

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ ĐỒNG NAI
KHOA CÔNG NGHỆ THÔNG TIN



KIỂM TRA CUỐI KỲ
BÀI TẬP LỚN

TÊN ĐỀ TÀI: Task Manager

Tên học phần: Lập Trình Mobile

Mã học phần: 0370091

Số tín chỉ: 03

Học kỳ: 06 Năm học: 2025 - 2026

Chuyên ngành: Công nghệ thông tin

Khóa: 18 (2022 - 2026)

Sinh viên thực hiện:

Lương Dương Thành Đạt - 1822040268 – 22DTH2

Giảng viên: Lê Nhật Tùng

ĐỒNG NAI – 2025

Mục Lục

Tài liệu Ứng dụng Task Manager trên Flutter	4
I. Giới thiệu chung.....	4
1. Tên đề tài	4
2. Mục đích của đề tài.....	4
3. Phạm vi của dự án	4
4. Lợi ích của ứng dụng.....	5
II. Phân tích yêu cầu	5
1. Yêu cầu chức năng	5
2. Yêu cầu phi chức năng	6
3. Đối tượng người dùng.....	7
4. Giả định và ràng buộc	8
III. Chi tiết các model	8
1. Model User.....	8
2. Model Task.....	9
IV. Chi tiết API	10
1. UserAPIService	10
2. TaskAPIService	10
V. Chi tiết View.....	11
1. TaskLoginScreen	11
2. TaskRegisterScreen.....	12
3. TaskListScreen	12
4. TaskForm	13
5. TaskAdd	13
6. TaskEdit	13
7. TaskDetail	13
8. TaskListItem.....	14
9. UserProfileScreen.....	14
10. UserEditScreen.dart	14
VI. Cài đặt API bằng XAMPP.....	15

1. Yêu cầu môi trường	15
2. Các bước cài đặt.....	15
3. Lưu ý khi cài đặt	16
VII. Tổng quát.....	17
VIII. Hướng dẫn sử dụng.....	17
1. Đăng ký và đăng nhập	17
2. Quản lý công việc	18
3. Tìm kiếm và phân loại công việc	19
4. Xem chi tiết công việc	19
5. Quản lý tài khoản.....	19
IX. Kế hoạch kiểm thử	20
1. Kiểm thử chức năng.....	20
2. Kiểm thử giao diện người dùng.....	20
3. Kiểm thử tích hợp	21
4. Kiểm thử hiệu suất.....	21
5. Kiểm thử khả năng sử dụng.....	22
X. Bảo mật hệ thống	22
1. Bảo mật dữ liệu người dùng.....	22
2. Bảo mật API giả lập	22
3. Bảo mật ứng dụng.....	22
4. Hạn chế của API giả lập	23
XI. Hướng phát triển tiếp theo	23
1. Chuyển sang API thực tế.....	23
2. Thêm thông báo đẩy	23
3. Hỗ trợ lưu trữ cục bộ.....	23
4. Phân tích và báo cáo	23
5. Tăng cường bảo mật	24
XII. Kết luận	24

Tài liệu Ứng dụng Task Manager trên Flutter

I. Giới thiệu chung

1. Tên đề tài

Quản lý công việc với **Task Manager Application** trên nền tảng Android sử dụng Flutter và API giả lập trên XAMPP.

2. Mục đích của đề tài

Ứng dụng Task Manager được phát triển nhằm cung cấp một công cụ quản lý công việc hiệu quả, cho phép người dùng tổ chức, theo dõi, và hoàn thành các nhiệm vụ cá nhân hoặc nhóm trên thiết bị Android. Mục tiêu chính bao gồm:

- Tạo ra một ứng dụng di động với giao diện thân thiện, dễ sử dụng để quản lý công việc, bao gồm các chức năng như tạo, cập nhật, xóa, và theo dõi công việc.
- Tích hợp với API giả lập chạy trên XAMPP để mô phỏng việc lưu trữ và truy xuất dữ liệu, thay vì sử dụng cơ sở dữ liệu thực tế như MySQL.
- Hỗ trợ các tính năng nâng cao như phân loại công việc theo trạng thái, mức độ ưu tiên, danh mục, và tìm kiếm công việc theo từ khóa.
- Đáp ứng nhu cầu của cá nhân hoặc nhóm làm việc nhỏ, với thiết kế tối ưu cho trải nghiệm người dùng trên các thiết bị Android.

Ứng dụng này không chỉ giúp người dùng quản lý công việc một cách khoa học mà còn tạo nền tảng để thử nghiệm và phát triển các tính năng phức tạp hơn trong tương lai, chẳng hạn như tích hợp với API thực tế hoặc thêm các chức năng như thông báo đẩy.

3. Phạm vi của dự án

- **Nền tảng triển khai:** Ứng dụng di động trên Android, được xây dựng bằng Flutter.
- **Công nghệ sử dụng:**
 - **Frontend:** Flutter (Dart) để phát triển giao diện di động mượt mà, trực quan.
 - **Backend:** API giả lập sử dụng PHP chạy trên XAMPP, mô phỏng các thao tác tạo, đọc, cập nhật, và xóa dữ liệu thông qua dữ liệu JSON tĩnh hoặc logic PHP đơn giản.

- **API:** RESTful API giả lập, trả về dữ liệu JSON để tương tác với ứng dụng Flutter.
- **Đối tượng sử dụng:** Người dùng cá nhân, nhóm làm việc nhỏ, quản trị viên cần quản lý công việc, và nhà phát triển muốn kiểm thử ứng dụng với API giả lập.
- **Tính năng chính:** Quản lý thông tin người dùng, quản lý công việc, phân công công việc, phân loại công việc, tìm kiếm công việc, và tích hợp với API giả lập.

