

BỘ KHOA HỌC VÀ CÔNG NGHỆ
HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG



Môn: THỰC TẬP CƠ SỞ

Báo cáo tuần

Website quản lý lớp học dành cho giáo viên và học sinh

Giảng viên: Kim Ngọc Bách
Học và tên: Nguyễn Tuấn Anh
Mã sinh viên: B22DCVT026
Lớp: E22CQCN05-B

Hà Nội - 2025

1. GIỚI THIỆU

1.1. Lý do chọn đề tài

Trong kỷ nguyên số hóa, công nghệ thông tin đóng vai trò quan trọng trong việc chuyển đổi các lĩnh vực, đặc biệt là giáo dục. Việc ứng dụng công nghệ vào quản lý lớp học không chỉ giúp tối ưu hóa quy trình dạy và học mà còn tạo điều kiện thuận lợi cho sự tương tác giữa giáo viên và học sinh. Mặc dù nhiều nền tảng quản lý lớp học trực tuyến đã xuất hiện, phần lớn vẫn còn những hạn chế như giao diện phức tạp, thiếu tính năng cần thiết hoặc không đủ thân thiện với người dùng. Dự án "Website Quản lý Lớp học" được đề xuất nhằm giải quyết các vấn đề trên bằng cách cung cấp một nền tảng trực tuyến toàn diện, dễ sử dụng và hiệu quả.

Dự án này không chỉ giải quyết nhu cầu thực tiễn mà còn mang ý nghĩa học thuật, giúp sinh viên rèn luyện và nâng cao kỹ năng lập trình, thiết kế hệ thống, quản lý dự án.

1.2. Mục tiêu của dự án

Dự án nhằm xây dựng một website quản lý lớp học trực tuyến với các mục tiêu chính sau:

- **Xây dựng nền tảng quản lý lớp học trực tuyến:** Phát triển website với chức năng đăng ký, đăng nhập, tạo và tham gia lớp học.
- **Hỗ trợ quản lý lớp học hiệu quả:** Cung cấp công cụ giao bài tập, nộp bài, theo dõi tiến độ học tập và quản lý danh sách học sinh.
- **Tích hợp lịch và công việc thông minh:** Tạo hệ thống lịch hiển thị thời gian thực và quản lý danh sách công việc.
- **Tùy chỉnh cá nhân hóa:** Cho phép người dùng cập nhật thông tin cá nhân như mật khẩu và ảnh đại diện.
- **Đảm bảo tính mở rộng và ổn định:** Xây dựng hệ thống với kiến trúc linh hoạt, dễ nâng cấp và đảm bảo hoạt động ổn định.
- **Tối ưu trải nghiệm người dùng:** Thiết kế giao diện trực quan, thân thiện, tương thích trên nhiều thiết bị.
- **Đóng góp vào giáo dục:** Tạo công cụ hỗ trợ giảng dạy và học tập, nâng cao chất lượng giáo dục.

1.3. Phạm vi dự án

Dự án tập trung vào các chức năng cốt lõi cho việc quản lý lớp học trực tuyến, bao gồm:

- Quản lý tài khoản người dùng (giáo viên, học sinh).
- Quản lý lớp học (tạo, tham gia, xem thông tin, quản lý thành viên).
- Quản lý bài tập (giao bài, nộp bài, xem bài nộp, chấm điểm sơ bộ).
- Các tiện ích hỗ trợ (lịch, danh sách việc cần làm).
- Cài đặt tài khoản cá nhân.

Các tính năng nâng cao như thảo luận trực tuyến thời gian thực, phân tích dữ liệu học tập chi tiết, hoặc tích hợp sâu với các hệ thống bên thứ ba nằm ngoài phạm vi của dự án này nhưng có thể được xem xét cho các phiên bản phát triển sau.

1.4. Đối tượng sử dụng

Hệ thống hướng đến hai đối tượng người dùng chính:

- **Giáo viên:** Người tạo và quản lý lớp học, giao bài tập, theo dõi tiến độ và tương tác với học sinh.
- **Học sinh:** Người tham gia lớp học, nhận và nộp bài tập, theo dõi lịch học và công việc cá nhân.

1.5. Bố cục báo cáo

Báo cáo này được cấu trúc thành các phần chính sau:

- **Chương 1: Giới thiệu:** Tổng quan về đề tài, mục tiêu, phạm vi và đối tượng sử dụng.
- **Chương 2: Cơ sở lý thuyết và công nghệ sử dụng:** Trình bày các kiến thức nền tảng và công nghệ được áp dụng trong dự án.
- **Chương 3: Phân tích và thiết kế hệ thống:** Chi tiết hóa quá trình phân tích yêu cầu và thiết kế các thành phần của hệ thống.
- **Chương 4: Triển khai chức năng hệ thống:** Mô tả quá trình xây dựng và hiện thực hóa các chức năng của website.
- **Chương 5: Thực hiện:** Bao gồm các hoạt động kiểm thử, triển khai và đánh giá hệ thống.
- **Chương 6: Kết luận và Hướng phát triển:** Tổng kết các kết quả đạt được, những hạn chế, bài học kinh nghiệm và định hướng phát triển trong tương lai.

2. CƠ SỞ LÝ THUYẾT VÀ CÔNG NGHỆ SỬ DỤNG

2.1. Cơ sở lý thuyết

- **Ứng dụng web (Web Application):** Là một ứng dụng phần mềm chạy trên máy chủ web, người dùng tương tác thông qua trình duyệt web. Dự án này xây dựng một ứng dụng web động, cho phép tương tác hai chiều giữa người dùng và hệ thống.
- **Mô hình Client-Server:** Hệ thống hoạt động dựa trên mô hình client-server. Trình duyệt của người dùng (client) gửi yêu cầu đến máy chủ web (server). Server xử lý yêu cầu, tương tác với cơ sở dữ liệu nếu cần, và trả về kết quả (thường là HTML, CSS, JavaScript) cho client hiển thị.
- **Hệ quản trị cơ sở dữ liệu quan hệ (RDBMS):** MySQL được sử dụng làm RDBMS để lưu trữ và quản lý dữ liệu có cấu trúc của hệ thống (thông tin người dùng, lớp học, bài tập, bài nộp).
- **Ngôn ngữ lập trình phía máy chủ (Server-side Scripting):** PHP được sử dụng để xử lý logic nghiệp vụ ở phía server, tương tác với cơ sở dữ liệu, xác thực người dùng và tạo các trang web động.
- **Ngôn ngữ lập trình phía máy khách (Client-side Scripting):** HTML5, CSS3 và JavaScript được sử dụng để xây dựng cấu trúc, giao diện và các tương tác động trên trình duyệt của người dùng. Fetch API trong JavaScript được dùng để giao tiếp bất đồng bộ với backend API.
- **RESTful API (Representational State Transfer Application Programming Interface):** Dự án xây dựng các API theo nguyên tắc RESTful để frontend có thể giao tiếp và trao đổi dữ liệu với backend một cách chuẩn hóa thông qua các phương thức HTTP (GET, POST, PUT, DELETE).
- **Bảo mật web:**
 - **Mã hóa mật khẩu:** Sử dụng thuật toán bcrypt để mã hóa mật khẩu người dùng trước khi lưu vào cơ sở dữ liệu, tăng cường tính bảo mật.
 - **Phòngกัน Cross-Site Scripting (XSS):** Sử dụng các hàm như htmlspecialchars() để xử lý dữ liệu đầu ra, tránh việc mã độc được chèn vào trang web.
 - **Phòngกัน Cross-Site Request Forgery (CSRF):** Sử dụng CSRF token trong các biểu mẫu yêu cầu thay đổi dữ liệu để đảm bảo yêu cầu xuất phát từ người dùng hợp lệ.
 - **Kiểm tra dữ liệu đầu vào:** Xác thực và làm sạch dữ liệu đầu vào từ người dùng để tránh các lỗi và tấn công tiềm ẩn.
 - **Giới hạn đăng nhập thất bại:** Theo dõi số lần đăng nhập thất bại và tạm khóa tài khoản để chống tấn công brute-force.

