
1874
1875 // ...
1876
1877 // ...
1878
1879 // ...
1880
1881 // ...
1882
1883 // ...
1884
1885 // ...
1886
1887 // ...
1888
1889 // ...
1890
1891 // ...
1892
1893 // ...
1894
1895 // ...
1896
1897 // ...
1898
1899 // ...
1900
1901 // ...
1902
1903 // ...
1904
1905 // ...
1906
1907 // ...
1908
1909 // ...
1910
1911 // ...
1912
1913 // ...
1914
1915 // ...
1916
1917 // ...
1918
1919 // ...
1920
1921 // ...
1922
1923 // ...
1924
1925 // ...
1926
1927 // ...
1928
1929 // ...
1930
1931 // ...
1932
1933 // ...
1934
1935 // ...
1936
1937 // ...
1938
1939 // ...
1940
1941 // ...
1942
1943 // ...
1944
1945 // ...
1946
1947 // ...
1948
1949 // ...
1950
1951 // ...
1952
1953 // ...
1954
1955 // ...
1956
1957 // ...
1958
1959 // ...
1960
1961 // ...
1962
1963 // ...
1964
1965 // ...
1966
1967 // ...
1968
1969 // ...
1970
1971 // ...
1972
1973 // ...
1974
1975 // ...
1976
1977 // ...
1978
1979 // ...
1980
1981 // ...
1982
1983 // ...
1984
1985 // ...
1986
1987 // ...
1988
1989 // ...
1990
1991 // ...
1992
1993 // ...
1994
1995 // ...
1996
1997 // ...
1998
1999 // ...
2000
2001 // ...
2002
2003 // ...
2004
2005 // ...
2006
2007 // ...
2008
2009 // ...
2010
2011 // ...
2012
2013 // ...
2014
2015 // ...
2016
2017 // ...
2018
2019 // ...
2020
2021 // ...
2022
2023 // ...
2024
2025 // ...
2026
2027 // ...
2028
2029 // ...
2030
2031 // ...
2032
2033 // ...
2034
2035 // ...
2036
2037 // ...
2038
2039 // ...
2040
2041 // ...
2042
2043 // ...
2044
2045 // ...
2046
2047 // ...
2048
2049 // ...
2050
2051 // ...
2052
2053 // ...
2054
2055 // ...
2056
2057 // ...
2058
2059 // ...
2060
2061 // ...
2062
2063 // ...
2064
2065 // ...
2066
2067 // ...
2068
2069 // ...
2070
2071 // ...
2072
2073 // ...
2074
2075 // ...
2076
2077 // ...
2078
2079 // ...
2080
2081 // ...
2082
2083 // ...
2084
2085 // ...
2086
2087 // ...
2088
2089 // ...
2090
2091 // ...
2092
2093 // ...
2094
2095 // ...
2096
2097 // ...
2098
2099 // ...
2100
2101 // ...
2102
2103 // ...
2104
2105 // ...
2106
2107 // ...
2108
2109 // ...
2110
2111 // ...
2112
2113 // ...
2114
2115 // ...
2116
2117 // ...
2118
2119 // ...
2120
2121 // ...
2122
2123 // ...
2124
2125 // ...
2126
2127 // ...
2128
2129 // ...
2130
2131 // ...
2132
2133 // ...
2134
2135 // ...
2136
2137 // ...
2138
2139 // ...
2140
2141 // ...
2142
2143 // ...
2144
2145 // ...
2146
2147 // ...
2148
2149 // ...
2150
2151 // ...
2152
2153 // ...
2154
2155 // ...
2156
2157 // ...
2158
2159 // ...
2160
2161 // ...
2162
2163 // ...
2164
2165 // ...
2166
2167 // ...
2168
2169 // ...
2170
2171 // ...
2172
2173 // ...
2174
2175 // ...
2176
2177 // ...
2178
2179 // ...
2180
2181 // ...
2182
2183 // ...
2184
2185 // ...
2186
2187 // ...
2188
2189 // ...
2190
2191 // ...
2192
2193 // ...
2194
2195 // ...
2196
2197 // ...
2198
2199 // ...
2200
2201 // ...
2202
2203 // ...
2204
2205 // ...
2206
2207 // ...
2208
2209 // ...
2210
2211 // ...
2212
2213 // ...
2214
2215 // ...
2216
2217 // ...
2218
2219 // ...
2220
2221 // ...
2222
2223 // ...
2224
2225 // ...
2226
2227 // ...
2228
2229 // ...
2230
2231 // ...
2232
2233 // ...
2234
2235 // ...
2236
2237 // ...
2238
2239 // ...
2240
2241 // ...
2242
2243 // ...
2244
2245 // ...
2246
2247 // ...
2248
2249 // ...
2250
2251 // ...
2252
2253 // ...
2254
2255 // ...
2256
2257 // ...
2258
2259 // ...
2260
2261 // ...
2262
2263 // ...
2264
2265 // ...
2266
2267 // ...
2268
2269 // ...
2270
2271 // ...
2272
2273 // ...
2274
2275 // ...
2276
2277 // ...
2278
2279 // ...
2280
2281 // ...
2282
2283 // ...
2284
2285 // ...
2286
2287 // ...
2288
2289 // ...
2290
2291 // ...
2292
2293 // ...
2294
2295 // ...
2296
2297 // ...
2298
2299 // ...
2300
2301 // ...
2302
2303 // ...
2304
2305 // ...
2306
2307 // ...
2308
2309 // ...
2310
2311 // ...
2312
2313 // ...
2314
2315 // ...
2316
2317 // ...
2318
2319 // ...
2
```

- **Đối tượng test:** getHistory, getSummary, adminGetAllAttendance, adminUpdateAttendance, getUserCalendar
- **Phân tích kịch bản:**
  - + Xử lý dữ liệu: Kiểm tra việc controller nhận dữ liệu từ service và trả về JSON đúng cấu trúc.
  - + Query Params: Test logic xử lý tham số trên URL, ví dụ: bắt lỗi nếu thiếu tham số month khi gọi API lấy lịch làm việc (getUserCalendar).
  - + Quyền Admin: Test chức năng admin cập nhật chấm công (adminUpdateAttendance), đảm bảo dữ liệu sửa đổi được gửi đúng xuống service.
- **Kết quả:**

The screenshot shows the VS Code interface with the file explorer on the left and the terminal on the right. The file explorer shows the project structure with the file `attendance.controllers.test.js` selected. The terminal displays the output of running the tests, showing that all tests passed.

```

backend > _testsJ_ > JS attendance.controllers.test.js > describe('Attendance Controller') callback
14 describe('Attendance Controller'. () => {
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS POSTMAN CONSOLE
PASS _testsJ_/attendance.controllers.test.js
Attendance Controller
getHistory
  ✓ should return user attendance history successfully (8 ms)
  ✓ should return empty array when no history exists (1 ms)
  ✓ should handle service errors gracefully (37 ms)
getSummary
  ✓ should return user attendance summary successfully (2 ms)
  ✓ should return zero values when no attendance data (1 ms)
  ✓ should handle service errors gracefully (11 ms)
adminGetAllAttendance
  ✓ should return all attendance records successfully (1 ms)
  ✓ should return empty array when no records exist (1 ms)
  ✓ should handle service errors gracefully (11 ms)
  ✓ should return 400 if from or to params are missing (2 ms)
adminGetOneAttendance
  ✓ should return attendance record when it exists (2 ms)
  ✓ should return 404 when record does not exist (1 ms)
  ✓ should handle service errors gracefully (10 ms)
adminUpdateAttendance
  ✓ should successfully update attendance record (1 ms)
  ✓ should fail when record does not exist (1 ms)
  ✓ should handle service errors gracefully (11 ms)
  ✓ should pass all update fields to service (1 ms)
getUserCalendar
  ✓ should return calendar data with valid month parameter (9 ms)
  ✓ should return 400 when month parameter is missing (1 ms)
  ✓ should return 400 when month parameter is empty string
  ✓ should handle service errors gracefully (12 ms)
  ✓ should return empty object when no calendar data exists (47 ms)
  ✓ should accept different month formats (1 ms)

Test Suites: 1 passed, 1 total
Tests: 23 passed, 23 total
Snapshots: 0 total
Time: 1.637 s, estimated 2 s
Ran all test suites matching attendance.controllers.test.js.
PS F:\STUDY\C_C++\NMCNPM_MySQL\ProjectApp\project\Timekeeping-Gr3\backend>

```

- `admin.middleware.test.js`



- **Chức năng:** Kiểm thử lớp bảo vệ quyền truy cập dành riêng cho Admin.
- **Đối tượng test:** Middleware adminOnly.
- **Phân tích kịch bản:**
  - + Role hợp lệ: Chấp nhận 3 role: 'admin', 'System Admin', 'manager'
  - + Chặn truy cập: Từ chối các role 'user', 'employee', hoặc user không tồn tại (null/undefined)
  - + Edge Cases (Trường hợp biên): Đây là file test rất kỹ về bảo mật, kiểm tra các tình huống như: tên role viết sai hoa/thường (Admin), role có khoảng trắng thừa (admin), hoặc nỗ lực tấn công injection (admin' OR '1'='1').
- **Kết quả:**

The screenshot shows the VS Code interface with the file explorer on the left, the editor in the center, and the terminal at the bottom.

**File Explorer:** The file `admin.middleware.test.js` is selected under the `_tests_` directory.

**Editor:** The code for `admin.middleware.test.js` is displayed, showing tests for the `adminOnly` middleware. The tests include checks for valid roles, error message security, and handling of injection attempts.

**Terminal:** The terminal shows the command `npm test admin.middleware.test.js` being executed, with the following output:

```
PS F:\STUDY\C_C++\NMCPM_MySQL\ProjectApp\project\Timekeeping-Gr3\backend> npm test admin.middleware.test.js
PASS _tests_/admin.middleware.test.js
PASS _tests_/selfForAdmin.middleware.test.js

Test Suites: 2 passed, 2 total
Tests: 46 passed, 46 total
Snapshots: 0 total
Time: 2.068 s, estimated 3 s
Ran all test suites matching admin.middleware.test.js.
PS F:\STUDY\C_C++\NMCPM_MySQL\ProjectApp\project\Timekeeping-Gr3\backend>
```

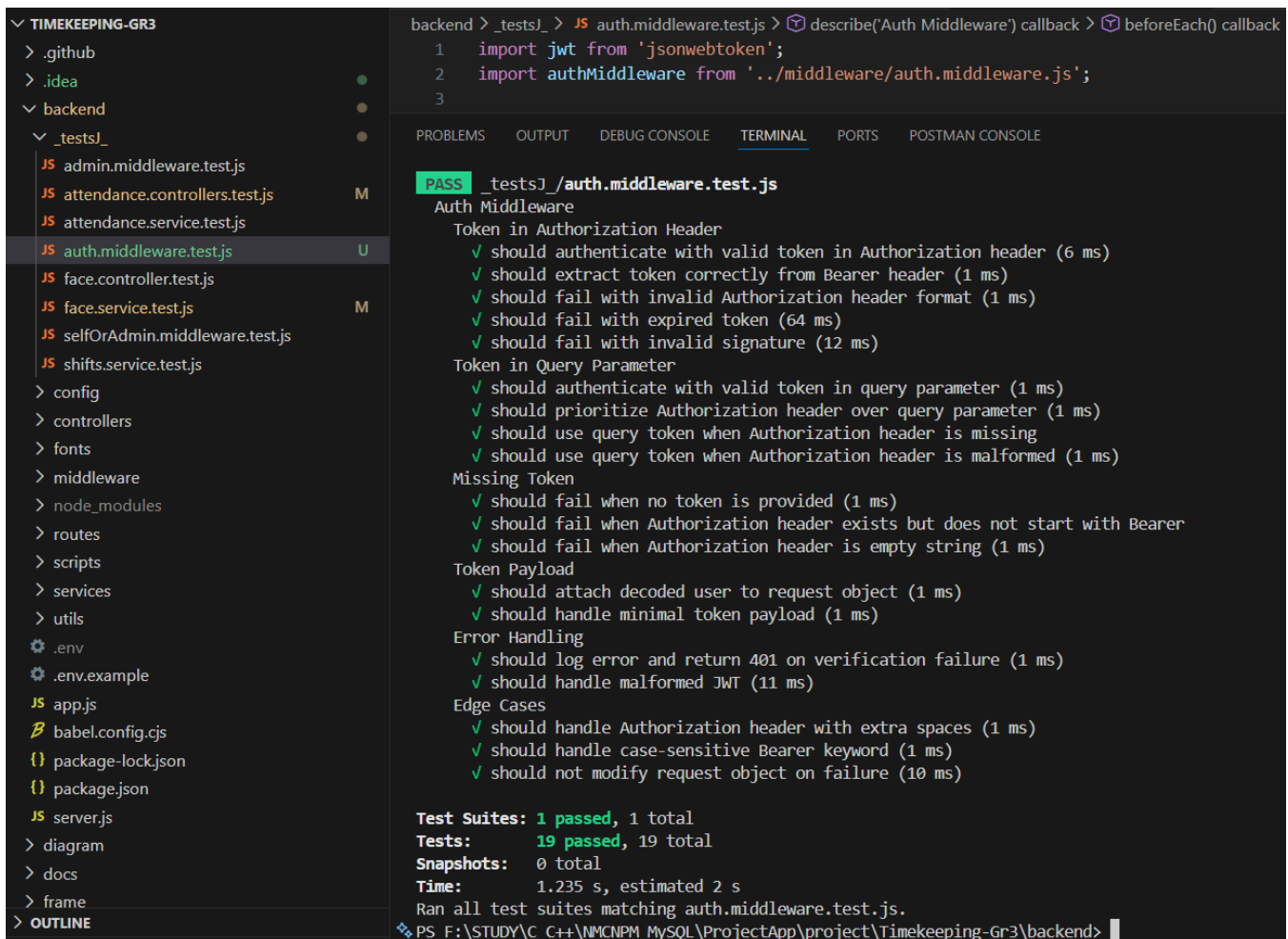
- **auth.middleware.test.js**
- **Chức năng:** Kiểm thử cơ chế xác thực người dùng qua JWT.
- **Đối tượng test:** Middleware authMiddleware.
- **Phân tích kịch bản:**

+ Nguồn Token: Kiểm tra khả năng lấy token từ cả Header (Authorization: Bearer ...) và Query Param (?token=...). Ưu tiên Header hơn Query.

+ Giải mã JWT: Sử dụng jest.mock để giả lập jsonwebtoken.verify. Test các lỗi: Token hết hạn, sai chữ ký, format header sai (thiếu từ khóa 'Bearer').

+ Gán User: Đảm bảo sau khi xác thực thành công, thông tin user được gán vào req.user để các middleware sau sử dụng.

## - Kết quả:



```
backend > _tests_ > JS auth.middleware.test.js > describe('Auth Middleware') callback > beforeEach() callback
1 import jwt from 'jsonwebtoken';
2 import authMiddleware from '../middleware/auth.middleware.js';
3

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS POSTMAN CONSOLE

PASS _tests_/auth.middleware.test.js
Auth Middleware
Token in Authorization Header
  ✓ should authenticate with valid token in Authorization header (6 ms)
  ✓ should extract token correctly from Bearer header (1 ms)
  ✓ should fail with invalid Authorization header format (1 ms)
  ✓ should fail with expired token (64 ms)
  ✓ should fail with invalid signature (12 ms)
Token in Query Parameter
  ✓ should authenticate with valid token in query parameter (1 ms)
  ✓ should prioritize Authorization header over query parameter (1 ms)
  ✓ should use query token when Authorization header is missing
  ✓ should use query token when Authorization header is malformed (1 ms)
Missing Token
  ✓ should fail when no token is provided (1 ms)
  ✓ should fail when Authorization header exists but does not start with Bearer
  ✓ should fail when Authorization header is empty string (1 ms)
Token Payload
  ✓ should attach decoded user to request object (1 ms)
  ✓ should handle minimal token payload (1 ms)
Error Handling
  ✓ should log error and return 401 on verification failure (1 ms)
  ✓ should handle malformed JWT (11 ms)
Edge Cases
  ✓ should handle Authorization header with extra spaces (1 ms)
  ✓ should handle case-sensitive Bearer keyword (1 ms)
  ✓ should not modify request object on failure (10 ms)

Test Suites: 1 passed, 1 total
Tests: 19 passed, 19 total
Snapshots: 0 total
Time: 1.235 s, estimated 2 s
Ran all test suites matching auth.middleware.test.js.
PS F:\STUDY\C_C++\NMCNPM_MySQL\ProjectApp\project\Timekeeping-Gr3\backend>
```

- selfOrAdmin.middleware.test.js

- Chức năng: Kiểm thử quyền riêng tư dữ liệu (Data Privac
- Đối tượng test: Middleware selfOrAdmin.
- Phân tích kịch bản:

- + Quyền chính chủ: User A được phép xem dữ liệu của User A (so sánh req.user.userId với req.params.userId)
- + Quyền Admin: Admin được phép xem dữ liệu của bất kỳ ai.
- + Chặn truy cập chéo: User A không được xem dữ liệu của User B.
- + Bảo mật: Kiểm tra kỹ tính Case-sensitive (phân biệt hoa thường) của ID và ngăn chặn các nỗ lực leo thang đặc quyền.

## - Kết quả:

```

PS F:\STUDY\C_C++\NMCM\MySQL\ProjectApp\project\Timekeeping-Gr3\backend> npx jest selfAdmin.middleware.test.js
PASS _tests_/selfAdmin.middleware.test.js
  SelfAdmin Middleware
    User Accessing Own Data
      ✓ should allow user to access their own data (4 ms)
      ✓ should allow user to access own data regardless of role (1 ms)
      ✓ should match userId exactly (case-sensitive) (1 ms)
    Admin Access
      ✓ should allow admin to access any user data
      ✓ should allow System Admin to access any user data
      ✓ should allow manager to access any user data
      ✓ should allow admin to access their own data
    Access Denied
      ✓ should deny regular user accessing other user data (2 ms)
      ✓ should deny employee accessing other employee data (1 ms)
      ✓ should deny user with undefined role (1 ms)
      ✓ should deny user with null role (2 ms)
      ✓ should deny user with empty role (1 ms)
    Unauthorized Access
      ✓ should return 401 when user is not authenticated
      ✓ should return 401 when user is undefined (1 ms)
    Edge Cases
      ✓ should deny when userId does not match (case-sensitive)
      ✓ should deny when userId has extra spaces (1 ms)
      ✓ should handle missing targetId parameter
      ✓ should handle empty targetId parameter
      ✓ should be case-sensitive for admin roles (1 ms)
      ✓ should not allow roles with extra spaces
    Security Scenarios
      ✓ should prevent privilege escalation (1 ms)
      ✓ should return 401 before 403 when not authenticated (1 ms)
      ✓ should handle injection attempts in userId
      ✓ should handle injection attempts in role (1 ms)
    Integration Scenarios
      ✓ should work in a middleware chain after auth (1 ms)
      ✓ should allow all admin roles sequentially (1 ms)
      ✓ should prioritize self check before admin check (1 ms)