4. Lợi ích của ứng dụng

- **Tính di động:** Người dùng có thể truy cập và quản lý công việc mọi lúc, mọi nơi thông qua điện thoại Android.
- **Giao diện thân thiện:** Thiết kế giao diện hiện đại, dễ sử dụng, phù hợp với cả người dùng không chuyên về công nghệ.
- **Triển khai đơn giản:** Sử dụng API giả lập trên XAMPP giúp giảm độ phức tạp trong quá trình phát triển và kiểm thử, không yêu cầu cài đặt cơ sở dữ liệu.
- **Khả năng mở rộng:** Ứng dụng được thiết kế theo cấu trúc mô-đun, dễ dàng nâng cấp sang API thực tế hoặc tích hợp các tính năng bổ sung như lưu trữ cục bộ hoặc thông báo đẩy.
- **Hiệu quả tổ chức:** Giúp người dùng sắp xếp công việc khoa học, ưu tiên nhiệm vụ quan trọng, và theo dõi tiến độ một cách rõ ràng.

II. Phân tích yêu cầu

1. Yêu cầu chức năng

Ứng dụng Task Manager cần đáp ứng các chức năng chính sau để đảm bảo đáp ứng nhu cầu quản lý công việc:

- **Quản lý người dùng:**
 - Đăng ký tài khoản mới với các thông tin cơ bản như tên đăng nhập, mật khẩu, và địa chỉ email.
 - Đăng nhập vào ứng dụng để truy cập các chức năng, với sự phân biệt giữa quyền của quản trị viên và người dùng thông thường.
 - Cập nhật thông tin cá nhân, bao gồm email, ảnh đại diện, hoặc thay đổi mật khẩu.

- Xóa tài khoản người dùng, chỉ dành cho quản trị viên để quản lý hệ thống.
- **Quản lý công việc:**
 - Tạo công việc mới với các thông tin như tiêu đề, mô tả chi tiết, trạng thái (mở, đang thực hiện, đang xem xét, hoàn thành), mức độ ưu tiên (cao, trung bình, thấp), hạn chót, danh mục, và tệp đính kèm.
 - Chỉnh sửa thông tin công việc hiện có, bao gồm thay đổi trạng thái, ưu tiên, hoặc người được giao.
 - Xóa công việc không còn cần thiết.
 - Phân công công việc cho một người dùng khác trong hệ thống.
 - Xem chi tiết công việc, bao gồm thông tin về người tạo, người được giao, thời gian tạo, thời gian cập nhật, và danh sách tệp đính kèm.
 - Tìm kiếm công việc bằng từ khóa hoặc lọc công việc theo trạng thái, mức độ ưu tiên, hoặc danh mục.
- **Tích hợp API giả lập:**
 - Gửi yêu cầu đến các endpoint API giả lập để mô phỏng việc tạo, đọc, cập nhật, và xóa dữ liệu người dùng và công việc.
 - Nhận dữ liệu JSON từ API giả lập để hiển thị trên các màn hình của ứng dụng, chẳng hạn như danh sách công việc hoặc thông tin người dùng.
- **Tìm kiếm và phân loại:**
 - Cung cấp ô tìm kiếm để người dùng nhập từ khóa và tìm công việc liên quan.
 - Hỗ trợ bộ lọc để sắp xếp công việc theo các tiêu chí như trạng thái, ưu tiên, hoặc danh mục, giúp người dùng dễ dàng quản lý công việc.

2. Yêu cầu phi chức năng

Để đảm bảo ứng dụng hoạt động hiệu quả và đáp ứng kỳ vọng của người dùng, các yêu cầu phi chức năng sau được đặt ra:

- **Bảo mật:**

- Mật khẩu người dùng được mô phỏng mã hóa trong API giả lập để đảm bảo an toàn dữ liệu.
- API giả lập kiểm tra tính hợp lệ của các yêu cầu (ví dụ: xác thực tên đăng nhập và mật khẩu) để ngăn chặn truy cập không mong muốn.
- **Giao diện người dùng:**
 - Giao diện phải tương thích với nhiều kích thước màn hình Android, từ điện thoại nhỏ (4 inch) đến tablet (10 inch).
 - Hỗ trợ chế độ sáng và tối, tự động điều chỉnh theo cài đặt thiết bị.
 - Ngôn ngữ chính là tiếng Việt, với các nhãn và hướng dẫn rõ ràng, dễ hiểu.
- **Dễ triển khai:**
 - API giả lập trên XAMPP dễ dàng cài đặt trên máy tính cá nhân, không yêu cầu cấu hình cơ sở dữ liệu phức tạp.
 - Ứng dụng Flutter có thể chạy trên trình giả lập Android hoặc thiết bị thực mà không cần cấu hình bổ sung.
- **Khả năng mở rộng:**
 - Ứng dụng được thiết kế để dễ dàng chuyển đổi từ API giả lập sang API thực tế với cơ sở dữ liệu như MySQL hoặc Firebase.
 - Cấu trúc mô-đun cho phép tích hợp các tính năng mới, chẳng hạn như thông báo đẩy hoặc lưu trữ cục bộ.

3. Đối tượng người dùng

Ứng dụng Task Manager phục vụ các nhóm đối tượng sau:

- **Cá nhân hoặc nhân viên:** trong nhóm cần tổ chức và theo dõi công việc cá nhân hoặc công việc được giao. Họ sử dụng ứng dụng để tạo công việc, cập nhật trạng thái, và tìm kiếm nhiệm vụ.
- **Người quản lý nhóm:** có quyền tạo tài khoản người dùng, phân công công việc, và xem danh sách người dùng hoặc công việc trong hệ thống.
- **Nhà phát triển:** Những người muốn kiểm thử ứng dụng với API giả lập hoặc phát triển thêm các tính năng mới, chẳng hạn như tích hợp với hệ thống bên ngoài.

4. Giả định và ràng buộc

- **Giả định:**

- Người dùng có thiết bị Android chạy API 21 trở lên và đã cài đặt ứng dụng từ file APK hoặc qua quá trình phát triển.
- API giả lập trên XAMPP chạy trên mạng cục bộ (localhost) và trả về dữ liệu JSON mẫu ổn định.
- Người dùng có kiến thức cơ bản về cách sử dụng ứng dụng di động, chẳng hạn như nhập dữ liệu hoặc nhấn nút.