- **HTTPS:** (Mặc dù không triển khai trực tiếp trong code PHP, việc sử dụng HTTPS trên môi trường production là một yêu cầu bảo mật quan trọng).
- **Responsive Web Design:** Thiết kế giao diện website có khả năng tự động điều chỉnh để hiển thị tối ưu trên các kích thước màn hình khác nhau (máy tính, máy tính bảng, điện thoại di động).
- **SMTP (Simple Mail Transfer Protocol):** Giao thức được sử dụng để gửi email, tích hợp thông qua PHPMailer để gửi email xác nhận và thông báo.

2.2. Công nghệ sử dụng

Dự án đã sử dụng các công nghệ và công cụ sau:

- **Frontend:**
 - **HTML5:** Xây dựng cấu trúc ngữ nghĩa cho các trang web.
 - **CSS3:** Thiết kế giao diện, bố cục và hiệu ứng.
 - **JavaScript (ES6+):** Xử lý các tương tác người dùng, thao tác DOM, và giao tiếp bất đồng bộ với backend thông qua Fetch API.
 - **Bootstrap 5.3.2:** Framework CSS/JS để xây dựng giao diện responsive nhanh chóng và nhất quán. (Suy luận từ includes/footer.php và cấu trúc class CSS)
- **Backend:**
 - **PHP (phiên bản 8.1/8.2):** Ngôn ngữ lập trình chính phía server để xử lý logic, xây dựng API.
 - **MySQL:** Hệ quản trị cơ sở dữ liệu quan hệ để lưu trữ dữ liệu.
- **Gửi Email:**
 - **PHPMailer:** Thư viện PHP để tích hợp chức năng gửi email.
 - **SMTP:** Sử dụng Gmail SMTP cho môi trường local và SMTP của InfinityFree cho môi trường production.
- **Máy chủ:**
 - **XAMPP:** Môi trường phát triển cục bộ (Apache, PHP, MySQL).
 - **InfinityFree:** Dịch vụ hosting miễn phí để triển khai website lên môi trường thực tế.
- **Công cụ hỗ trợ:**
 - **Visual Studio Code (VS Code):** Môi trường phát triển tích hợp (IDE) để viết mã.
 - **GitHub:** Hệ thống quản lý phiên bản mã nguồn.
 - **Postman:** Công cụ để kiểm thử các API endpoint.
 - **Figma:** Công cụ thiết kế giao diện người dùng và tạo prototype.
 - **Balsamiq:** Công cụ thiết kế wireframe.
 - **MySQL Workbench:** Công cụ thiết kế và quản lý cơ sở dữ liệu MySQL, tạo ERD.

- **FileZilla:** Công cụ FTP để tải mã nguồn lên hosting.

3. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

Quá trình phân tích và thiết kế hệ thống được thực hiện trong những tuần đầu của dự án, đặt nền móng cho việc triển khai sau này.

3.1. Phân tích yêu cầu

Việc phân tích yêu cầu được tiến hành kỹ lưỡng để xác định rõ các nhu cầu của người dùng và hệ thống.

- **Yêu cầu chức năng:** Các chức năng chính của hệ thống được xác định như sau:
 - **Quản lý Người dùng:**
 - Đăng ký tài khoản mới (họ tên, email, mật khẩu, vai trò giáo viên/học sinh).
 - Đăng nhập bằng email và mật khẩu, phân quyền dựa trên vai trò.
 - Xác thực email sau khi đăng ký.
 - Khôi phục mật khẩu qua email.
 - Cập nhật thông tin cá nhân (họ tên, ngày sinh, giới tính, ảnh đại diện).
 - Thay đổi mật khẩu.
 - **Quản lý Lớp học:**
 - Giáo viên: Tạo lớp học mới (tên lớp, mô tả, mã lớp tự động tạo), xem danh sách lớp đã tạo, chỉnh sửa thông tin lớp, xóa lớp.
 - Học sinh: Tham gia lớp học bằng mã lớp, xem danh sách các lớp đã tham gia, rời lớp học.
 - Hiện thị thông tin chi tiết lớp học (tên lớp, giáo viên, mã lớp, mô tả).
 - Quản lý thành viên trong lớp (giáo viên xem danh sách học sinh, xóa học sinh khỏi lớp; học sinh xem danh sách thành viên).
 - **Quản lý Bài tập và Bài nộp:**
 - Giáo viên: Tạo bài tập mới (tiêu đề, mô tả, thời hạn, tệp đính kèm), xem danh sách bài tập đã giao, chỉnh sửa bài tập, xóa bài tập. Xem danh sách bài nộp của học sinh cho từng bài tập, chấm điểm và cung cấp phản hồi.
 - Học sinh: Xem danh sách bài tập được giao trong lớp, xem chi tiết bài tập (mô tả, thời hạn, tệp đính kèm từ giáo viên),

nộp bài (nhập văn bản hoặc tải tệp lên), xem lại bài đã nộp, xem điểm và phản hồi từ giáo viên.

- **Tính năng Hỗ trợ:**
 - **Lịch:** Hiện thị lịch theo thời gian thực, tự động đánh dấu các sự kiện quan trọng như thời hạn nộp bài. Trong quá trình triển khai, chức năng này được hiện thực bằng cách nhúng Google Calendar.
 - **Việc cần làm:** Hiện thị danh sách các công việc (bài tập) được giao từ tất cả các lớp học, phân loại theo trạng thái (quá hạn, sắp đến hạn, đã giao).
 - **Thông báo:** Hệ thống gửi email thông báo (xác nhận đăng ký, thông báo nộp bài).
 - **Xem tài liệu lớp học:** Học sinh và giáo viên có thể xem danh sách các tệp tài liệu được đính kèm trong các bài tập của lớp.
- **Yêu cầu phi chức năng:**
 - **Bảo mật:** Mã hóa mật khẩu (bcrypt), bảo vệ chống XSS, CSRF, kiểm tra dữ liệu đầu vào, giới hạn đăng nhập thất bại.
 - **Hiệu suất:** Thời gian phản hồi của hệ thống dưới 2 giây cho các tác vụ thông thường, hỗ trợ khoảng 1000 người dùng đồng thời (mục tiêu ban đầu).
 - **Tính khả dụng:** Giao diện người dùng trực quan, dễ sử dụng, tương thích trên nhiều thiết bị (máy tính, điện thoại, máy tính bảng).
 - **Độ tin cậy:** Hệ thống hoạt động ổn định, xử lý lỗi tốt, mục tiêu thời gian hoạt động 99.9%.
 - **Khả năng mở rộng:** Kiến trúc cho phép dễ dàng nâng cấp và thêm tính năng mới trong tương lai.

Để đảm bảo các yêu cầu này sát với thực tế, em đã tham khảo ý kiến từ bạn bè đóng vai trò giáo viên và học sinh, đặc biệt là về giao diện và các định dạng tệp cần hỗ trợ.

3.2. Thiết kế hệ thống

3.2.1. Sơ đồ Use Case

Sơ đồ Use Case được xây dựng để mô tả các tương tác chính giữa các tác nhân (Actor) và hệ thống. Các tác nhân chính bao gồm:

- **Người dùng (Chung):**
 - Đăng ký
 - Đăng nhập

- Quản lý hồ sơ cá nhân (Xem/Chỉnh sửa thông tin, Đổi mật khẩu, Cập nhật ảnh đại diện)
- Xem lịch
- Xem việc cần làm
- Đăng xuất
- **Giáo viên (Kế thừa từ Người dùng):**
 - Tạo Lớp học
 - Xem Danh sách Lớp học của tôi
 - Xem Chi tiết Lớp học
 - Chỉnh sửa Thông tin Lớp học
 - Xóa Lớp học
 - Mời Học sinh (thông qua chia sẻ mã lớp)
 - Xem Danh sách Thành viên Lớp
 - Xóa Học sinh khỏi Lớp
 - Tạo Bài tập
 - Xem Danh sách Bài tập đã giao
 - Chỉnh sửa Bài tập
 - Xóa Bài tập
 - Xem Danh sách Bài nộp của Học sinh
 - Chấm điểm và Phản hồi Bài nộp
 - Xem Tiến độ Lớp học
 - Xem/Tải tài liệu lớp học
- **Học sinh (Kế thừa từ Người dùng):**
 - Tham gia Lớp học (bằng mã lớp)
 - Xem Danh sách Lớp học đã tham gia
 - Xem Chi tiết Lớp học
 - Rời Lớp học
 - Xem Danh sách Thành viên Lớp
 - Xem Danh sách Bài tập được giao
 - Xem Chi tiết Bài tập
 - Nộp Bài tập
 - Xem lại Bài đã nộp
 - Xem Điểm và Phản hồi
 - Xem/Tải tài liệu lớp học

Sơ đồ Use Case đã được kiểm tra và điều chỉnh để bao quát các tương tác quan trọng, ví dụ như thêm trường hợp "Quên Mật khẩu".