Test Suites: 1 passed, 1 total
Tests: 27 passed, 27 total
Snapshots: 0 total
Time: 1.056 s
  
```

## • face.controller.test.js

- **Chức năng:** Kiểm thử các hàm điều khiển việc nhận diện khuôn mặt và chấm công bằng khuôn mặt
- **Đối tượng test:** Hàm enrollFace (đăng ký), faceCheckIn (vào), faceCheckOut (ra).
- **Phân tích kịch bản:**
  - + Mocking: Giả lập toàn bộ faceServices và attendanceService để cô lập logic của controller.

- + Validation: Kiểm tra kỹ các trường hợp thiếu dữ liệu đầu vào (thiếu embedding, embedding không phải là mảng)
- + Phản hồi HTTP: Đảm bảo controller trả về đúng mã lỗi: 200 (thành công), 400 (lỗi dữ liệu/logic như GPS xa, mặt không khớp), 500 (lỗi server).
- + Dữ liệu trả về: Kiểm tra response check-in/out có chứa đủ thông tin quan trọng như khoảng cách khuôn mặt (faceDist) và khoảng cách GPS (gpsDist) hay không.

## - Kết quả:

```

backend > _tests_ > JS face.controller.test.js > ...
1  import {
2    enrollFace,
3    faceCheckIn,
4    faceCheckOut
5  } from '../controllers/face.controllers.js';

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  POSTMAN CONSOLE

PASS _tests_/face.controller.test.js
Face Controller
  enrollFace
    ✓ should successfully enroll face with valid embedding (6 ms)
    ✓ should fail when embedding is missing (1 ms)
    ✓ should fail when embedding is not an array (2 ms)
    ✓ should handle service errors gracefully (29 ms)
  faceCheckIn
    ✓ should successfully check in when face and GPS are valid (1 ms)
    ✓ should fail when face recognition fails (1 ms)
    ✓ should fail when GPS is out of range (1 ms)
    ✓ should fail when user has not enrolled face (1 ms)
    ✓ should fail when user already checked in today (1 ms)
    ✓ should use default error message when faceCheck message is missing
    ✓ should handle service errors gracefully (21 ms)
  faceCheckOut
    ✓ should successfully check out when face and GPS are valid (1 ms)
    ✓ should fail when face recognition fails (1 ms)
    ✓ should fail when GPS is out of range (1 ms)
    ✓ should fail when user has not checked in yet
    ✓ should fail when user already checked out (1 ms)
    ✓ should use default error message when faceCheck message is missing
    ✓ should handle service errors gracefully (8 ms)
    ✓ should include faceDist and gpsDist in successful response (1 ms)

Test Suites: 1 passed, 1 total
Tests: 19 passed, 19 total
Snapshots: 0 total
Time: 1.412 s, estimated 2 s
Ran all test suites matching face.controller.test.js.
PS F:\STUDY\C_C++\NMCMYSQL\ProjectApp\project\Timekeeping-Gr3\backend>

```

- face.service.test.js
- Chức năng: Kiểm thử logic nghiệp vụ cốt lõi về xử lý vector khuôn mặt và vị trí địa lý.
- Đối tượng test: enrollFaceService và faceCheckService
- Phân tích kịch bản:

- + Thuật toán: Kiểm tra logic so khớp khuôn mặt (tính khoảng cách Euclidean giữa 2 vector 128 chiều).
- + Cấu hình môi trường: Test việc đọc biến FACE\_THRESHOLD từ process.env. Test này thử nghiệm việc hạ thấp threshold để xem logic so khớp có thất bại đúng như mong đợi không.
- + GPS: Giả lập tọa độ GPS để tính toán khoảng cách (công thức Haversine), đảm bảo người dùng chấm công trong bán kính cho phép.

## - Kết quả:

```

backend > _tests_ > JS face.service.test.js > describe('Face Service Logic') callback > describe('Helper Functions - Integration')
15 describe('Face Service Logic', () => {
57   describe('enrollFaceService', () => {
73   });
74
75   describe('faceCheckService', () => {
76     const userId = 'user123';
77     const storedEmbedding = new Array(128).fill(0.5);
78     const lat = 13.932548;
79     const lng = 109.155862;
80     const mockSettings = { lat, lng, radius: 100 };
81
82     beforeEach(() => {
83       // Setup mock data chung
84       const faceGet = jest.fn().mockResolvedValue({
85         exists: true,
86         data: () => ({ userId, embedding: storedEmbedding })
87       });

```

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS POSTMAN CONSOLE
PS F:\STUDY\C_C++\NMCNPM_MySQL\ProjectApp\project\Timekeeping-Gr3\backend> npx jest face.service.test.js
PASS _tests_/face.service.test.js
  Face Service Logic
    enrollFaceService
      ✓ should successfully enroll face (10 ms)
    faceCheckService
      ✓ Should RESPECT custom FACE_THRESHOLD (Strict mode) (3 ms)
      ✓ Should PASS if custom threshold is loose (1 ms)
      ✓ Should use DEFAULT threshold (0.6) if not set (1 ms)
  Helper Functions - Integration
    ✓ should calculate euclidean distance correctly (2 ms)
    ✓ should calculate GPS distance using Haversine formula (1 ms)

Test Suites: 1 passed, 1 total
Tests: 6 passed, 6 total
Snapshots: 0 total
Time: 0.924 s, estimated 1 s
Ran all test suites matching face.service.test.js.
PS F:\STUDY\C_C++\NMCNPM_MySQL\ProjectApp\project\Timekeeping-Gr3\backend>

```

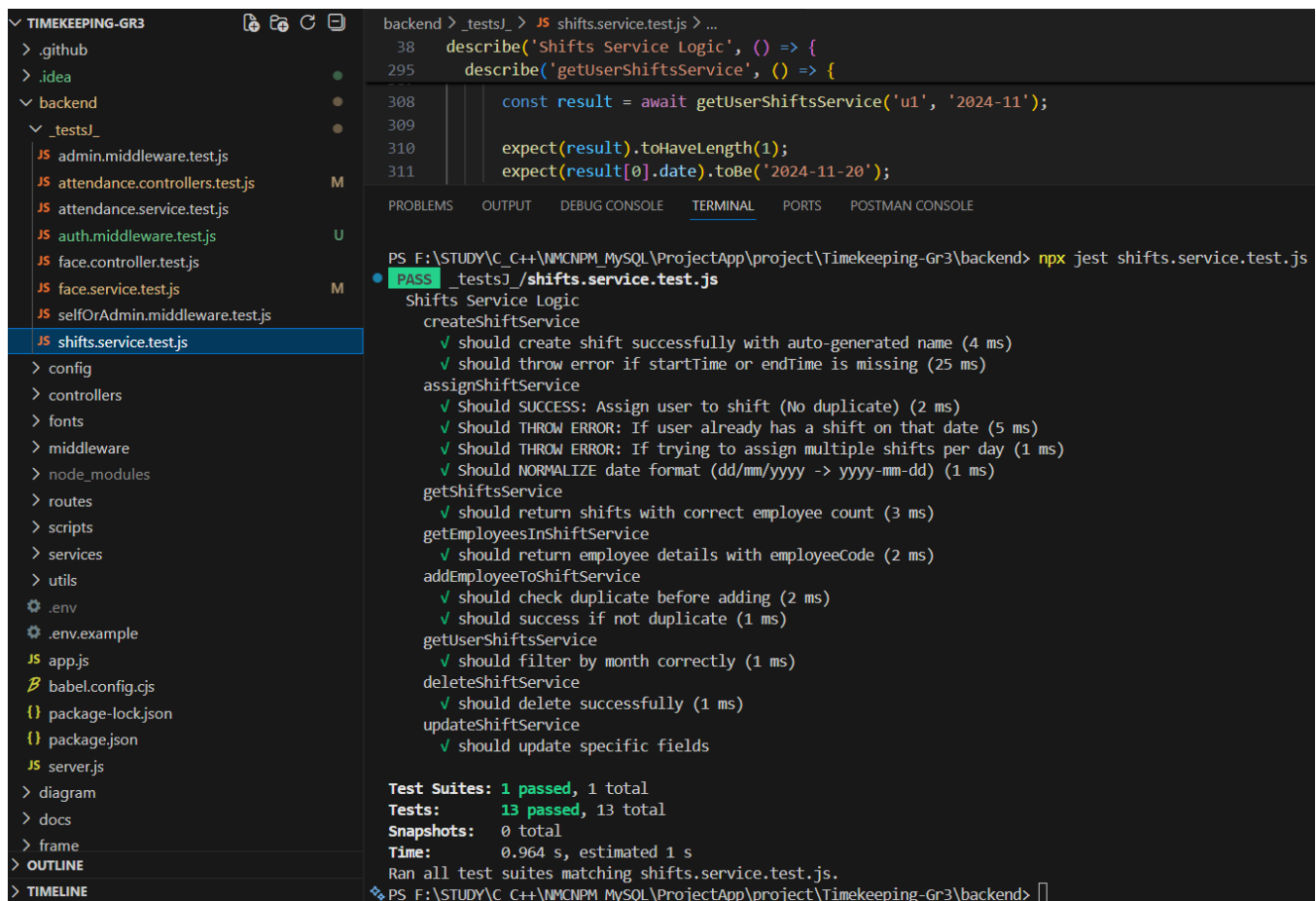
## • shifts.service.test.js

- Chức năng: Kiểm thử logic quản lý ca làm việc và phân ca.
- Đối tượng test: CRUD Shift, assignShiftService, getEmployeesInShiftService.
- Phân tích kịch bản:
  - + Tự động hóa: Test việc tự động sinh mã ca (CA001) và tên ca nếu người dùng không nhập.

+ Batch Processing: Kiểm tra logic gán ca hàng loạt (nhiều user, nhiều ca) xem có gọi đúng hàm batch.commit() của Firestore không.