- **Ràng buộc:**

- Ứng dụng chỉ chạy trên Android, không hỗ trợ iOS hoặc các nền tảng khác trong giai đoạn này.
- API giả lập không lưu trữ dữ liệu lâu dài, chỉ trả về dữ liệu tĩnh hoặc xử lý tạm thời trong phiên làm việc.
- Không hỗ trợ các tính năng phức tạp như lưu trữ cục bộ hoặc thông báo đẩy trong phiên bản hiện tại.

III. Chi tiết các model

1. Model User

Model User định nghĩa thông tin của một người dùng trong hệ thống, được sử dụng để xác thực, phân quyền, và quản lý tài khoản. Các thuộc tính bao gồm:

- **id:** Mã định danh duy nhất, dùng để phân biệt mỗi người dùng trong hệ thống.
- **username:** Tên đăng nhập, yêu cầu duy nhất, dài từ 6 đến 20 ký tự, giúp người dùng đăng nhập vào ứng dụng.
- **password:** Mật khẩu, được mô phỏng mã hóa trong API giả lập, yêu cầu tối thiểu 8 ký tự để đảm bảo an toàn.
- **email:** Địa chỉ email hợp lệ, duy nhất trong hệ thống, dùng để xác minh hoặc liên lạc với người dùng.
- **avatar:** Đường dẫn đến ảnh đại diện, có thể để trống nếu người dùng không cung cấp.
- **createdAt:** Thời gian tạo tài khoản, định dạng ngày giờ, giúp theo dõi lịch sử tài khoản.

- **lastActive:** Thời gian hoạt động gần nhất, dùng để kiểm tra mức độ tương tác của người dùng.
- **role:** Vai trò của người dùng, xác định quyền truy cập, với hai giá trị là quản trị viên (admin) hoặc người dùng thông thường (user).

Model này được API giả lập trả về dưới dạng JSON, mô phỏng thông tin người dùng để ứng dụng Flutter hiển thị hoặc xử lý.

2. Model Task

Model MyTask định nghĩa thông tin của một công việc trong hệ thống, bao gồm tất cả các chi tiết cần thiết để quản lý và theo dõi. Các thuộc tính bao gồm:

- **id:** Mã định danh duy nhất, dùng để phân biệt mỗi công việc.
- **title:** Tiêu đề công việc, ngắn gọn, tối đa 100 ký tự, mô tả nội dung chính của nhiệm vụ.
- **description:** Mô tả chi tiết, tối đa 1000 ký tự, cung cấp thông tin bổ sung về công việc.
- **status:** Trạng thái hiện tại của công việc, bao gồm các giá trị: mở (open), đang thực hiện (in progress), đang xem xét (review), hoặc hoàn thành (completed).
- **priority:** Mức độ ưu tiên, với các giá trị cao (1), trung bình (2), hoặc thấp (3), giúp người dùng tập trung vào nhiệm vụ quan trọng.
- **dueDate:** Hạn chót của công việc, định dạng ngày giờ, có thể để trống nếu không có thời hạn cụ thể.
- **createdAt:** Thời gian tạo công việc, định dạng ngày giờ.
- **updatedAt:** Thời gian cập nhật gần nhất, giúp theo dõi lịch sử thay đổi.
- **createdBy:** Mã định danh của người tạo công việc, liên kết với model User.
- **assignedTo:** Mã định danh của người được giao công việc, có thể để trống nếu chưa phân công.
- **category:** Danh mục công việc, ví dụ: công việc (Work), cá nhân (Personal), hoặc học tập (Study), giúp phân loại nhiệm vụ.
- **attachments:** Danh sách đường dẫn đến các tệp đính kèm, như hình ảnh hoặc tài liệu, có thể để trống.

- **completed:** Trạng thái hoàn thành, xác định công việc đã xong (true) hay chưa (false).

Model này được API giả lập trả về dưới dạng JSON, mô phỏng thông tin công việc để ứng dụng Flutter hiển thị trên các màn hình như danh sách công việc hoặc chi tiết công việc.

IV. Chi tiết API

API giả lập được triển khai bằng PHP trên XAMPP, sử dụng các file như tasks.php và users.php để mô phỏng các thao tác xử lý dữ liệu mà không cần cơ sở dữ liệu. API trả về dữ liệu JSON tĩnh hoặc được xử lý bởi logic PHP đơn giản, đáp ứng các yêu cầu từ ứng dụng Flutter. Các nhóm API chính bao gồm:

1. UserAPIService

API này mô phỏng việc quản lý thông tin người dùng, với các chức năng sau:

- **Tạo người dùng mới:** Nhận thông tin như tên đăng nhập, mật khẩu, email, và vai trò, trả về dữ liệu JSON mô phỏng việc tạo tài khoản thành công.
- **Lấy danh sách người dùng:** Trả về danh sách người dùng mẫu, chỉ dành cho quản trị viên, mô phỏng việc truy xuất thông tin tất cả người dùng.
- **Lấy thông tin người dùng:** Truy xuất chi tiết một người dùng dựa trên mã định danh, trả về dữ liệu JSON chứa thông tin như tên đăng nhập, email, và vai trò.
- **Đăng nhập:** Xác thực người dùng bằng tên đăng nhập và mật khẩu, trả về dữ liệu JSON mô phỏng thông tin xác thực thành công hoặc lỗi nếu thông tin không hợp lệ.
- **Cập nhật thông tin người dùng:** Mô phỏng việc thay đổi thông tin như email, ảnh đại diện, hoặc mật khẩu, trả về dữ liệu JSON xác nhận cập nhật.
- **Xóa người dùng:** Mô phỏng việc xóa tài khoản dựa trên mã định danh, chỉ dành cho quản trị viên, trả về dữ liệu JSON xác nhận xóa thành công.

2. TaskAPIService

API này mô phỏng việc quản lý công việc, với các chức năng sau:

- **Tạo công việc mới:** Nhận thông tin công việc như tiêu đề, mô tả, trạng thái, mức độ ưu tiên, hạn chót, và người được giao, trả về dữ liệu JSON mô phỏng công việc mới.