3.2.2. Sơ đồ luồng dữ liệu (DFD)

Sơ đồ luồng dữ liệu (Data Flow Diagram - DFD) được xây dựng để minh họa cách dữ liệu di chuyển trong hệ thống.

- **DFD Cấp 0 (Context Diagram):**
 - Mô tả tổng quan hệ thống như một tiến trình duy nhất.
 - Các thực thể bên ngoài: Giáo viên, Học sinh.
 - Luồng dữ liệu chính: Yêu cầu đăng ký/đăng nhập, Thông tin lớp học, Thông tin bài tập, Bài nộp, Điểm số, Thông báo.
- **DFD Cấp 1:**
 - Phân rã tiến trình chính ở DFD Cấp 0 thành các tiến trình con chi tiết hơn, ví dụ:
 - **Xác thực Người dùng:** Xử lý đăng ký, đăng nhập, xác thực email, quản lý phiên.
 - **Quản lý Lớp học:** Xử lý tạo lớp, tham gia lớp, hiển thị thông tin lớp, quản lý thành viên.
 - **Quản lý Bài tập:** Xử lý tạo bài tập, giao bài, hiển thị bài tập.
 - **Quản lý Bài nộp:** Xử lý nộp bài, lưu trữ bài nộp, hiển thị bài nộp, chấm điểm.
 - **Quản lý Thông báo:** Gửi email thông báo.
 - **Quản lý Lịch & Việc cần làm:** Xử lý dữ liệu cho lịch và danh sách công việc.
 - Các kho dữ liệu (Data Store) tương ứng với các bảng trong cơ sở dữ liệu (Users, Classes, Assignments, Submissions, Enrollments) được thể hiện.
 - Mô tả luồng dữ liệu chi tiết, ví dụ từ khi giáo viên tạo bài tập đến khi học sinh nộp bài và giáo viên chấm điểm.

Các sơ đồ DFD đã được kiểm tra kỹ lưỡng để đảm bảo tính rõ ràng và phù hợp với yêu cầu đã phân tích.

3.2.3. Thiết kế cơ sở dữ liệu

Cơ sở dữ liệu được thiết kế bằng MySQL, với mô hình quan hệ thực thể (ERD) được hoàn thiện bằng MySQL Workbench.

Mô hình Quan hệ Thực thể (ERD) và Mô tả Bảng:

Dựa trên tệp CREATE_TABLE.sql và các báo cáo, các bảng chính bao gồm:

1. **users:** Lưu thông tin người dùng.
 - id INT PRIMARY KEY AUTO_INCREMENT
 - full_name VARCHAR(255)
 - dob DATE (Ngày sinh)
 - gender VARCHAR(50) (Giới tính)
 - email VARCHAR(255) UNIQUE NOT NULL
 - password VARCHAR(255) NOT NULL (được mã hóa bcrypt)

- profile_picture VARCHAR(255) (đường dẫn tới ảnh đại diện, mặc định 'default.png')
 - email_verification_code VARCHAR(255) (mã xác thực email)
 - email_verified_at DATETIME NULL (thời điểm email được xác thực)
 - password_reset_token VARCHAR(255) NULL (token khôi phục mật khẩu)
 - password_reset_expires DATETIME NULL (thời gian hết hạn token)
 - created_at DATETIME DEFAULT CURRENT_TIMESTAMP
2. **classes:** Lưu thông tin lớp học.
- id INT PRIMARY KEY AUTO_INCREMENT
 - class_name VARCHAR(255) NOT NULL
 - class_code VARCHAR(50) UNIQUE NOT NULL (mã lớp để tham gia)
 - teacher_id INT NOT NULL (Khóa ngoại tham chiếu đến users.id)
 - description TEXT NULL (Mô tả lớp học)
 - created_at DATETIME DEFAULT CURRENT_TIMESTAMP
3. **enrollments:** Lưu thông tin học sinh tham gia lớp học (quan hệ nhiều-nhiều giữa users và classes).
- id INT PRIMARY KEY AUTO_INCREMENT
 - user_id INT NOT NULL (Khóa ngoại tham chiếu đến users.id - học sinh)
 - class_id INT NOT NULL (Khóa ngoại tham chiếu đến classes.id)
 - enrollment_date DATETIME DEFAULT CURRENT_TIMESTAMP
 - UNIQUE KEY user_class_unique (user_id, class_id) (Đảm bảo mỗi học sinh chỉ tham gia một lớp một lần)
4. **assignments:** Lưu thông tin bài tập.
- id INT PRIMARY KEY AUTO_INCREMENT
 - class_id INT NOT NULL (Khóa ngoại tham chiếu đến classes.id)
 - title VARCHAR(255) NOT NULL
 - description TEXT NULL
 - due_date DATETIME NOT NULL
 - file_path VARCHAR(255) NULL (đường dẫn tệp đính kèm của bài tập)
 - created_at DATETIME DEFAULT CURRENT_TIMESTAMP
5. **submissions:** Lưu thông tin bài nộp của học sinh.
- id INT PRIMARY KEY AUTO_INCREMENT
 - assignment_id INT NOT NULL (Khóa ngoại tham chiếu đến assignments.id)
 - student_id INT NOT NULL (Khóa ngoại tham chiếu đến users.id)

- submission_text TEXT NULL (nội dung bài nộp dạng văn bản)
- file_path VARCHAR(255) NULL (đường dẫn tệp bài nộp của học sinh)
- submitted_at DATETIME DEFAULT CURRENT_TIMESTAMP
- grade VARCHAR(50) NULL (điểm số, có thể là text để linh hoạt hơn, ví dụ "Đạt/Không đạt", "A+", "8.5/10")
- feedback TEXT NULL (phản hồi từ giáo viên)
- status VARCHAR(50) DEFAULT 'submitted' (trạng thái: submitted, late, graded, pending_grading)
- UNIQUE KEY assignment_student_unique (assignment_id, student_id) (Đảm bảo mỗi học sinh chỉ nộp một bài cho một bài tập)

Ghi chú: Bảng events được định nghĩa trong CREATE_TABLE.sql nhưng không thấy được sử dụng rõ ràng trong logic chính của các báo cáo tuần, có thể là một phần dự định cho tính năng lịch nâng cao hơn. Trong báo cáo này, sẽ tập trung vào các bảng đã được triển khai chức năng.

Các chỉ mục (indexes) đã được thêm vào các cột thường xuyên được truy vấn như email trong bảng users và class_code trong bảng classes để tối ưu hiệu suất. Các mối quan hệ khóa ngoại được thiết lập để đảm bảo tính toàn vẹn dữ liệu. Dữ liệu mẫu cũng được chèn vào để phục vụ quá trình kiểm thử.

3.2.4. Thiết kế giao diện người dùng (UI)

- **Wireframes:**

- Sử dụng công cụ Balsamiq để tạo wireframe cho các trang chính, tập trung vào bố cục và các thành phần chức năng.
- Các trang được thiết kế wireframe bao gồm: Trang chủ, Đăng ký, Đăng nhập, Bảng điều khiển, Trang Lớp học, Lịch, Việc cần làm, và Cài đặt.
- Thiết kế hướng đến sự đơn giản, dễ sử dụng và có tính đáp ứng (responsive).