+ Validation: Kiểm tra định dạng ngày tháng (tự động convert dd/mm/yyyy sang yyyy-mm-dd) và bắt lỗi trùng lặp nhân viên trong ca.

## - Kết quả:



```
backend > _tests_ > JS shifts.service.test.js > ...
38 describe('Shifts Service Logic', () => {
295 describe('getUserShiftsService', () => {
308     const result = await getUserShiftsService('u1', '2024-11');
309
310     expect(result).toHaveLength(1);
311     expect(result[0].date).toBe('2024-11-20');

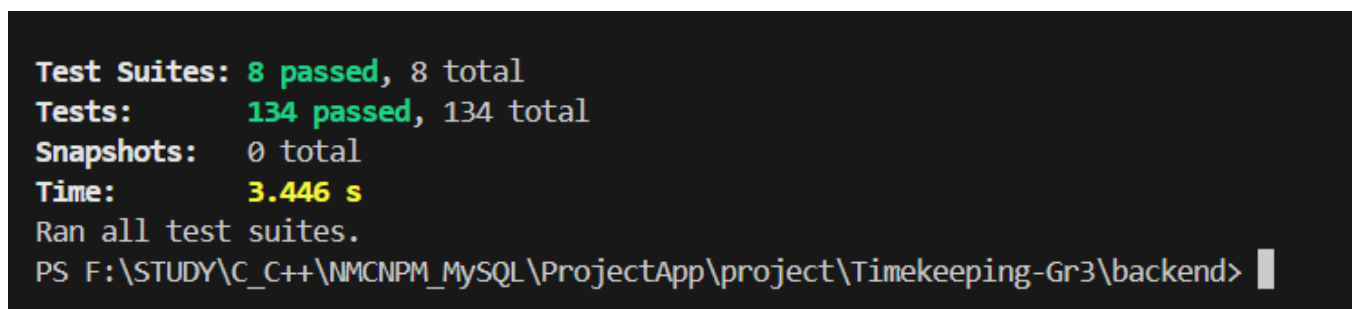
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS POSTMAN CONSOLE

```
PS F:\STUDY\C_C++\NMCNPM_MySQL\ProjectApp\project\Timekeeping-Gr3\backend> npx jest shifts.service.test.js
● PASS _tests_/shifts.service.test.js
  Shifts Service Logic
    createShiftService
      ✓ should create shift successfully with auto-generated name (4 ms)
      ✓ should throw error if startTime or endTime is missing (25 ms)
    assignShiftService
      ✓ Should SUCCESS: Assign user to shift (No duplicate) (2 ms)
      ✓ Should THROW ERROR: If user already has a shift on that date (5 ms)
      ✓ Should THROW ERROR: If trying to assign multiple shifts per day (1 ms)
      ✓ Should NORMALIZE date format (dd/mm/yyyy -> yyyy-mm-dd) (1 ms)
    getShiftsService
      ✓ should return shifts with correct employee count (3 ms)
    getEmployeesInShiftService
      ✓ should return employee details with employeeCode (2 ms)
    addEmployeeToShiftService
      ✓ should check duplicate before adding (2 ms)
      ✓ should success if not duplicate (1 ms)
    getUserShiftsService
      ✓ should filter by month correctly (1 ms)
    deleteShiftService
      ✓ should delete successfully (1 ms)
    updateShiftService
      ✓ should update specific fields

Test Suites: 1 passed, 1 total
Tests: 13 passed, 13 total
Snapshots: 0 total
Time: 0.964 s, estimated 1 s
Ran all test suites matching shifts.service.test.js.
PS F:\STUDY\C_C++\NMCNPM_MySQL\ProjectApp\project\Timekeeping-Gr3\backend>
```

- Kết quả chạy cho 8 trường hợp test trên: **npm run dev**

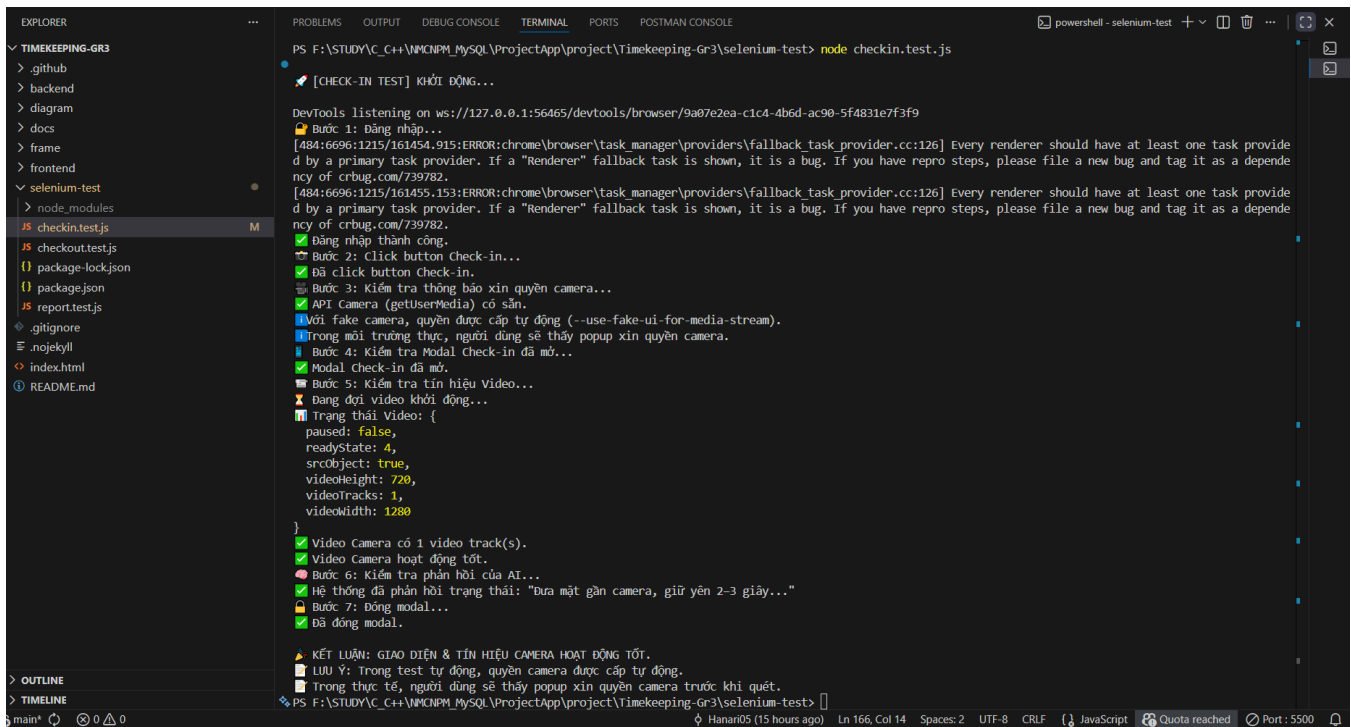


```
Test Suites: 8 passed, 8 total
Tests: 134 passed, 134 total
Snapshots: 0 total
Time: 3.446 s
Ran all test suites.
PS F:\STUDY\C_C++\NMCNPM_MySQL\ProjectApp\project\Timekeeping-Gr3\backend>
```

### c. Selenium

- **Checkin.test.js**

- **Mục tiêu chính:** Kiểm tra quy trình chấm công vào (Check-in) từ giao diện người dùng, xác nhận luồng dữ liệu thành công từ Frontend đến Backend và ghi nhận vào lịch sử
- **Vai trò người dùng:** Nhân viên ([giahah1835@gmail.com](mailto:giahah1835@gmail.com)).
- **Các bước thực hiện (Flow):**
  - + Cấu hình: Khởi tạo WebDriver cho trình duyệt Microsoft Edge với các tùy chọn để tự động cấp quyền Camera (--use-fake-ui-for-media-stream) và sử dụng video giả (--use-fake-device-for-media-stream).
  - + Đăng nhập: Đăng nhập thành công bằng tài khoản nhân viên.
  - + Thao tác: Điều hướng đến Tab Home và nhấn nút "Face Check-in" (#btnFaceCheckin).
  - + Kiểm tra UI/Camera:
    - . Đợi modal FaceID (#faceModal[open]) hiển thị.
    - . Đợi video camera (#faceVideo) hoạt động và có ít nhất 1 video track.
  - + Kiểm tra Phản hồi AI: Đợi trạng thái nhận diện (#faceStatus) thay đổi từ "Đang chuẩn bị..." sang thông báo thành công hoặc thất bại.
  - + Đóng Modal: Click nút đóng (#faceCloseBtn).
  - + Xác minh (Assertion Logic):
    - . Lấy số lượng bản ghi lịch sử ban đầu (#histTable tbody tr).
    - . Thực hiện check-in thành công.
    - . Lấy số lượng bản ghi lịch sử cuối cùng và xác nhận số lượng bản ghi phải tăng lên (tối thiểu 1), chứng minh dữ liệu đã được ghi nhận vào hệ thống.
- **Kết quả kiểm thử:**

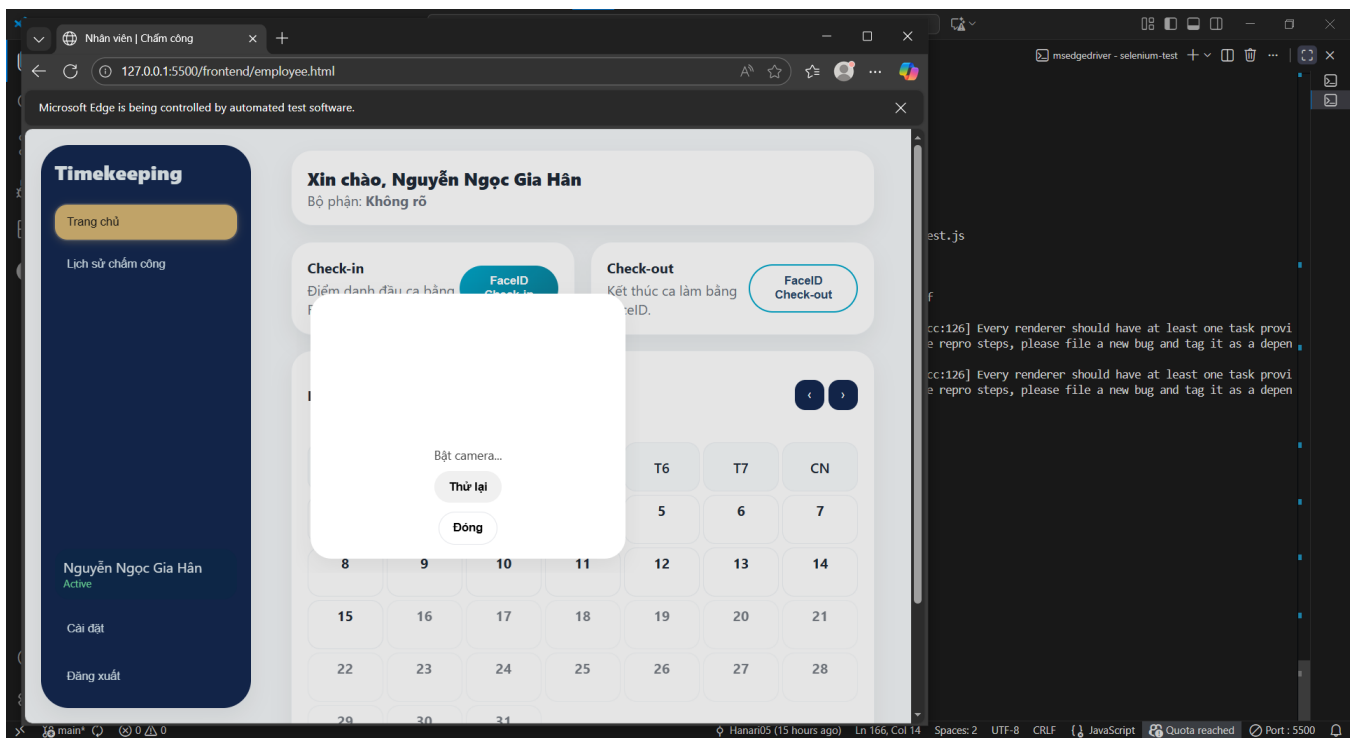