- **Lấy danh sách công việc:** Trả về danh sách công việc mẫu, mô phỏng việc truy xuất tất cả công việc trong hệ thống.
- **Cập nhật công việc:** Nhận thông tin cập nhật như trạng thái, ưu tiên, hoặc hạn chót, trả về dữ liệu JSON mô phỏng công việc đã được chỉnh sửa.
- **Xóa công việc:** Mô phỏng việc xóa công việc dựa trên mã định danh, trả về dữ liệu JSON xác nhận xóa thành công.
- **Lấy công việc theo người dùng:** Trả về danh sách công việc liên quan đến một người dùng cụ thể, mô phỏng việc lọc công việc theo mã định danh người dùng.
- **Tìm kiếm công việc:** Nhận từ khóa hoặc bộ lọc (trạng thái, ưu tiên, danh mục), trả về danh sách công việc phù hợp, mô phỏng chức năng tìm kiếm.

API giả lập được thiết kế để hoạt động trên môi trường cục bộ (localhost), trả về dữ liệu JSON cố định hoặc được xử lý bởi PHP, đảm bảo dễ dàng tích hợp với ứng dụng Flutter và kiểm thử trong quá trình phát triển.

V. Chi tiết View

Ứng dụng Task Manager sử dụng Flutter để xây dựng các màn hình giao diện, được thiết kế theo chuẩn Material Design để mang lại trải nghiệm người dùng mượt mà, trực quan trên các thiết bị Android. Các màn hình chính bao gồm:

1. TaskLoginScreen

Màn hình đăng nhập cho phép người dùng nhập thông tin để truy cập ứng dụng. Các thành phần giao diện bao gồm:

- Logo ứng dụng ở đầu trang, sử dụng màu sắc và thiết kế tối giản để tạo nhận diện thương hiệu.
- Trường nhập tên đăng nhập, hiển thị nhãn rõ ràng và hỗ trợ kiểm tra định dạng hợp lệ.
- Trường nhập mật khẩu, với tùy chọn hiển thị hoặc ẩn mật khẩu để tăng tính tiện lợi.
- Nút đăng nhập, được thiết kế nổi bật với màu sắc tương phản, dễ nhận biết.
- Thông báo lỗi, hiển thị dưới dạng văn bản màu đỏ nếu người dùng nhập sai tên đăng nhập hoặc mật khẩu.

- Liên kết chuyển sang màn hình đăng ký, đặt ở cuối màn hình để hỗ trợ người dùng mới.

Giao diện được bố trí theo chiều dọc, với các thành phần căn giữa để phù hợp với các kích thước màn hình khác nhau.

2. TaskRegisterScreen

Màn hình đăng ký cho phép người dùng tạo tài khoản mới. Các thành phần giao diện bao gồm:

- Các trường nhập thông tin, bao gồm tên đăng nhập, email, mật khẩu, và xác nhận mật khẩu, mỗi trường có nhãn hướng dẫn rõ ràng.
- Nút đăng ký, nổi bật và đặt ở vị trí dễ nhấn, kích hoạt kiểm tra tính hợp lệ của thông tin (ví dụ: mật khẩu khớp, email hợp lệ).
- Thông báo lỗi, hiển thị khi thông tin nhập vào không hợp lệ, chẳng hạn như mật khẩu không khớp hoặc email đã tồn tại.
- Liên kết quay lại màn hình đăng nhập, hỗ trợ người dùng đã có tài khoản.

Giao diện được thiết kế tối giản, với các trường nhập được sắp xếp gọn gàng và khoảng cách hợp lý để đảm bảo dễ sử dụng.

3. TaskListScreen

Màn hình danh sách công việc hiển thị tất cả công việc của người dùng, được tổ chức dưới dạng danh sách cuộn hoặc thẻ. Các tính năng chính bao gồm:

- Mỗi công việc hiển thị các thông tin chính như tiêu đề, trạng thái, mức độ ưu tiên, và hạn chót, được sắp xếp theo thứ tự thời gian hoặc ưu tiên.
- Ô tìm kiếm ở đầu màn hình, cho phép nhập từ khóa để tìm công việc nhanh chóng.
- Bộ lọc, hiển thị dưới dạng nút hoặc menu, hỗ trợ sắp xếp công việc theo trạng thái (mở, đang thực hiện, hoàn thành), mức độ ưu tiên (cao, trung bình, thấp), hoặc danh mục (công việc, cá nhân).
- Nút tạo công việc mới, thường là nút nổi hình dấu cộng ở góc dưới bên phải, dẫn đến màn hình tạo công việc.
- Hành động chạm, cho phép người dùng nhấn vào một công việc để xem chi tiết hoặc chỉnh sửa.

Giao diện sử dụng các thẻ hoặc hàng có viền mềm mại, với màu sắc khác nhau để biểu thị trạng thái hoặc ưu tiên, giúp người dùng dễ dàng nhận biết công việc quan trọng.

4. TaskFormScreen

Biểu mẫu chung để nhập thông tin công việc, được sử dụng cho cả tạo mới và chỉnh sửa công việc. Các trường bao gồm:

- **Tiêu đề công việc:** Trường nhập văn bản bắt buộc, giới hạn 100 ký tự, với nhãn hướng dẫn rõ ràng.
- **Mô tả chi tiết:** Trường nhập văn bản dài, hỗ trợ nhập tối đa 1000 ký tự, cho phép mô tả chi tiết nhiệm vụ.
- **Trạng thái:** Menu thả xuống với các tùy chọn mở, đang thực hiện, đang xem xét, hoặc hoàn thành.
- **Mức độ ưu tiên:** Menu thả xuống với các giá trị cao, trung bình, hoặc thấp, được đánh dấu bằng màu sắc để dễ nhận biết.
- **Hạn chót:** Trường chọn ngày giờ từ lịch tích hợp, có thể để trống nếu không cần thời hạn.
- **Danh mục:** Menu thả xuống hoặc trường nhập tùy chỉnh, hỗ trợ các giá trị như công việc, cá nhân, hoặc học tập.
- **Tệp đính kèm:** Tùy chọn chọn tệp từ thiết bị, hiển thị danh sách tệp đã chọn với khả năng xem trước.