- **Prototype:**

- Phát triển prototype chi tiết hơn trong Figma dựa trên wireframe đã có.
- Phong cách thiết kế: Hiện đại, thân thiện, lấy cảm hứng từ Material Design với bảng màu xanh-trắng.
- Tương tác: Các yếu tố tương tác như nút bấm, trường nhập liệu, menu điều hướng được mô phỏng.
- Responsive: Thiết kế cho cả phiên bản máy tính và di động sử dụng auto-layout của Figma.

- Ví dụ chi tiết: Trang Lớp học có các tab cho bài tập, học sinh, thông tin; Trang Lịch có hiệu ứng hiển thị chi tiết khi tương tác.
- Tự đánh giá và điều chỉnh prototype, ví dụ thêm thanh tìm kiếm vào trang lớp học.

4. TRIỂN KHAI CHỨC NĂNG HỆ THỐNG

Quá trình triển khai chức năng hệ thống bao gồm việc thiết lập môi trường, xây dựng giao diện người dùng (Frontend), phát triển logic xử lý phía máy chủ (Backend), tích hợp hai phần này và hiện thực hóa các chức năng cụ thể.

4.1. Môi trường phát triển

- **Môi trường cục bộ (Local):**
 - Sử dụng XAMPP để tạo môi trường phát triển cục bộ trên máy tính cá nhân.
 - Phiên bản PHP được sử dụng là PHP 8.1/8.2, cùng với máy chủ web Apache và cơ sở dữ liệu MySQL.
 - Tập php.ini được cấu hình để tăng upload_max_filesize lên 10MB-15MB, bật các extension cần thiết như openssl và mbstring.
- **Môi trường triển khai (Production):**
 - Sử dụng dịch vụ hosting miễn phí InfinityFree.
 - InfinityFree hỗ trợ PHP (ví dụ PHP 8.1) và MySQL (ví dụ MySQL 5.7).

4.2. Phát triển Frontend

Giao diện người dùng được xây dựng bằng HTML5, CSS3 và JavaScript, tập trung vào trải nghiệm người dùng và tính tương thích trên nhiều thiết bị. Quá trình này diễn ra chủ yếu trong Tuần 3 và Tuần 4.

- **Cấu trúc HTML5:** Xây dựng cấu trúc ngữ nghĩa cho các trang, đảm bảo tuân thủ chuẩn W3C.
- **Thiết kế CSS3:**
 - Sử dụng CSS thuần kết hợp với các nguyên tắc từ Bootstrap (dù không thấy @import trực tiếp Bootstrap trong style.css, nhưng các class như container, row, col-md-X, btn, card, modal, navbar cho thấy sự ảnh hưởng hoặc sử dụng một phần của Bootstrap, có thể là thông qua CDN link trong header).
 - Tạo tập assets/css/style.css chung để định nghĩa các biến CSS (ví dụ: --gc-primary, --gc-text-primary) và các lớp tái sử dụng. Các tập CSS cụ thể cho từng trang như landing-page.css, dashboard-

legacy.css, class-view.css, todo-page.css cũng được tạo để quản lý style tốt hơn.

- Áp dụng bảng màu xanh-trắng và phong cách Material Design như trong prototype.
- Thiết kế responsive bằng media queries để điều chỉnh giao diện trên các kích thước màn hình khác nhau (desktop, tablet, mobile).
- **JavaScript (Fetch API):**
 - Sử dụng Fetch API để gửi yêu cầu bất đồng bộ đến các API endpoint của backend.
 - Xử lý dữ liệu JSON trả về từ API để cập nhật nội dung động trên các trang.
 - Thêm các tương tác người dùng như chuyển tab, lọc danh sách, hiển thị thông báo.
 - Sử dụng localStorage để lưu trữ tạm thời một số dữ liệu (token phiên, danh sách lớp học) nhằm cải thiện trải nghiệm và giảm số lần gọi API.
 - Tập assets/js/script.js chứa các mã JavaScript chung, ví dụ như kích hoạt tooltip của Bootstrap và xử lý active class cho navigation.
- **Các trang chính được phát triển:**
 - **Trang chủ (Landing Page - index.php):** Giao diện giới thiệu website, các nút kêu gọi hành động "Đăng ký", "Đăng nhập", và các mục giới thiệu tính năng. Sử dụng AOS (Animate On Scroll) library để tạo hiệu ứng.
 - **Trang Đăng ký (register.php) và Đăng nhập (login.php):** Các biểu mẫu cho phép người dùng tạo tài khoản và đăng nhập.
 - **Trang Bảng điều khiển (dashboard.php):** Giao diện chính sau khi đăng nhập, hiển thị danh sách lớp học (phân theo vai trò giáo viên/học sinh), các bài tập sắp đến hạn. Có sidebar điều hướng.
 - **Trang Xem Lớp học (class_view.php):** Hiển thị chi tiết một lớp học với các tab: Bảng tin & Bài tập, Thành viên, Thông tin lớp, và Tiến độ (cho giáo viên). Cho phép giáo viên tạo bài tập, quản lý thành viên; học sinh xem bài tập và nộp bài.
 - **Trang Việc cần làm (todo.php):** Liệt kê các bài tập và nhiệm vụ cần hoàn thành, với bộ lọc theo lớp học, thời hạn, trạng thái.
 - **Trang Lịch (calendar.php):** Nhúng Google Calendar để hiển thị lịch cá nhân và các sự kiện của lớp học.
 - **Trang Cài đặt (settings.php):** Cho phép người dùng cập nhật thông tin cá nhân (họ tên, ngày sinh, giới tính), thay đổi mật khẩu, và cập nhật ảnh đại diện.
 - **Trang Xem tài liệu lớp học (view_class_documents.php):** Hiển thị danh sách các tệp tài liệu được đính kèm trong các bài tập của một lớp cụ thể.

Kiểm tra giao diện được thực hiện trên các trình duyệt phổ biến (Chrome, Firefox, Edge, Safari) và các thiết bị khác nhau (máy tính, điện thoại, máy tính bảng) để đảm bảo tính tương thích.

4.3. Phát triển Backend

Phần backend được xây dựng bằng PHP thuần, tập trung vào việc tạo các API RESTful để xử lý logic nghiệp vụ và tương tác với cơ sở dữ liệu MySQL. Quá trình này diễn ra chủ yếu trong Tuần 5 và Tuần 6.

- **API RESTful:**

- Thiết kế các endpoint API theo chuẩn REST. Mỗi endpoint chịu trách nhiệm cho một tài nguyên hoặc một hành động cụ thể.
- Sử dụng các phương thức HTTP (POST, GET) để thực hiện các thao tác CRUD (Create, Read, Update, Delete) trên dữ liệu.
- Đầu vào của API thường được nhận qua \$_POST hoặc \$_GET, và đầu ra được trả về dưới dạng JSON.
- Các endpoint chính bao gồm:
 - Đăng ký người dùng: POST /api/register (trong register.php đã xử lý trực tiếp, hoặc có thể tách ra file riêng)
 - Đăng nhập người dùng: POST /api/login (trong login.php đã xử lý trực tiếp)
 - Tạo lớp học: POST /api/classes/create (trong create_class.php đã xử lý trực tiếp)
 - Tham gia lớp học: POST /api/classes/join (trong join_class.php đã xử lý trực tiếp)
 - Lấy danh sách lớp học: GET /api/classes/list (logic trong dashboard.php)
 - Tạo bài tập: POST /api/assignments/create (trong create_assignment.php đã xử lý trực tiếp)
 - Nộp bài tập: POST /api/submissions/submit (trong submit_assignment.php đã xử lý trực tiếp)
 - Lấy danh sách bài nộp: GET /api/submissions/list (logic trong view_submissions.php)
 - Thay đổi mật khẩu: POST /api/settings/change-password (logic trong settings.php)
 - Tải ảnh đại diện: POST /api/settings/upload-avatar (logic trong settings.php)
- Lưu ý: Dựa trên cấu trúc file hiện tại, nhiều logic API được xử lý trực tiếp trong các file PHP chính thay vì có thư mục api/ riêng biệt với các endpoint độc lập như kế hoạch ban đầu trong báo cáo tuần. Tuy nhiên, nguyên tắc xử lý request và response JSON vẫn được áp dụng ở một số nơi.