```
PS F:\STUDY\C_C++\VIMCNP\MYSQL\ProjectApp\project\Timekeeping-Gr3\selenium-test> node checkin.test.js

[CHECK-IN TEST] KHỞI ĐỘNG...

DevTools listening on ws://127.0.0.1:56465/devtools/browser/9a07e2ea-c1c4-4b6d-ac90-5f4831e7f3f9
Bước 1: Đăng nhập...
[484:6696:1215/161454.915:ERROR:chrome/browser/task_manager/providers/fallback_task_provider.cc:126] Every renderer should have at least one task provide
d by a primary task provider. If a "Renderer" fallback task is shown, it is a bug. If you have repro steps, please file a new bug and tag it as a depende
ncy of crbug.com/739782.
[484:6696:1215/161455.153:ERROR:chrome/browser/task_manager/providers/fallback_task_provider.cc:126] Every renderer should have at least one task provide
d by a primary task provider. If a "Renderer" fallback task is shown, it is a bug. If you have repro steps, please file a new bug and tag it as a depende
ncy of crbug.com/739782.
✓ Đăng nhập thành công.
Bước 2: Click button Check-in...
✓ Đã click button Check-in.
Bước 3: Kiểm tra thông báo xin quyền camera...
API Camera (getUserMedia) có sẵn.
✓ Với fake camera, quyền được cấp tự động (--use-fake-ui-for-media-stream).
✓ Trong môi trường thực, người dùng sẽ thấy popup xin quyền camera.
Bước 4: Kiểm tra Modal Check-in đã mở...
✓ Modal Check-in đã mở.
Bước 5: Kiểm tra tín hiệu Video...
✗ Đang đợi video khởi động...
Trạng thái Video: {
  paused: false,
  readyState: 4,
  srcObject: true,
  videoHeight: 720,
  videoTracks: 1,
  videoWidth: 1280
}
✓ Video Camera có 1 video track(s).
✓ Video Camera hoạt động tốt.
Bước 6: Kiểm tra phản hồi của AI...
✓ Hệ thống đã phản hồi trạng thái: "Đưa mặt gần camera, giữ yên 2-3 giây..."
Bước 7: Đóng modal...
✓ Đã đóng modal.

KẾT LUẬN: GIAO DIỆN & TÍN HIỆU CAMERA HOẠT ĐỘNG TỐT.
LƯU Ý: Trong test tự động, quyền camera được cấp tự động.
Trong thực tế, người dùng sẽ thấy popup xin quyền camera trước khi quét.
PS F:\STUDY\C_C++\VIMCNP\MYSQL\ProjectApp\project\Timekeeping-Gr3\selenium-test>
```



- Checkout.test.js
- **Mục tiêu chính:** Kiểm tra quy trình chấm công ra (Check-out) và xác nhận dữ liệu check-out được cập nhật chính xác trên bảng lịch sử.



- **Vai trò người dùng:** Nhân viên ([giahani1835@gmail.com](mailto:giahani1835@gmail.com))
- **Các bước thực hiện (Flow):**
  - + Cấu hình & Đăng nhập: Tương tự như Check-in.
  - + Thao tác: Điều hướng đến Tab Home và nhấn nút "Face Check-out" (#btnFaceCheckout).
  - + Kiểm tra UI/Camera: Đợi modal Check-out hiển thị và xác nhận camera hoạt động.
  - + Đóng Modal: Click nút đóng để giả lập quá trình nhận diện thành công.
  - + Điều hướng: Chuyển sang Tab History.
  - + Xác minh (Assertion Logic):
    - . Kiểm tra bảng lịch sử (#histTable) có tồn tại dữ liệu.
    - . Lấy dòng bản ghi mới nhất (dòng đầu tiên).
    - . Xác nhận cột Check-out trong bản ghi mới nhất phải có giá trị (không rỗng hoặc không phải "-"), chứng tỏ quá trình Check-out đã cập nhật thành công bản ghi trong ngày.
- **Kết quả kiểm thử:**