Biểu mẫu được thiết kế với các trường được sắp xếp theo thứ tự logic, có nhãn hướng dẫn và thông báo lỗi nếu nhập sai định dạng.

5. TaskAddScreen

Màn hình tạo công việc mới, sử dụng biểu mẫu TaskForm. Giao diện được tối ưu để hướng dẫn người dùng nhập đầy đủ thông tin, với nút lưu ở vị trí dễ nhấn và nút hủy để quay lại màn hình danh sách.

6. TaskEditScreen

Màn hình chỉnh sửa công việc, sử dụng biểu mẫu TaskForm với các trường được điền sẵn từ thông tin công việc hiện tại. Người dùng có thể thay đổi bất kỳ trường nào, chẳng hạn như trạng thái, ưu tiên, hoặc hạn chót, và lưu lại hoặc hủy bỏ thay đổi.

7. TaskDetailScreen

Màn hình chi tiết công việc, hiển thị đầy đủ thông tin của một công việc, bao gồm:

- Tiêu đề công việc, được làm nổi bật ở đầu màn hình.
- Mô tả chi tiết, hiển thị dưới dạng văn bản có thể cuộn nếu nội dung dài.

- Trạng thái và mức độ ưu tiên, được biểu thị bằng màu sắc hoặc biểu tượng.
- Hạn chót, hiển thị dưới dạng ngày giờ rõ ràng.
- Thông tin người tạo và người được giao, bao gồm tên đăng nhập hoặc email.
- Danh mục công việc, giúp người dùng hiểu ngữ cảnh nhiệm vụ.
- Danh sách tệp đính kèm, với tùy chọn xem trước hoặc tải xuống tệp.
- Nút chỉnh sửa, dẫn đến màn hình TaskEdit.
- Nút xóa, hiển thị cảnh báo xác nhận trước khi xóa công việc.

Giao diện được bố trí rõ ràng, với các phần thông tin được phân chia bằng tiêu đề phụ hoặc đường phân cách để dễ theo dõi.

8. TaskListItem

Thành phần giao diện hiển thị từng công việc trong danh sách trên TaskListScreen. Mỗi mục bao gồm:

- Biểu tượng trạng thái, sử dụng màu sắc khác nhau để biểu thị trạng thái (xanh cho hoàn thành, đỏ cho ưu tiên cao, vàng cho đang thực hiện).
- Tiêu đề công việc, được làm nổi bật bằng phông chữ đậm.
- Thông tin phụ, như hạn chót hoặc danh mục, hiển thị dưới dạng văn bản nhỏ hơn.
- Nút hành động, chẳng hạn như biểu tượng chỉnh sửa hoặc xóa, đặt ở bên phải mục.

Mỗi mục được thiết kế dưới dạng thẻ với viền mềm mại, có hiệu ứng nhấn để tăng tính tương tác.

9. UserProfileScreen

- Chức năng: Hiển thị thông tin cá nhân người dùng.
- Dữ liệu hiển thị: Avatar, tên đăng nhập, email, ngày tạo.
- Tính năng
- Tải dữ liệu từ SharedPreferences.
- Nút chuyển sang màn hình đổi mật khẩu (UserEditScreen).

10. UserEditScreen.dart

- Chức năng: Cho phép người dùng đổi mật khẩu.
- Kiểm tra đầu vào:

- Mật khẩu cũ phải đúng.
- Mật khẩu mới ≥ 6 ký tự.
- Xác nhận mật khẩu phải khớp.
- Tính năng:
- Gọi API cập nhật mật khẩu.
- Hiện thị thông báo lỗi/thành công.
- Chuyển về UserProfileScreen sau khi đổi thành công

VI. Cài đặt API bằng XAMPP

1. Yêu cầu môi trường

Để triển khai API giả lập và ứng dụng Flutter, cần đáp ứng các yêu cầu sau:

- **Máy tính phát triển:** Windows, macOS, hoặc Linux với XAMPP (phiên bản 7.4 trở lên) và Flutter SDK (phiên bản 3.0 trở lên).
- **Thiết bị kiểm thử:** Điện thoại Android (API 21 trở lên) hoặc trình giả lập Android trong Android Studio.
- **Môi trường mạng:** API giả lập chạy trên mạng cục bộ (localhost), yêu cầu máy tính và thiết bị kiểm thử kết nối cùng mạng Wi-Fi.

2. Các bước cài đặt

Để thiết lập API giả lập trên XAMPP, thực hiện các bước sau:

+ Tải và cài đặt XAMPP:

- Truy cập trang web chính thức của XAMPP và tải phiên bản phù hợp với hệ điều hành (Windows, macOS, hoặc Linux).
- Cài đặt XAMPP vào thư mục mặc định, ví dụ: C:\xampp trên Windows.
- Đảm bảo cài đặt bao gồm Apache và PHP, không cần MySQL vì API giả lập không sử dụng cơ sở dữ liệu.

+ Khởi động Apache:

- Mở XAMPP Control Panel, nhấn nút "Start" cho Apache để kích hoạt máy chủ web.
- Kiểm tra xem cổng 80 (mặc định của Apache) không bị chiếm dụng bởi ứng dụng khác.

- Truy cập <http://localhost> trên trình duyệt để xác nhận Apache đang chạy.

+ Cấu hình API giả lập:

- Copy các file `tasks.php` và `users.php` vào thư mục `htdocs` của XAMPP, ví dụ: `C:\xampp\htdocs`.
- Các file này chứa logic PHP để trả về dữ liệu JSON tĩnh hoặc xử lý yêu cầu mà không cần cơ sở dữ liệu.
- Đảm bảo các file được cấu hình để trả về dữ liệu JSON đúng định dạng, chẳng hạn như danh sách công việc hoặc thông tin người dùng.