- **Tương tác Cơ sở dữ liệu:**
 - Sử dụng MySQLi extension của PHP để kết nối và thực hiện các truy vấn SQL đến cơ sở dữ liệu MySQL.
 - Sử dụng Prepared Statements để phòng tránh SQL Injection.
- **Xử lý Logic Nghiệp vụ:**
 - **Xác thực người dùng:** Kiểm tra thông tin đăng nhập, tạo và quản lý phiên làm việc (\$_SESSION).
 - **Phân quyền:** Kiểm tra vai trò người dùng (giáo viên/học sinh) để cho phép hoặc từ chối các hành động nhất định (ví dụ: chỉ giáo viên mới được tạo lớp, giao bài). Các hàm như isTeacherOfClass(), isEnrolledInClass() trong includes/functions.php hỗ trợ việc này.
 - **Quản lý Lớp học:** Logic tạo mã lớp duy nhất, ghi danh học sinh, kiểm tra quyền sở hữu lớp.
 - **Quản lý Bài tập:** Logic giao bài, xử lý tệp đính kèm, kiểm tra thời hạn.
 - **Quản lý Bài nộp:** Logic nộp bài, lưu tệp bài nộp, cập nhật trạng thái (đúng hạn, muộn, đã chấm).
- **Tích hợp PHPMailer:**
 - Sử dụng thư viện PHPMailer để gửi email.
 - Cấu hình SMTP (ban đầu với Gmail, sau đó với SMTP của InfinityFree) để gửi các email như:
 - Email xác thực tài khoản khi đăng ký.
 - Email thông báo cho giáo viên khi có học sinh nộp bài.
 - Email hướng dẫn khôi phục mật khẩu.
- **Bảo mật Backend:**
 - **Mã hóa mật khẩu:** Sử dụng hàm password_hash() với thuật toán bcrypt để lưu trữ mật khẩu an toàn. Hàm password_verify() được dùng để kiểm tra mật khẩu khi đăng nhập.
 - **Kiểm tra dữ liệu đầu vào:** Sử dụng các hàm như trim(), htmlspecialchars(), filter_var() để làm sạch và xác thực dữ liệu nhận từ client.
 - **Xử lý tải tệp an toàn:** Kiểm tra định dạng tệp, kích thước tệp tối đa, tạo tên tệp duy nhất và lưu vào thư mục được chỉ định (uploads/assignment_files/, uploads/submission_files/, uploads/profile_pictures/).
 - **Quản lý phiên (Session Management):** Sử dụng session của PHP để duy trì trạng thái đăng nhập của người dùng. Hàm session_regenerate_id(true) được sử dụng sau khi đăng nhập thành công để tăng cường bảo mật.
 - **CSRF Token:** Có đề cập đến việc thêm CSRF token đơn giản trong các biểu mẫu (Tuần 6). Cần kiểm tra lại việc triển khai thực tế trong code.

Tài liệu API cơ bản được soạn thảo dưới dạng Markdown, mô tả các endpoint, tham số và định dạng phản hồi.

4.4. Tích hợp Frontend và Backend

Quá trình tích hợp frontend và backend được thực hiện chủ yếu trong Tuần 7 và Tuần 8.

- **Gọi API từ Frontend:** Sử dụng Fetch API trong JavaScript để gửi yêu cầu đến các endpoint PHP ở backend.
- **Hiển thị Dữ liệu Động:** Dữ liệu JSON trả về từ API được xử lý bằng JavaScript để cập nhật nội dung các trang web một cách động, thay vì tải lại toàn bộ trang. Ví dụ: danh sách lớp học trên dashboard, danh sách bài tập trong một lớp.
- **Xử lý Form:** Các biểu mẫu HTML (đăng ký, đăng nhập, tạo lớp, tạo bài tập, nộp bài, cài đặt) được xử lý bằng JavaScript để thu thập dữ liệu, gửi đến API tương ứng, và hiển thị thông báo thành công hoặc lỗi dựa trên phản hồi từ backend.
- **Đồng bộ hóa dữ liệu:** Đảm bảo rằng các thay đổi dữ liệu ở backend (ví dụ: tạo lớp mới, nộp bài) được phản ánh chính xác và kịp thời trên giao diện người dùng. Có những lúc cần gọi lại API để làm mới dữ liệu sau một thao tác.

4.5. Các chức năng chính và cách triển khai

Dưới đây là mô tả chi tiết hơn về cách triển khai một số chức năng chính:

1. Đăng ký (register.php)

- Frontend: Form HTML với các trường: họ tên, ngày sinh, giới tính, email, mật khẩu, xác nhận mật khẩu, ảnh đại diện (tùy chọn).
- Backend:
 - Nhận dữ liệu từ POST.
 - Validation: Kiểm tra tính hợp lệ và đầy đủ của dữ liệu (họ tên, email, mật khẩu mạnh, ngày sinh, giới tính). Kiểm tra email có tồn tại chưa.
 - Xử lý tải lên ảnh đại diện: Nếu có, kiểm tra định dạng, kích thước, tạo tên file duy nhất và lưu vào uploads/profile_pictures/. Mặc định là default.png.
 - Mã hóa mật khẩu bằng password_hash().
 - Tạo mã xác thực email (email_verification_code).
 - Lưu thông tin người dùng vào bảng users.
 - Gửi email xác thực chứa mã code bằng PHPMailer.
 - Chuyển hướng đến trang xác thực email (verify_email.php).

2. Xác thực Email (verify_email.php)

- Frontend: Form nhập mã xác thực đã được gửi qua email. Nút "Gửi lại mã".
- Backend:
 - Nhận mã code và email từ POST.
 - Kiểm tra mã code với email_verification_code trong bảng users cho email tương ứng.
 - Nếu khớp: Cập nhật email_verified_at thành thời gian hiện tại, xóa email_verification_code. Thông báo thành công và chuyển hướng đến trang đăng nhập.
 - Nếu không khớp: Thông báo lỗi.
 - Xử lý "Gửi lại mã": Tạo mã mới, cập nhật vào DB, gửi lại email.

3. Đăng nhập (login.php)

- Frontend: Form HTML với email và mật khẩu.
- Backend:
 - Nhận email và mật khẩu từ POST.
 - Truy vấn bảng users để tìm người dùng theo email.
 - Kiểm tra tài khoản đã được xác thực email (email_verified_at IS NOT NULL).
 - Nếu tìm thấy và đã xác thực, so sánh mật khẩu bằng password_verify().
 - Nếu thành công: Lưu user_id, full_name vào \$_SESSION, tái tạo session ID, chuyển hướng đến dashboard.php.
 - Nếu thất bại: Thông báo lỗi. Xử lý giới hạn số lần đăng nhập thất bại.

4. Quên mật khẩu (forgot_password.php) và Đặt lại mật khẩu (reset_password.php)

- forgot_password.php:
 - Frontend: Form nhập email.
 - Backend: Kiểm tra email tồn tại và đã xác thực. Tạo password_reset_token và password_reset_expires (ví dụ 1 giờ). Lưu vào bảng users. Gửi email chứa link dạng reset_password.php?token=TOKEN.
- reset_password.php:
 - Frontend: Form nhập mật khẩu mới và xác nhận mật khẩu mới (hiển thị nếu token hợp lệ).
 - Backend: Kiểm tra token từ URL có hợp lệ và chưa hết hạn không. Nếu hợp lệ, cho phép người dùng đặt mật khẩu mới. Mã hóa mật khẩu mới, cập nhật vào bảng users, xóa password_reset_token.

5. Tạo Lớp học (create_class.php)

- Frontend: Form nhập tên lớp, mô tả.
- Backend (chỉ giáo viên):
 - Nhận thông tin lớp từ POST.
 - Validation: Tên lớp là bắt buộc.
 - Tạo class_code duy nhất (ví dụ: 6 ký tự chữ hoa và số).
 - Lưu thông tin vào bảng classes với teacher_id là \$_SESSION['user_id'].
 - Tự động ghi danh giáo viên (người tạo lớp) vào bảng enrollments cho lớp vừa tạo.
 - Chuyển hướng đến class_view.php?id=NEW_CLASS_ID.