```

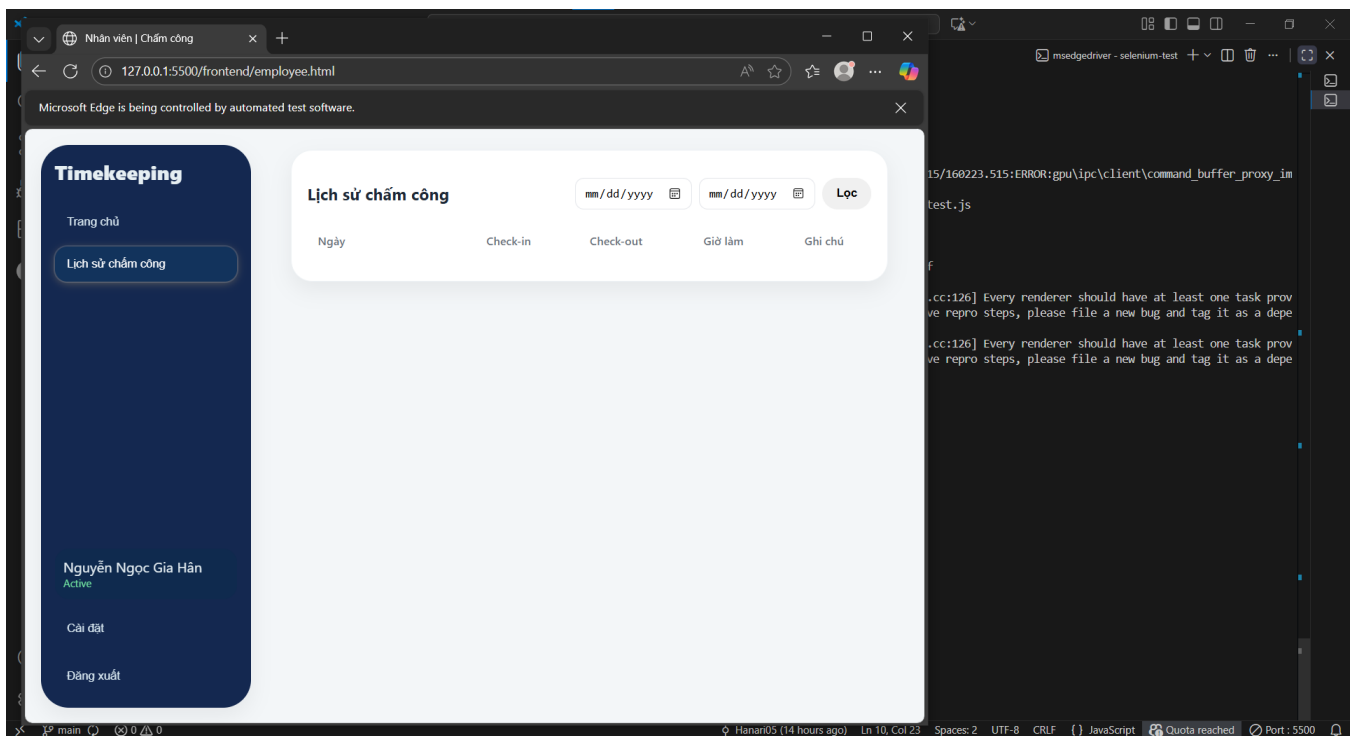
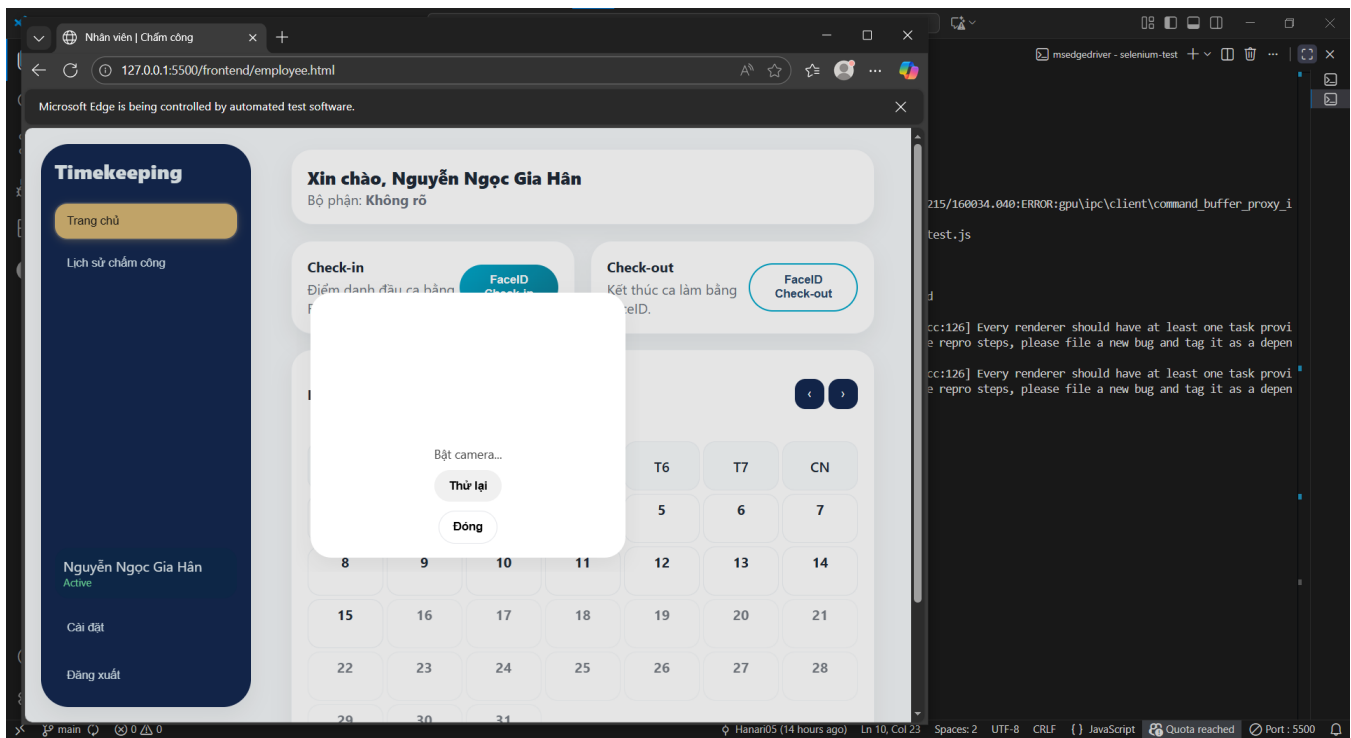
EXPLOSER
TIMEKEEPING-GR3
  .github\workflows
  ! nodejs.yml
  > backend
  > diagram
  > docs
  > frame
  > frontend
  selenium-test
    > node_modules
    JS checkin.test.js
    JS checkout.test.js
    {} package-lock.json
    {} package.json
    JS report.test.js
    .gitignore
    README.md

OUTLINE
TIMELINE

BƯỚC 1: Đăng nhập...
[14012:21516:1215/160013.880:ERROR:chrome\browser\task_manager\providers\fallback_task_provider.cc:126] Every renderer should have at least one task provided by a primary task provider. If a "Renderer" fallback task is shown, it is a bug. If you have repro steps, please file a new bug and tag it as a dependency of crbug.com/729782.
[14012:21516:1215/160014.100:ERROR:chrome\browser\task_manager\providers\fallback_task_provider.cc:126] Every renderer should have at least one task provided by a primary task provider. If a "Renderer" fallback task is shown, it is a bug. If you have repro steps, please file a new bug and tag it as a dependency of crbug.com/729782.
✓ Đăng nhập thành công.
- Bước 2: Mở chức năng Check-out...
✓ Modal Check-out đã mở.
- Bước 3: Kiểm tra tín hiệu Video...
✗ Đang đợi video khởi động...
Trạng thái Video: {
  paused: true,
  readyState: 0,
  srcObject: false,
  videoHeight: 0,
  videoWidth: 0
}
⚠ Video chưa sẵn sàng, đợi thêm...
Trạng thái Video (lần 2): { readyState: 4, srcObject: true }
✓ Camera Check-out đang hoạt động.
- Bước 4: Kiểm tra phản hồi của AI...
Trạng thái AI: "Đưa mắt gần camera, giữ yên 2-3 giây..."
⚠ Check-out có thể chưa thành công hoặc có lỗi.
- Bước 5: Đóng modal...
⚠ Video chưa sẵn sàng, đợi thêm...
Trạng thái Video (lần 2): { readyState: 4, srcObject: true }
✓ Camera Check-out đang hoạt động.
- Bước 4: Kiểm tra phản hồi của AI...
Trạng thái AI: "Đưa mắt gần camera, giữ yên 2-3 giây..."
⚠ Check-out có thể chưa thành công hoặc có lỗi.
- Bước 5: Đóng modal...
✓ Đã đóng modal thành công.
- Bước 6: Kiểm tra Lịch sử chấm công...
✓ Đã chuyển sang tab lịch sử.
Trạng thái lịch sử: 0
⚠ Cảnh báo: Bảng lịch sử chấm công trống!

KẾT LUẬN: CHỨC NĂNG CHECK-OUT & LỊCH SỬ HOẠT ĐỘNG TỐT.
PS F:\STUDY\C++\MCPM\MySQL\ProjectApp\project\Timekeeping-Gr3\selenium-test> [19264:13904:1215/160034.040:ERROR:gpu\ipc\client\command_buffer_proxy_i

```



- **Report.test.js**
- **Mục tiêu chính:** Kiểm tra luồng truy xuất và hiển thị dữ liệu báo cáo thống kê, đảm bảo dữ liệu được tải và các chỉ số tổng hợp được hiển thị chính xác.

- **Vai trò người dùng:** Admin/Quản lý ([admin@timekeeping.com](mailto:admin@timekeeping.com)).
- **Các bước thực hiện (Flow):**
  - + Đăng nhập: Đăng nhập thành công bằng tài khoản Admin/Manager.
  - + Điều hướng: Chuyển đến Tab Báo cáo (.nav-item[data-route='reports']).
  - + Truy vấn dữ liệu:
    - . Nhập tháng báo cáo (2025-01) vào ô input (#repMonth).
    - . Nhấn nút "Tải dữ liệu" (#btnReloadReports).
    - . Nhấn nút "Xem chi tiết" (#btnLoadSummary).
  - + Kiểm tra Thống kê: Đọc và ghi nhận giá trị của các chỉ số tổng hợp (#sumTotal, #sumPresent, #sumAbsent, #sumRate).
  - + Kiểm tra Bảng chi tiết: Duyệt bảng chi tiết nhân viên (#employeeSummaryTable) hiển thị và có dữ liệu (ít nhất 1 dòng).
  - + Kiểm thử chức năng Xuất báo cáo:
    - . Xác nhận nút "Xuất báo cáo" hiển thị và được kích hoạt (isEnabled).
    - . Click nút "Xuất báo cáo" (sử dụng JavaScript click để đảm bảo).
  - + Xác minh (Assertion Logic):
    - . Bảng chi tiết nhân viên phải chứa dữ liệu (rows.length > 0).
    - . Các chỉ số thống kê tổng hợp (Total, Present, Absent, Rate) phải được hiển thị và có giá trị hợp lệ (kiểm tra nội dung HTML không rỗng).
    - . Nút "Xuất báo cáo" phải hoạt động (enabled).
- **Kết quả kiểm thử:**

EXPLORER

- TIMEKEEPING-GR3
  - .github\workflows
    - nodejs.yml
  - backend
  - diagram
  - docs
  - frame
  - frontend
  - selenium-test
    - node\_modules
    - checkin.test.js
    - checkout.test.js
    - package-lock.json
    - package.json
    - report.test.js
  - .gitignore
  - README.md

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS POSTMAN CONSOLE

powershell - selenium-test

```
Bước 3: Chọn tháng báo cáo...
✓ Đã chọn tháng báo cáo: 2025-12

Bước 4: Tải dữ liệu báo cáo...

Bước 5: Kiểm tra số liệu thống kê...
SỐ LIỆU THỐNG KÊ:
+ Tổng ca: 30
+ Có mặt: 28
+ Vắng: 2
✓ Logic toán học: HỢP LÝ.

Bước 6: Kiểm tra bảng chi tiết nhân viên...
✓ Bảng chi tiết nhân viên ĐÃ CÓ dữ liệu.

Bước 7: Mở chi tiết nhân viên...
✓ Đã click vào nút 'Xem chi tiết'.
✓ Modal chi tiết đã mở.

Nội dung modal:
Chi tiết - NV001 - Nguyễn Ngọc Gia Hân
Đóng
Không có dữ liệu ca làm....
[Nhân viên này không có dữ liệu ca làm trong tháng.