+ Kiểm tra API giả lập:

- Truy cập các endpoint API qua trình duyệt, ví dụ: <http://localhost/tasks.php> hoặc <http://localhost/users.php>.
- Kiểm tra phản hồi JSON, đảm bảo dữ liệu trả về chứa thông tin mẫu như danh sách công việc hoặc người dùng.
- Sử dụng công cụ như Postman để gửi yêu cầu POST hoặc GET đến API và kiểm tra tính đúng đắn của phản hồi.

+ Kết nối ứng dụng Flutter với API:

- Cấu hình ứng dụng Flutter để gửi yêu cầu HTTP đến các endpoint API giả lập, sử dụng địa chỉ <http://localhost> trên máy tính hoặc <http://10.0.2.2> trên trình giả lập Android.
- Kiểm tra việc hiển thị dữ liệu từ API trên các màn hình của ứng dụng, chẳng hạn như danh sách công việc trên `TaskListScreen` hoặc thông tin người dùng trên `TaskDetail`.

3. Lưu ý khi cài đặt

- **Mạng cục bộ:** Nếu kiểm thử trên thiết bị Android thực, đảm bảo thiết bị và máy tính chạy XAMPP kết nối cùng mạng Wi-Fi. Sử dụng địa chỉ IP của máy tính (ví dụ: <http://192.168.1.100/tasks.php>) thay vì `localhost`.
- **Cấu hình file PHP:** Đảm bảo các file `tasks.php` và `users.php` được cấu hình đúng để trả về dữ liệu JSON mẫu, tránh lỗi cú pháp hoặc định dạng.
- **Tương thích Flutter:** Ứng dụng Flutter cần được cấu hình để xử lý các phản hồi JSON từ API giả lập, bao gồm kiểm tra lỗi và định dạng dữ liệu.

VII. Tổng quát

Ứng dụng Task Manager tuân theo mô hình **MVC (Model-View-Controller)**, được điều chỉnh để phù hợp với Flutter và API giả lập trên XAMPP:

- **Model:** Đại diện cho dữ liệu trong hệ thống, bao gồm model User (thông tin người dùng) và model MyTask (thông tin công việc). Model xử lý dữ liệu JSON nhận được từ API giả lập.
- **View:** Các màn hình giao diện của ứng dụng, được xây dựng bằng các widget Flutter, chẳng hạn như TaskLoginScreen, TaskListScreen, TaskDetail. View hiển thị dữ liệu từ model và cung cấp các nút hoặc trường nhập để người dùng tương tác.
- **Controller:** Quản lý trạng thái của ứng dụng và xử lý các yêu cầu từ View. Controller gửi yêu cầu HTTP đến API giả lập, nhận dữ liệu JSON, và cập nhật model để View hiển thị.

Luồng dữ liệu trong ứng dụng:

1. Người dùng tương tác với giao diện, chẳng hạn như nhấn nút đăng nhập, tạo công việc, hoặc tìm kiếm.
2. View gửi yêu cầu đến Controller thông qua các sự kiện, chẳng hạn như nhấn nút hoặc nhập dữ liệu vào ô tìm kiếm.
3. Controller gửi yêu cầu HTTP (GET, POST, PUT, DELETE) đến các endpoint API giả lập trên XAMPP, ví dụ: `http://localhost/tasks.php`.
4. API giả lập xử lý yêu cầu và trả về dữ liệu JSON tĩnh hoặc được tạo bởi logic PHP đơn giản.
5. Controller nhận dữ liệu JSON, cập nhật model, và thông báo cho View để hiển thị kết quả, chẳng hạn như danh sách công việc hoặc thông tin chi tiết công việc.

VIII. Hướng dẫn sử dụng

Ứng dụng Task Manager được thiết kế để dễ sử dụng, với các hướng dẫn rõ ràng cho người dùng trên các màn hình chính. Dưới đây là cách sử dụng ứng dụng:

1. Đăng ký và đăng nhập

- **Đăng ký tài khoản:**
 - Mở ứng dụng và truy cập màn hình đăng ký (TaskRegisterScreen).

- Nhập thông tin bắt buộc, bao gồm tên đăng nhập (6-20 ký tự), email hợp lệ, mật khẩu (tối thiểu 8 ký tự), và xác nhận mật khẩu.
 - Nhấn nút đăng ký để gửi thông tin đến API giả lập.
 - Nếu đăng ký thành công, người dùng được chuyển đến màn hình đăng nhập.
 - Nếu có lỗi (ví dụ: email đã tồn tại hoặc mật khẩu không khớp), thông báo lỗi sẽ hiển thị để hướng dẫn người dùng chỉnh sửa.
- **Đăng nhập:**
 - Truy cập màn hình đăng nhập (TaskLoginScreen).
 - Nhập tên đăng nhập và mật khẩu đã đăng ký.
 - Nhấn nút đăng nhập để xác thực thông qua API giả lập.
 - Nếu thông tin đúng, người dùng được chuyển đến màn hình danh sách công việc.
 - Nếu thông tin sai, thông báo lỗi hiển thị, yêu cầu nhập lại.

2. Quản lý công việc

- **Xem danh sách công việc:**
 - Sau khi đăng nhập, người dùng thấy màn hình danh sách công việc (TaskListScreen), hiển thị tất cả công việc của mình.
 - Mỗi công việc hiển thị tiêu đề, trạng thái, mức độ ưu tiên, và hạn chót, được sắp xếp theo thứ tự thời gian hoặc ưu tiên.
 - Người dùng có thể cuộn danh sách để xem tất cả công việc hoặc sử dụng ô tìm kiếm và bộ lọc để thu hẹp kết quả.
- **Tạo công việc mới:**
 - Nhấn nút tạo mới (nút nổi hình dấu cộng) để mở màn hình tạo công việc (TaskAdd).
 - Điền thông tin vào biểu mẫu, bao gồm tiêu đề, mô tả, trạng thái, ưu tiên, hạn chót, danh mục, và tệp đính kèm (nếu có).
 - Nhấn nút lưu để gửi thông tin đến API giả lập và thêm công việc vào danh sách.
- **Chỉnh sửa công việc:**

- Nhấn vào một công việc trong danh sách để xem chi tiết (TaskDetail).
- Nhấn nút chỉnh sửa để mở màn hình chỉnh sửa (TaskEdit), nơi thông tin công việc được điền sẵn.
- Thay đổi các trường cần thiết và nhấn nút lưu để cập nhật thông tin qua API giả lập.
- **Xóa công việc:**
 - Từ màn hình chi tiết công việc, nhấn nút xóa.
 - Xác nhận hành động xóa trong cửa sổ bật lên để gửi yêu cầu đến API giả lập.
 - Công việc sẽ bị xóa khỏi danh sách và không còn hiển thị.