6. Tham gia Lớp học (join_class.php)

- Frontend: Form nhập mã lớp.
- Backend (chỉ học sinh):
 - Nhận class_code từ POST.
 - Tìm class_id từ class_code trong bảng classes.
 - Kiểm tra học sinh không phải là giáo viên của lớp đó.
 - Kiểm tra học sinh chưa tham gia lớp đó (trong enrollments).
 - Nếu hợp lệ: Thêm bản ghi vào enrollments với user_id của học sinh và class_id tìm được.
 - Chuyển hướng đến class_view.php?id=CLASS_ID.

7. Xem Lớp học (class_view.php)

- Backend:
 - Lấy class_id từ \$_GET['id'].
 - Kiểm tra người dùng hiện tại có quyền xem lớp này không (là giáo viên hoặc đã ghi danh).
 - Lấy thông tin chi tiết lớp (tên lớp, tên giáo viên, mã lớp, mô tả) từ bảng classes và users.
 - Tab "Bảng tin & Bài tập": Lấy danh sách bài tập của lớp từ bảng assignments, sắp xếp theo ngày tạo giảm dần. Với mỗi bài tập, nếu người xem là học sinh, kiểm tra xem họ đã nộp bài chưa và trạng thái bài nộp từ bảng submissions. Nếu là giáo viên, hiển thị số lượng bài đã nộp.
 - Tab "Thành viên": Lấy danh sách thành viên (giáo viên và học sinh) từ bảng enrollments và users.
 - Tab "Thông tin lớp": Hiển thị mô tả chi tiết lớp.
 - Tab "Tiến độ" (chỉ giáo viên): Thống kê tỷ lệ hoàn thành bài tập của lớp.
- Frontend:
 - Hiển thị banner lớp học (tên lớp, giáo viên, mã lớp cho GV).
 - Các nút hành động: "Tạo mới" (bài tập/thông báo cho GV), "Xóa lớp" (GV), "Rời lớp" (HS).
 - Hệ thống tab điều hướng.

- Khu vực tạo bài đăng nhanh cho giáo viên.
- Danh sách bài tập dạng card, mỗi card có tiêu đề, người đăng, hạn nộp, mô tả (có thể collapse), tệp đính kèm (nếu có), và các nút hành động (Xem bài nộp/Sửa/Xóa cho GV; Nộp bài/Xem bài của tôi/Sửa bài nộp cho HS).
- Danh sách thành viên với ảnh đại diện, tên, email, vai trò (GV/HS) và nút xóa thành viên (cho GV).
- Thông tin mô tả và nội quy lớp.
- Bảng tiến độ bài tập cho giáo viên.

8. Tạo/Sửa Bài tập (create_assignment.php, edit_assignment.php)

- Frontend: Form nhập tiêu đề, mô tả, thời hạn nộp, tệp đính kèm.
- Backend (chỉ giáo viên):
 - Validation: Tiêu đề, thời hạn là bắt buộc.
 - Xử lý tải lên tệp đính kèm (nếu có) vào uploads/assignment_files/, lưu tên file vào file_path của bảng assignments.
 - Lưu/Cập nhật thông tin bài tập vào bảng assignments.
 - Chuyển hướng về class_view.php.

9. Nộp Bài tập (submit_assignment.php)

- Frontend: Form nhập nội dung văn bản, tải lên tệp bài làm. Hiện thị thông tin bài tập (tiêu đề, hạn nộp). Nếu đã nộp, hiển thị bài đã nộp và cho phép sửa nếu chưa quá hạn và chưa chấm.
- Backend (chỉ học sinh đã ghi danh và không phải GV của lớp):
 - Kiểm tra hạn nộp. Nếu quá hạn, vẫn cho nộp nhưng đánh dấu status là 'late'.
 - Kiểm tra xem đã nộp bài này chưa. Nếu rồi và được phép nộp lại (chưa quá hạn, chưa chấm điểm), thì cho cập nhật.
 - Xử lý tải lên tệp bài làm (nếu có) vào uploads/submission_files/, lưu tên file vào file_path của bảng submissions.
 - Lưu/Cập nhật thông tin bài nộp (assignment_id, student_id, submission_text, file_path, submitted_at, status) vào bảng submissions.
 - Thông báo thành công, chuyển hướng về class_view.php.

10. Xem Bài nộp (view_submissions.php)

- Backend (chỉ giáo viên):
 - Lấy assignment_id từ \$_GET.
 - Lấy danh sách tất cả học sinh trong lớp (trừ giáo viên).
 - Với mỗi học sinh, lấy thông tin bài nộp của họ (nếu có) từ bảng submissions.
 - Sắp xếp danh sách (ví dụ: ưu tiên bài chưa chấm).
- Frontend:

- Thống kê số lượng đã nộp, chưa nộp.
- Sử dụng accordion để hiển thị bài nộp của từng học sinh. Mỗi item accordion bao gồm: tên HS, ảnh đại diện, trạng thái nộp, thời gian nộp, nội dung text, file đính kèm.
- Form cho giáo viên nhập điểm (grade) và phản hồi (feedback) cho từng bài nộp. Khi submit form này, cập nhật vào bảng submissions và đổi status thành 'graded'.

11. Xóa Bài tập (delete_assignment.php) và Xóa Lớp học (delete_class.php)

- Backend (chỉ giáo viên sở hữu):
 - Kiểm tra quyền.
 - Xóa delete_assignment.php: Xóa các tệp bài nộp liên quan, xóa các bản ghi trong submissions, xóa tệp đính kèm của bài tập, rồi xóa bản ghi trong assignments.
 - Xóa delete_class.php: Xóa tất cả các bài nộp, sau đó xóa tất cả bài tập (bao gồm cả file), sau đó xóa tất cả ghi danh, cuối cùng xóa lớp học.
 - Sử dụng transaction để đảm bảo tính toàn vẹn.

12. Lịch (calendar.php)

- Hiện tại: Nhúng Google Calendar bằng iframe. Người dùng được hướng dẫn mở Google Calendar để quản lý sự kiện.
- Kế hoạch ban đầu: Tích hợp API để lấy dữ liệu bài tập và hiển thị lên lịch tùy chỉnh.

13. Việc cần làm (todo.php)

- Backend:
 - Với vai trò học sinh: Lấy danh sách các bài tập từ các lớp đã tham gia mà chưa nộp.
 - Với vai trò giáo viên: Lấy danh sách các bài tập có bài nộp đang chờ chấm (status 'submitted' hoặc 'late').
- Frontend:
 - Hiện thị danh sách công việc được phân loại (quá hạn, đến hạn hôm nay, trong tuần, cần chấm...).
 - Mỗi công việc hiển thị tiêu đề, tên lớp, hạn nộp, và link đến trang nộp bài hoặc trang chấm bài.

14. Cài đặt (settings.php)

- Backend:
 - Cập nhật thông tin cá nhân (họ tên, ngày sinh, giới tính) vào bảng users.
 - Thay đổi mật khẩu: Xác thực mật khẩu cũ, mã hóa mật khẩu mới, cập nhật vào bảng users.

- Cập nhật ảnh đại diện: Xử lý tải tệp ảnh mới lên uploads/profile_pictures/, xóa ảnh cũ (nếu không phải default), cập nhật profile_picture trong bảng users.
- Frontend: Ba form riêng biệt cho thông tin cá nhân, đổi mật khẩu, và tải ảnh đại diện.

5. THỰC HIỆN

Phần này mô tả quá trình kiểm thử, triển khai và đánh giá hệ thống sau khi các chức năng đã được phát triển.

5.1. Kiểm thử hệ thống

Quá trình kiểm thử được thực hiện trong Tuần 9 và Tuần 10, bao gồm kiểm thử đơn vị, kiểm thử tích hợp và kiểm thử giao diện.