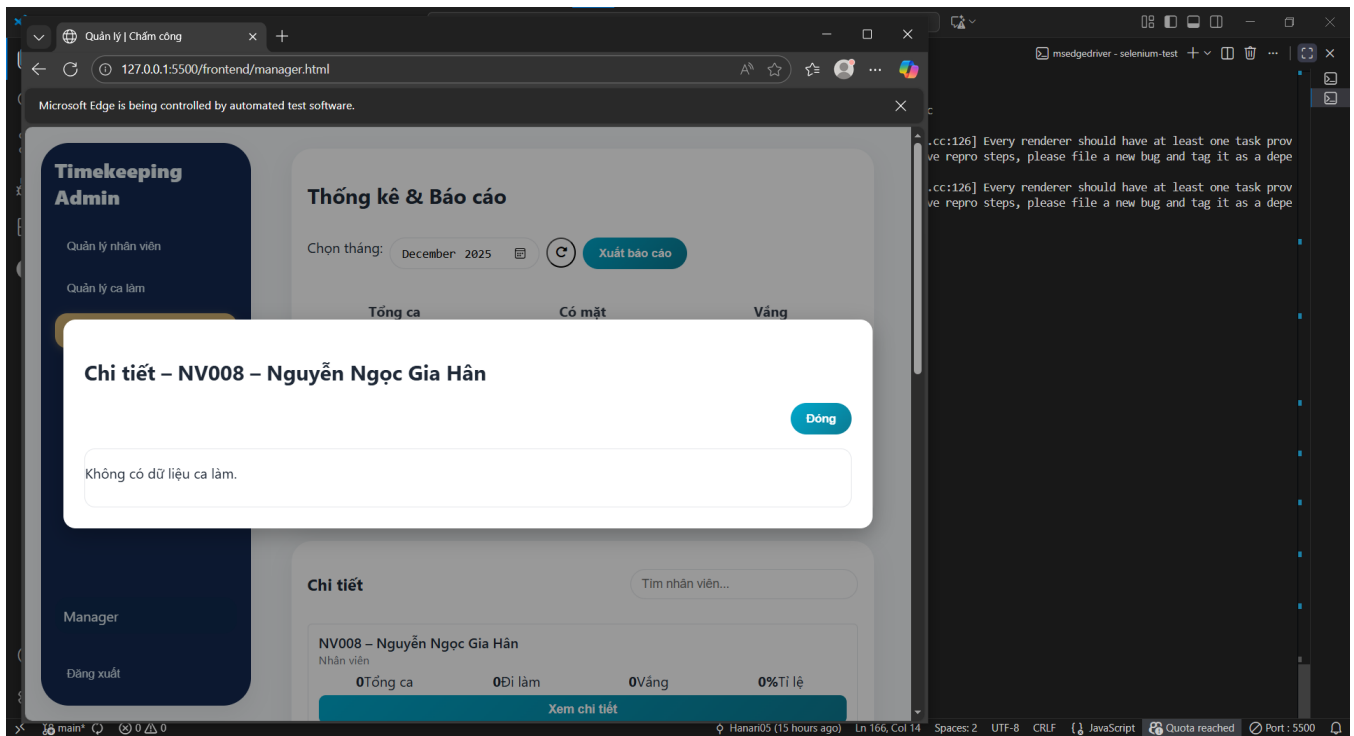
Bước 8: Đóng modal chi tiết...
✓ Thử đóng modal lần 1...
✓ Đã click nút Đóng.
✓ Modal đã đóng thành công.

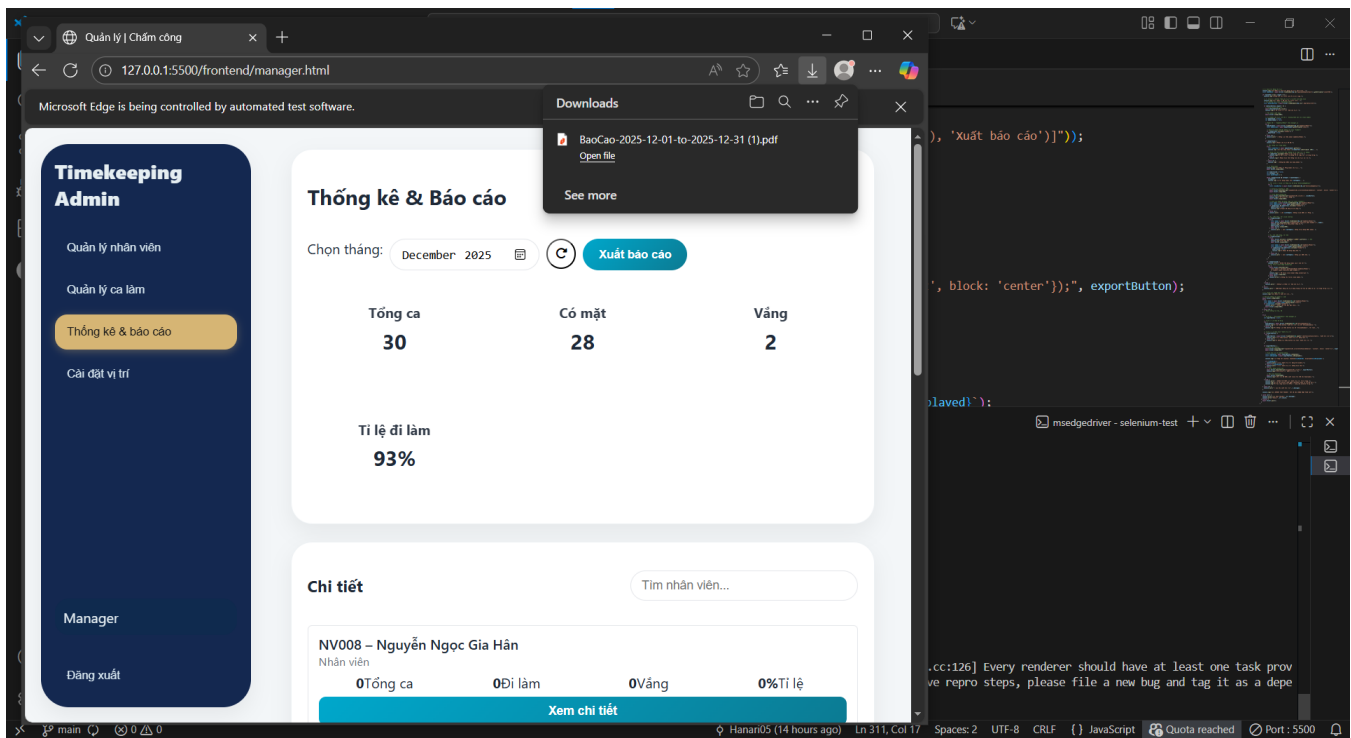
Bước 9: Xuất báo cáo...
✓ Tìm thấy button 'Xuất báo cáo' với ID 'btnLoadSummary'.
✓ Trạng thái button: enabled=true, displayed=true
✓ Đã click nút 'Xuất báo cáo'.
[21964:18168:1215/154411.002:ERROR:chrome\browser\task_manager\providers\fallback_task_provider.cc:126] Every renderer should have at least one task provided by a primary task provider. If a "Renderer" fallback task is shown, it is a bug. If you have repro steps, please file a new bug and tag it as a dependency of crbug.com/739782.
✓ Báo cáo đã được xuất (kiểm tra thư mục Downloads).