3. Tìm kiếm và phân loại công việc

- **Tìm kiếm:**
 - Trên màn hình danh sách công việc, nhập từ khóa vào ô tìm kiếm ở đầu màn hình.
 - Kết quả tìm kiếm hiển thị các công việc có tiêu đề hoặc mô tả khớp với từ khóa.
- **Phân loại:**
 - Sử dụng bộ lọc để sắp xếp công việc theo trạng thái (mở, đang thực hiện, hoàn thành), mức độ ưu tiên (cao, trung bình, thấp), hoặc danh mục (công việc, cá nhân).
 - Bộ lọc được truy cập qua nút hoặc menu, với giao diện trực quan để người dùng chọn tiêu chí.

4. Xem chi tiết công việc

- Nhấn vào một công việc trong danh sách để mở màn hình chi tiết (TaskDetail).
- Xem thông tin đầy đủ, bao gồm tiêu đề, mô tả, trạng thái, ưu tiên, hạn chót, người tạo, người được giao, danh mục, và tệp đính kèm.
- Sử dụng các nút chỉnh sửa hoặc xóa để thực hiện hành động tiếp theo.

5. Quản lý tài khoản

- **Cập nhật thông tin:**

- Truy cập màn hình thông tin tài khoản
- Thay đổi mật khẩu và lưu thông tin qua API giả lập.

IX. Kế hoạch kiểm thử

Để đảm bảo ứng dụng Task Manager hoạt động đúng và đáp ứng yêu cầu, kế hoạch kiểm thử được thiết kế với các loại kiểm thử sau:

1. Kiểm thử chức năng

- **Mục tiêu:** Đảm bảo tất cả các chức năng của ứng dụng hoạt động đúng như thiết kế.
- **Kịch bản kiểm thử:**
 - Đăng ký tài khoản với thông tin hợp lệ và không hợp lệ để kiểm tra xử lý lỗi.
 - Đăng nhập với thông tin đúng và sai để kiểm tra xác thực và thông báo lỗi.
 - Tạo công việc mới với đầy đủ thông tin và kiểm tra xem công việc có xuất hiện trong danh sách không.
 - Chỉnh sửa công việc, thay đổi trạng thái và ưu tiên, kiểm tra cập nhật được phản ánh.
 - Xóa công việc và xác nhận công việc không còn trong danh sách.
 - Tìm kiếm công việc bằng từ khóa và kiểm tra kết quả trả về có khớp không.
 - Lọc công việc theo trạng thái, ưu tiên, hoặc danh mục, kiểm tra danh sách hiển thị đúng.

2. Kiểm thử giao diện người dùng

- **Mục tiêu:** Đảm bảo giao diện hiển thị đúng, nhất quán, và dễ sử dụng trên các thiết bị Android.
- **Kịch bản kiểm thử:**
 - Kiểm tra bố cục giao diện trên điện thoại nhỏ (4 inch) và tablet (10 inch) để đảm bảo responsive.
 - Chuyển đổi giữa chế độ sáng và tối, kiểm tra màu sắc và độ tương phản.

- Kiểm tra các nút, trường nhập, và menu có hiển thị đúng và phản hồi khi nhấn.
- Kiểm tra thông báo lỗi hiển thị rõ ràng và ở vị trí phù hợp khi nhập thông tin không hợp lệ.
- Kiểm tra hiệu ứng chuyển động (nhấn nút, cuộn danh sách) để đảm bảo mượt mà.

3. Kiểm thử tích hợp

- **Mục tiêu:** Đảm bảo ứng dụng tương tác đúng với API giả lập và xử lý dữ liệu JSON một cách chính xác.
- **Kịch bản kiểm thử:**
 - Gửi yêu cầu tạo người dùng đến API giả lập, kiểm tra dữ liệu JSON trả về có chứa thông tin đúng.
 - Gửi yêu cầu lấy danh sách công việc, kiểm tra danh sách hiển thị trên TaskListScreen khớp với dữ liệu JSON.
 - Cập nhật công việc qua API giả lập, kiểm tra dữ liệu JSON phản ánh thay đổi và giao diện cập nhật đúng.
 - Xóa công việc, kiểm tra dữ liệu JSON xác nhận xóa và công việc không còn hiển thị.
 - Tìm kiếm công việc, kiểm tra dữ liệu JSON trả về khớp với từ khóa hoặc bộ lọc.

4. Kiểm thử hiệu suất

- **Mục tiêu:** Đảm bảo ứng dụng chạy mượt mà và phản hồi nhanh với dữ liệu mẫu lớn.
- **Kịch bản kiểm thử:**
 - Tải danh sách 100 công việc từ API giả lập, kiểm tra thời gian hiển thị và cuộn danh sách.
 - Gửi liên tiếp 50 yêu cầu HTTP đến API giả lập, kiểm tra độ ổn định và thời gian phản hồi.
 - Kiểm tra ứng dụng trên thiết bị Android cấu hình thấp (API 21), đảm bảo không bị treo hoặc chậm.

5. Kiểm thử khả năng sử dụng

- **Mục tiêu:** Đảm bảo ứng dụng dễ sử dụng và phù hợp với người dùng không chuyên.
- **Kịch bản kiểm thử:**
 - Yêu cầu người dùng thử nghiệm tạo tài khoản, đăng nhập, và tạo công việc, ghi nhận phản hồi về độ dễ sử dụng.
 - Kiểm tra các nhãn, hướng dẫn, và thông báo lỗi có rõ ràng và dễ hiểu không.
 - Đánh giá bố cục giao diện, đảm bảo các nút và trường nhập được đặt ở vị trí trực quan.