- **Kiểm thử Đơn vị (Unit Testing):**
 - **Mục tiêu:** Kiểm tra từng chức năng riêng lẻ để đảm bảo chúng hoạt động đúng như mong đợi.
 - **Phương pháp:**
 - Backend: Sử dụng Postman để gửi các yêu cầu đến từng API endpoint với các kịch bản dữ liệu hợp lệ và không hợp lệ.
 - Frontend: Kiểm tra thủ công các thành phần giao diện, xử lý sự kiện JavaScript.
 - **Các chức năng được kiểm thử đơn vị:**
 - **Đăng ký:** Dữ liệu hợp lệ, email trùng lặp, mật khẩu ngắn, định dạng email không hợp lệ. Phát hiện lỗi không kiểm tra định dạng email, đã sửa bằng filter_var().
 - **Đăng nhập:** Thông tin đúng/sai, đăng nhập nhiều lần thất bại (kiểm tra giới hạn 5 lần và khóa 30 phút). Phát hiện lỗi giới hạn đăng nhập không reset, đã sửa.
 - **Tạo Lớp học:** Mã lớp duy nhất/trùng lặp, tên lớp rỗng. Phát hiện lỗi không kiểm tra độ dài mã lớp, đã sửa.
 - **Giao Bài và Nộp Bài:** Dữ liệu hợp lệ, tệp hợp lệ/không hợp lệ (kích thước, định dạng). Phát hiện lỗi quyền ghi thư mục uploads/, đã sửa.
 - **Lịch, Việc cần làm, Cài đặt:** Hiện thị thời hạn, lọc công việc, đổi mật khẩu, tải ảnh đại diện. Phát hiện lỗi ảnh đại diện lớn không hiển thị, đã sửa bằng cách thêm bước nén ảnh (hoặc giới hạn kích thước rõ ràng hơn khi upload).
 - **Kết quả:** 5 lỗi nhỏ được phát hiện và sửa chữa trong tuần 9.
- **Kiểm thử Tích hợp (Integration Testing):**

- **Mục tiêu:** Đảm bảo các chức năng hoạt động đồng bộ và dữ liệu được truyền tải chính xác giữa frontend và backend, cũng như giữa các module khác nhau.
- **Phương pháp:** Mô phỏng các quy trình người dùng hoàn chỉnh.
- **Các kịch bản chính:**
 - Quy trình Giáo viên: Đăng ký -> Đăng nhập -> Tạo lớp -> Giao bài -> Xem bài nộp -> Chấm điểm.
 - Quy trình Học sinh: Đăng ký -> Đăng nhập -> Tham gia lớp -> Xem bài tập -> Nộp bài -> Kiểm tra trên Lịch và Việc cần làm.
 - Quy trình Cài đặt: Đổi mật khẩu -> Đăng nhập lại; Tải ảnh đại diện -> Kiểm tra hiển thị.
- **Kết quả:**
 - Hầu hết các quy trình hoạt động mượt mà, dữ liệu đồng bộ.
 - Phát hiện lỗi danh sách "Việc cần làm" không tự cập nhật sau khi học sinh nộp bài, đã sửa bằng cách gọi lại API.
 - Phát hiện lỗi thông báo nộp bài không gửi đến giáo viên nếu email giáo viên không hợp lệ (trong DB), đã thêm kiểm tra.
- **Kiểm thử Giao diện (UI Testing):**
 - **Mục tiêu:** Đảm bảo giao diện hiển thị đúng và hoạt động tốt trên các trình duyệt và thiết bị khác nhau.
 - **Thiết bị và Trình duyệt:**
 - Máy tính: Chrome, Firefox, Edge, Safari (mượn máy).
 - Di động: Chrome trên Android (mô phỏng và thiết bị thật), Safari trên iPhone (mượn thiết bị).
 - **Kịch bản:** Kiểm tra bố cục responsive ở các độ phân giải khác nhau (320px, 768px, 1024px, 1920px), kiểm tra hiển thị và tương tác trên tất cả các trang.
 - **Kết quả:**
 - Giao diện responsive tốt trên hầu hết các trình duyệt.
 - Phát hiện lỗi thanh điều hướng sidebar bị lệch trên Edge, đã sửa bằng CSS box-sizing: border-box.
 - Phát hiện lỗi Lịch tháng hiển thị sai trên Safari (iPhone) do table-layout, đã sửa.
- **Báo cáo Kiểm thử:**
 - Một tệp test-report.md được tạo để ghi lại các lỗi, mô tả, kịch bản và trạng thái sửa chữa. Tổng cộng 8 lỗi đã được phát hiện và sửa chữa qua hai tuần kiểm thử.

5.2. Triển khai hệ thống

Quá trình triển khai website lên môi trường thực tế được thực hiện trong Tuần 11.

- **Chọn Hosting:** Dịch vụ hosting miễn phí InfinityFree được chọn để triển khai.
- **Các bước triển khai:**
 1. **Chuẩn bị môi trường hosting:** Đăng ký tài khoản InfinityFree, nhận thông tin FTP và MySQL. Kiểm tra cấu hình PHP, MySQL trên hosting.
 2. **Tải mã nguồn:** Sử dụng FileZilla để tải toàn bộ mã nguồn (HTML, CSS, JS, PHP) lên thư mục htdocs trên server. Cấu hình quyền ghi cho thư mục uploads/.
 3. **Thiết lập Cơ sở dữ liệu:** Tạo cơ sở dữ liệu trên InfinityFree. Xuất CSDL từ XAMPP và nhập (import) vào MySQL trên hosting thông qua phpMyAdmin. Cập nhật thông tin kết nối CSDL trong tệp includes/db.php (\$db_host, \$db_user, \$db_pass, \$db_name).
 4. **Cấu hình Email:** Tích hợp lại PHPMailer với thông tin SMTP được cung cấp bởi InfinityFree.
- **Kiểm tra hoạt động trên Hosting:**
 - Truy cập website qua URL do InfinityFree cung cấp.
 - Kiểm tra tất cả các trang và chức năng chính (đăng ký, đăng nhập, tạo lớp, giao bài, nộp bài) để đảm bảo hoạt động ổn định.
 - Sử dụng Postman để kiểm tra các API endpoint trên môi trường hosting.
 - Phát hiện và sửa lỗi gửi email do cấu hình SMTP sai trên InfinityFree.

5.3. Đánh giá và Thu thập phản hồi

Công việc này được thực hiện trong Tuần 12.

- **Kiểm tra ổn định liên tục:** Truy cập và sử dụng website hàng ngày trên hosting để đảm bảo không có lỗi mới phát sinh.
- **Tối ưu nhỏ:** Nén thêm các tệp JavaScript để cải thiện tốc độ tải trang.
- **Thu thập phản hồi từ người dùng thử nghiệm:** Mời 3 bạn bè (đóng vai giáo viên và học sinh) sử dụng website và đưa ra nhận xét.
 - **Phản hồi tích cực:** Giao diện thân thiện, dễ sử dụng; chức năng lịch và việc cần làm rõ ràng; thông báo email hoạt động tốt.
 - **Góp ý cải thiện:** Thanh điều hướng trên di động hơi nhỏ; cần thêm hướng dẫn sử dụng cho người mới.
 - **Điều chỉnh dựa trên phản hồi:** Tăng kích thước thanh điều hướng trên di động, thêm trang "Hướng dẫn" cơ bản.

6. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

6.1. Kết quả đạt được

Sau 12 tuần thực hiện, dự án "Website quản lý lớp học dành cho giáo viên và học sinh" đã đạt được các kết quả chính sau:

- **Hoàn thành hệ thống với đầy đủ các chức năng cốt lõi:**
 - Quản lý người dùng: Đăng ký, đăng nhập, xác thực email, quản lý hồ sơ, thay đổi mật khẩu, cập nhật ảnh đại diện.
 - Quản lý lớp học: Tạo lớp (GV), tham gia lớp (HS), xem danh sách lớp, xem chi tiết lớp, quản lý thành viên, rời lớp, xóa lớp (GV).
 - Quản lý bài tập: Giao bài (GV), xem danh sách bài tập, xem chi tiết bài tập, nộp bài (HS), xem bài nộp (GV), chấm điểm và phản hồi (GV).
 - Tiện ích hỗ trợ: Lịch (nhúng Google Calendar), trang Việc cần làm (liệt kê bài tập cần làm/cần chấm).
- **Website hoạt động ổn định trên hosting InfinityFree:** Hệ thống đã được triển khai thành công và có thể truy cập trực tuyến.
- **Giao diện responsive:** Website tương thích tốt trên các thiết bị khác nhau như máy tính, máy tính bảng và điện thoại di động, hỗ trợ các trình duyệt phổ biến.
- **Đảm bảo các yếu tố bảo mật cơ bản:** Mã hóa mật khẩu (bcrypt), kiểm tra đầu vào, giới hạn đăng nhập thất bại, giới hạn tải tệp.
- **Hiệu suất được tối ưu ở mức cơ bản:** Thời gian phản hồi API và tốc độ tải trang được cải thiện thông qua các biện pháp như tối ưu truy vấn, cache đơn giản và nén tài nguyên.
- **Hoàn thành tài liệu dự án:** Bao gồm đề cương, báo cáo tuần, và báo cáo tổng kết này.

Nhìn chung, dự án đã đáp ứng được các mục tiêu đề ra trong đề cương ban đầu, tạo ra một công cụ hữu ích cho việc quản lý dạy và học trực tuyến.

6.2. Hạn chế của dự án

Bên cạnh những kết quả đạt được, dự án vẫn còn một số hạn chế:

- **Làm việc cá nhân:** Việc thực hiện dự án một mình dẫn đến một số khó khăn trong việc quản lý thời gian, bao quát hết các khía cạnh và kiểm thử toàn diện trên nhiều môi trường khác nhau.
- **Hosting miễn phí:** Việc sử dụng InfinityFree có một số giới hạn về tốc độ, băng thông và tài nguyên, có thể ảnh hưởng đến trải nghiệm người dùng khi có lượng truy cập lớn.

- **Chức năng Lịch và Thông báo:** Chức năng Lịch hiện tại mới chỉ dừng ở việc nhúng Google Calendar thay vì tích hợp sâu API để hiển thị sự kiện từ cơ sở dữ liệu của hệ thống. Thông báo vẫn chủ yếu dựa vào email, chưa có thông báo real-time trong ứng dụng.
- **Thiếu Framework Backend:** Việc sử dụng PHP thuần cho backend, mặc dù giúp kiểm soát tốt hơn trong phạm vi dự án cá nhân, nhưng cũng làm tăng thời gian phát triển và khó khăn hơn trong việc quản lý, bảo trì và mở rộng API so với việc sử dụng một framework như Laravel.
- **Kiểm thử tự động:** Chưa áp dụng các công cụ kiểm thử tự động (ví dụ: PHPUnit cho backend, Jest/Selenium cho frontend), việc kiểm thử chủ yếu là thủ công.
- **Giao diện người dùng:** Mặc dù đã cố gắng tối ưu, một số phần giao diện có thể cần cải thiện thêm để trực quan và thân thiện hơn nữa, đặc biệt là phần hướng dẫn cho người dùng mới.

6.3. Bài học kinh nghiệm

Qua quá trình thực hiện dự án, em đã rút ra được nhiều bài học quý báu:

- **Tầm quan trọng của việc lập kế hoạch chi tiết:** Việc phân chia công việc theo tuần và đặt ra mục tiêu rõ ràng cho từng giai đoạn giúp quản lý tiến độ hiệu quả hơn, đặc biệt khi làm việc cá nhân.
- **Phân tích yêu cầu kỹ lưỡng là nền tảng:** Dành thời gian phân tích và làm rõ yêu cầu ngay từ đầu giúp giảm thiểu sai sót và chỉnh sửa lớn ở các giai đoạn sau.
- **Thiết kế trước khi code:** Việc tạo wireframe và prototype giúp hình dung rõ hơn về sản phẩm, luồng người dùng và phát hiện các vấn đề về UX/UI sớm.
- **Tái sử dụng code và component:** Xây dựng các hàm và thành phần giao diện có thể tái sử dụng giúp tiết kiệm thời gian và đảm bảo tính nhất quán.
- **Kiểm thử thường xuyên:** Việc kiểm thử liên tục sau mỗi chức năng nhỏ giúp phát hiện và sửa lỗi sớm, tránh tích tụ lỗi lớn.
- **Quản lý phiên bản (Git/Github):** Sử dụng Git giúp theo dõi thay đổi, quản lý mã nguồn hiệu quả và dễ dàng quay lại các phiên bản trước nếu cần.
- **Tầm quan trọng của bảo mật:** Ngay từ khi thiết kế và triển khai, cần chú trọng đến các khía cạnh bảo mật cơ bản để bảo vệ dữ liệu người dùng.
- **Kỹ năng giải quyết vấn đề:** Gặp lỗi trong quá trình cấu hình (SMTP, hosting) hay lập trình là điều không tránh khỏi. Kỹ năng tìm kiếm thông tin, đọc tài liệu và thử nghiệm các giải pháp khác nhau là rất quan trọng.

- **Học hỏi công nghệ mới:** Dự án là cơ hội để tìm hiểu và áp dụng các công nghệ, thư viện mới (ví dụ: PHPMailer, Fetch API).

6.4. Hướng phát triển trong tương lai

Để nâng cao và hoàn thiện hơn nữa website quản lý lớp học, một số hướng phát triển có thể được xem xét trong tương lai:

- **Nâng cấp hạ tầng:**
 - Chuyển sang sử dụng dịch vụ hosting trả phí (ví dụ: Hostinger, DigitalOcean) để cải thiện tốc độ, độ ổn định và băng thông.
- **Cải tiến Backend:**
 - Xem xét chuyển đổi sang sử dụng một PHP framework như Laravel hoặc Symfony để quản lý API, bảo mật và cấu trúc dự án tốt hơn.
 - Tích hợp hệ thống hàng đợi (Queue) cho các tác vụ chạy nền như gửi email hàng loạt.
- **Mở rộng chức năng:**
 - **Thông báo thời gian thực:** Sử dụng WebSocket hoặc các dịch vụ như Pusher để cung cấp thông báo real-time trong ứng dụng (ví dụ: khi có bài tập mới, có người nộp bài, nhận phản hồi).
 - **Thảo luận trực tuyến:** Thêm diễn đàn hoặc chatroom cho từng lớp học để tăng tương tác giữa giáo viên và học sinh.
 - **Lịch tùy chỉnh nâng cao:** Xây dựng module lịch riêng, không phụ thuộc vào Google Calendar, cho phép tạo sự kiện tùy chỉnh, nhắc nhở.
 - **Phân tích và báo cáo học tập:** Cung cấp các công cụ thống kê, biểu đồ trực quan về tiến độ học tập của học sinh và hiệu quả giảng dạy của giáo viên.
 - **Ngân hàng câu hỏi và tạo bài kiểm tra trắc nghiệm:** Cho phép giáo viên tạo các bài kiểm tra trắc nghiệm online và tự động chấm điểm.
 - **Hỗ trợ đa ngôn ngữ:** Mở rộng hỗ trợ nhiều ngôn ngữ hơn cho giao diện.
- **Cải thiện Trải nghiệm Người dùng (UX) và Giao diện Người dùng (UI):**
 - Thiết kế lại một số giao diện dựa trên phản hồi chi tiết hơn từ người dùng.
 - Xây dựng hệ thống hướng dẫn sử dụng (tutorial) chi tiết hơn cho người dùng mới.
- **Tăng cường Bảo mật:**
 - Thực hiện đánh giá bảo mật (security audit) chuyên sâu.
 - Tích hợp Two-Factor Authentication (2FA).

- **Kiểm thử và Mở rộng:**

- Áp dụng các quy trình kiểm thử tự động (Automated Testing).
- Mở rộng kiểm thử với số lượng lớn người dùng thực tế để đánh giá khả năng chịu tải và mở rộng của hệ thống.

Dự án "Website quản lý lớp học" đã đặt một nền tảng vững chắc. Với những cải tiến và mở rộng trong tương lai, hệ thống hứa hẹn sẽ trở thành một công cụ hỗ trợ đắc lực và toàn diện hơn nữa cho hoạt động giáo dục.