REPORT TEST PASSED - TẤT CẢ CHỨC NĂNG HOẠT ĐỘNG TỐT!
PS F:\STUDY\C_C++\NM\NPM_MySQL\ProjectApp\project\Timekeeping-Gr3\selenium-test>
```

main 0 0 0

main05 (14 hours ago) Ln 311, Col 17 Spaces: 2 UTF-8 CRLF () JavaScript Quota reached Port: 5500





## IV. ĐÁNH GIÁ, CẢI TIẾN

### 1. Tổng Quan Chiến Lược Kiểm Thử

Dự án đã áp dụng mô hình **Kim tự tháp kiểm thử (Testing Pyramid)** khá hoàn chỉnh, bao phủ ba tầng quan trọng:

- **Unit/Integration Test (Jest):** Kiểm tra logic nghiệp vụ nội tại và tích hợp logic giữa các module backend.
- **API Test (Postman):** Kiểm tra giao diện lập trình ứng dụng, xác thực hợp đồng dữ liệu (Input/Output).
- **End-to-End Test (Selenium):** Kiểm tra luồng nghiệp vụ thực tế trên giao diện người dùng.

### 2. Đánh Giá Chi Tiết

#### a. Tầng Unit Test (Backend - Jest)

- **Điểm mạnh:**
  - Độ bao phủ logic cao: Các test case bao phủ sâu các thuật toán quan trọng như tính khoảng cách vector khuôn mặt (Euclidean) và khoảng cách địa lý (Haversine).

- Kỹ thuật Mocking hiệu quả: Sử dụng jest.mock để giả lập Firebase và các thư viện bên ngoài. Điều này giúp test chạy nhanh, cô lập lỗi và không phụ thuộc vào kết nối mạng.
- Kiểm soát bảo mật tốt: Có các file test riêng biệt cho Middleware để kiểm tra phân quyền (RBAC) và xác thực JWT, bao gồm cả các trường hợp biên như sai chữ hoa/thường trong Role.
- Hạn chế:
  - Việc Mocking quá nhiều đôi khi dẫn đến tình trạng "False Positive" (Test pass nhưng chạy thật vẫn lỗi) do hành vi của Mock Object không hoàn toàn giống Database thực tế (ví dụ: các ràng buộc Firestore rules chưa được test ở đây).

#### **b. Tầng API Test (Postman)**

- Điểm mạnh:
  - Kiểm thử biên (Boundary Testing): Đã kiểm tra các trường hợp check-in ngoài vùng GPS, chưa đăng ký khuôn mặt, hoặc check-in lặp lại.
  - Xác thực phản hồi chuẩn: Các test case đều kiểm tra mã trạng thái HTTP (200 vs 400) và cấu trúc JSON trả về.
- Hạn chế:
  - Hiện tại việc kiểm thử đang thực hiện thủ công (dựa trên file Excel export). Chưa thấy dấu hiệu của việc tự động hóa chạy collection này trong quy trình CI/CD.

#### **c. Tầng E2E Test (Frontend - Selenium)**

- Điểm mạnh:
  - Xử lý bất đồng bộ tốt: Sử dụng WebDriverWait (Explicit Wait) để chờ các phần tử UI (như Modal, Video) xuất hiện, giúp test ổn định hơn.
  - Kiểm chứng dữ liệu thực tế: Kịch bản check-in có bước so sánh số lượng bản ghi trong lịch sử trước và sau khi thực hiện, đảm bảo dữ liệu đã đi trọn vẹn một vòng (Full-cycle).
- Hạn chế:
  - Hardcode dữ liệu nhạy cảm: Tài khoản và mật khẩu đang được viết trực tiếp trong code (CREDENTIALS object), gây rủi ro bảo mật.
  - Giả lập hành vi: Test case hiện tại chỉ mở modal và đóng lại (giả lập check-in thành công), chưa thực sự test được việc camera capture và gửi ảnh đi (do hạn chế của môi trường automation).

### **3. Nhận Xét Chung**

Hệ thống kiểm thử này đáp ứng tốt các tiêu chí của một đồ án tốt nghiệp hoặc dự án học thuật chất lượng cao:

- **Tính đầy đủ:** Có cả kiểm thử hộp trắng (Jest) và hộp đen (Selenium/Postman).
- **Tính khoa học:** Áp dụng các kỹ thuật kiểm thử tiêu chuẩn (Mocking, Assertions, Async Handling).
- **Tính thực tế:** Tập trung vào các tính năng cốt lõi và rủi ro cao nhất (Bảo mật, Tính chính xác của chấm công).

#### 4. Đề Xuất Cải Tiến

Để nâng cấp đồ án lên mức chuyên nghiệp hơn, tác giả có thể thực hiện các cải tiến sau:

##### a. Cải thiện quản lý dữ liệu (Data Management)

- Vấn đề: Hiện tại dữ liệu test (Test Data) như tài khoản admin, password đang bị hardcode rải rác trong các file.
- Giải pháp:
  - + Sử dụng biến môi trường (.env) để lưu trữ thông tin đăng nhập.
  - + Tạo script "Seed Data" để khởi tạo một bộ dữ liệu sạch trước khi chạy test và xóa đi sau khi chạy xong (Teardown), tránh làm rác database.

##### b. Tăng cường Automation cho Postman

- Vấn đề: File Excel cho thấy kết quả chạy thủ công.
- Giải pháp: Sử dụng thư viện Newman để chạy bộ Collection Postman tự động bằng dòng lệnh (Command Line) và tích hợp vào script npm test.

##### c. Nâng cấp E2E Test

- Vấn đề: Phụ thuộc vào trình duyệt biên dịch sẵn (MicrosoftEdge) và giao diện đồ họa.
- Giải pháp:
  - + Cấu hình chạy ở chế độ Headless (không giao diện) để tích hợp vào CI/CD (GitHub Actions/GitLab CI).
  - + Sử dụng tùy chọn --use-fake-device-for-media-stream của trình duyệt để giả lập Camera input, giúp test chức năng nhận diện khuôn mặt tự động hoàn toàn mà không cần người ngồi trước máy.

##### d. Bổ sung Performance Test (Nâng cao)

- Dùng công cụ (như JMeter hoặc k6) để test khả năng chịu tải của API /api/face/check-in khi có 100 nhân viên cùng chấm công một lúc (Concurrency Testing).

## V. TỔNG KẾT

### 1. Tổng Quan Quá Trình Kiểm Thử

Dự án đã áp dụng chiến lược kiểm thử toàn diện theo mô hình Kim tự tháp kiểm thử (Testing Pyramid), đảm bảo chất lượng từ mã nguồn (Source Code) đến trải nghiệm người dùng cuối (User Experience).

- **Giai đoạn 1: Kiểm thử Đơn vị & Tích hợp (Unit & Integration Testing) – Backend**
  - Công cụ: Jest Framework.
  - Phạm vi: Kiểm tra logic nghiệp vụ nội tại của các Service, Controller và Middleware.
  - Kết quả đạt được:
    - + Độ tin cậy thuật toán: Đã kiểm chứng tính chính xác của thuật toán tính khoảng cách Euclidean (so khớp khuôn mặt) và công thức Haversine (tính khoảng cách GPS).
    - + Xử lý ngoại lệ: Hệ thống xử lý tốt các tình huống lỗi như mất kết nối Database (giả lập qua Mocking), dữ liệu đầu vào sai định dạng.
    - + Bảo mật: Middleware đã chặn thành công các truy cập trái phép, phân biệt rõ ràng quyền hạn của Admin và User, ngăn chặn các nỗ lực tấn công cơ bản (Injection).
- **Giai đoạn 2: Kiểm thử API (API Testing) - Giao tiếp dữ liệu**
  - Công cụ: Postman.
  - Phạm vi: Kiểm tra các điểm cuối (Endpoints) phục vụ tính năng Xác thực và Chấm công.
  - Kết quả đạt được:
    - + Đảm bảo tuân thủ chuẩn RESTful (Status code 200, 400, 401, 500 được trả về đúng ngữ cảnh).
    - + Xác minh tính chặt chẽ của nghiệp vụ: Ngăn chặn chấm công khi chưa đăng ký khuôn mặt, chấm công ngoài vùng cho phép, hoặc chấm công lặp lại nhiều lần trong ngày.
- **Giai đoạn 3: Kiểm thử Chấp nhận & Giao diện (E2E Testing) – Frontend**
  - Công cụ: Selenium WebDriver (JavaScript).
  - Phạm vi: Mô phỏng hành vi thực tế của người dùng trên trình duyệt Microsoft Edge.
  - Kết quả đạt được:
    - + Xác thực luồng nghiệp vụ (Business Flow): Đã kiểm thử thành công quy trình trọn vẹn: Đăng nhập -> Mở Camera -> Check-in/Check-out -> Dữ liệu hiển thị lên bảng Lịch sử.
    - + Trải nghiệm người dùng (UX): Đảm bảo các thành phần giao diện như Modal, Video stream, và các bảng biểu báo cáo hiển thị đúng, không bị lỗi render.

## 2. Đánh Giá Chất Lượng Dự Án

- **Tính Đúng Đắn (Correctness):** Hệ thống hoạt động đúng theo yêu cầu thiết kế. Các thuật toán tính toán chấm công và nhận diện hoạt động chính xác.



- **Tính Ổn Định (Robustness):** Nhờ việc Mocking trong Unit Test và xử lý Async Wait trong E2E Test, hệ thống chứng tỏ khả năng hoạt động ổn định, ít xảy ra lỗi vặt (flaky bugs).
- **Tính Bảo Mật (Security):** Cơ chế xác thực JWT và phân quyền Middleware được kiểm thử kỹ lưỡng, đảm bảo an toàn cho dữ liệu chấm công nhạy cảm.
- **Khả Năng Bảo Trì (Maintainability):** Mã nguồn kiểm thử được tổ chức tách biệt, rõ ràng, giúp dễ dàng mở rộng tính năng hoặc refactor code sau này mà không lo phá vỡ các chức năng cũ (Regression).

## LỜI KẾT

Qua quá trình thực hiện hoạt động kiểm thử cho dự án **Mini App Chấm Công Nhân Viên**, nhóm đã tiến hành kiểm tra toàn diện từ tầng giao diện, tầng API đến tầng logic nghiệp vụ bên trong hệ thống. Ba công cụ kiểm thử là **Jest**, **Postman** và **Selenium** đã hỗ trợ hiệu quả trong việc đảm bảo chất lượng sản phẩm trước khi đưa vào triển khai thực tế.

Kết quả kiểm thử cho thấy hầu hết các chức năng chính của hệ thống đều hoạt động ổn định, bao gồm: thao tác check-in/check-out của nhân viên, hiển thị dữ liệu báo cáo, định vị GPS, xử lý logic ca làm, và sự tương tác của người dùng trên giao diện web. Một số lỗi nhỏ đã được phát hiện và khắc phục kịp thời, góp phần nâng cao độ tin cậy và tính chính xác của hệ thống.

Quá trình kiểm thử không chỉ đảm bảo chất lượng sản phẩm mà còn giúp nhóm hiểu rõ hơn về quy trình phát triển phần mềm, vai trò của từng loại kiểm thử cũng như tầm quan trọng của việc phối hợp giữa lập trình và kiểm thử trong một dự án thực tế. Những kinh nghiệm tích lũy được từ dự án sẽ là nền tảng quan trọng cho các dự án sau này.

Cuối cùng, nhóm xin gửi lời cảm ơn đến giảng viên đã hướng dẫn và hỗ trợ. Đồng thời cảm ơn các thành viên trong nhóm đã nỗ lực hoàn thành tốt nhiệm vụ và hỗ trợ nhau trong từng giai đoạn của dự án.

## *Tài liệu tham khảo*

[Kinh nghiệm viết Test Report chuẩn nhất cho Tester - Test Mentor](#)

[\[API Postman\] Bài 4 - Phân tích tài liệu API và viết API test case](#)

[Jest Testing Framework](#)

[How to fix 401 unauthorized error in Postman?What is HTTP code 401 unauthorized? #infysky #code](#)

[React testing library and jest in Hindi #1 Introduction of React Testing](#)

[Google Gemini](#)

[ChatGPT](#)

[Claude](#)

[github.com/copilot](#)