X. Bảo mật hệ thống

Mặc dù sử dụng API giả lập, ứng dụng Task Manager vẫn áp dụng các biện pháp bảo mật cơ bản để đảm bảo an toàn dữ liệu và trải nghiệm người dùng:

1. Bảo mật dữ liệu người dùng

- Mật khẩu người dùng được mô phỏng mã hóa trong API giả lập, đảm bảo không lưu trữ dưới dạng văn bản thô.
- Thông tin nhạy cảm như email hoặc tên đăng nhập được xử lý cẩn thận trong dữ liệu JSON trả về, tránh tiết lộ thông tin không cần thiết.
- API giả lập kiểm tra tính hợp lệ của tên đăng nhập và mật khẩu khi đăng nhập, mô phỏng xác thực an toàn.

2. Bảo mật API giả lập

- API giả lập chỉ cho phép truy cập từ mạng cục bộ (localhost hoặc IP nội bộ), giảm nguy cơ truy cập từ bên ngoài.
- Các yêu cầu HTTP đến API được kiểm tra để đảm bảo định dạng đúng, tránh lỗi hoặc yêu cầu không hợp lệ.
- Dữ liệu JSON trả về được cấu trúc rõ ràng, không chứa thông tin nhạy cảm ngoài những gì cần thiết cho ứng dụng.

3. Bảo mật ứng dụng

- Ứng dụng yêu cầu đăng nhập lại sau một khoảng thời gian không hoạt động, đảm bảo an toàn nếu thiết bị bị truy cập trái phép.
- Dữ liệu tạm thời (như danh sách công việc hoặc thông tin người dùng) được xóa khỏi bộ nhớ khi người dùng đăng xuất.

- Các trường nhập như mật khẩu hoặc email được kiểm tra định dạng để ngăn chặn nhập dữ liệu không hợp lệ.

4. Hạn chế của API giả lập

- Vì không sử dụng cơ sở dữ liệu, API giả lập không lưu trữ dữ liệu lâu dài, dẫn đến việc dữ liệu có thể bị mất khi khởi động lại XAMPP.
- Các biện pháp bảo mật chỉ ở mức cơ bản, phù hợp cho môi trường phát triển và kiểm thử, không đủ cho triển khai thực tế.

XI. Hướng phát triển tiếp theo

Ứng dụng Task Manager hiện tại sử dụng API giả lập để đơn giản hóa quá trình phát triển, nhưng có tiềm năng mở rộng với các tính năng và cải tiến sau:

1. Chuyển sang API thực tế

- Thay thế API giả lập bằng API thực tế sử dụng cơ sở dữ liệu như MySQL hoặc Firebase để lưu trữ dữ liệu người dùng và công việc lâu dài.
- Tích hợp các cơ chế xác thực mạnh mẽ như JSON Web Token (JWT) hoặc OAuth để bảo vệ API.
- Triển khai các endpoint API thực tế với khả năng xử lý dữ liệu phức tạp, chẳng hạn như phân quyền chi tiết hoặc báo cáo công việc.

2. Thêm thông báo đẩy

- Tích hợp dịch vụ thông báo đẩy, chẳng hạn như Firebase Cloud Messaging, để gửi cảnh báo khi công việc đến hạn, được giao mới, hoặc có cập nhật trạng thái.
- Cho phép người dùng bật/tắt thông báo trong cài đặt ứng dụng, tùy chỉnh tần suất và nội dung thông báo.

3. Hỗ trợ lưu trữ cục bộ

- Thêm khả năng lưu trữ công việc và thông tin người dùng trên thiết bị Android, cho phép sử dụng ứng dụng khi không có kết nối mạng.
- Đồng bộ dữ liệu cục bộ với API khi kết nối mạng được khôi phục, đảm bảo không mất dữ liệu.

4. Phân tích và báo cáo

- Thêm tính năng xuất báo cáo tiến độ công việc, hiển thị thống kê về số lượng công việc hoàn thành, đang thực hiện, hoặc quá hạn.

- Tích hợp biểu đồ trực quan, chẳng hạn như biểu đồ tròn hoặc cột, để minh họa trạng thái công việc hoặc hiệu suất người dùng.

5. Tăng cường bảo mật

- Triển khai mã hóa dữ liệu cục bộ trên thiết bị Android để bảo vệ thông tin người dùng.
- Áp dụng các biện pháp chống tấn công như kiểm tra SQL Injection hoặc Cross-Site Scripting (XSS) khi chuyển sang API thực tế.

XII. Kết luận

Ứng dụng Task Manager trên Flutter cung cấp một giải pháp quản lý công việc hiệu quả, được thiết kế đặc biệt cho các thiết bị Android. Với giao diện thân thiện, tích hợp API giả lập trên XAMPP, và các tính năng như quản lý người dùng, phân công công việc, tìm kiếm, và phân loại, ứng dụng đáp ứng tốt nhu cầu của cá nhân và nhóm làm việc nhỏ trong việc tổ chức và theo dõi công việc.

Việc sử dụng API giả lập giúp đơn giản hóa quá trình phát triển và kiểm thử, cho phép nhà phát triển tập trung vào xây dựng giao diện và logic ứng dụng mà không cần cấu hình cơ sở dữ liệu phức tạp. Thiết kế theo mô hình MVC và sử dụng Flutter đảm bảo ứng dụng dễ bảo trì, dễ mở rộng, và có thể chuyển đổi sang hệ thống thực tế với cơ sở dữ liệu trong tương lai.

Mặc dù hiện tại ứng dụng chỉ hỗ trợ Android và sử dụng API giả lập, các tính năng cốt lõi đã được triển khai đầy đủ, mang lại giá trị thực tế cho người dùng. Trong tương lai, việc bổ sung các tính năng như thông báo đẩy, lưu trữ cục bộ, và tích hợp trí tuệ nhân tạo sẽ nâng cao trải nghiệm người dùng, đưa Task Manager trở thành một công cụ quản lý công việc toàn diện và cạnh tranh trên thị trường.