**TỔNG QUAN DỰ ÁN MINTEST**

**Mục tiêu:** Xây dựng một ứng dụng web hoàn chỉnh để tạo, quản lý, chia sẻ và thực hiện các bài thi trắc nghiệm trực tuyến.

**Đối tượng người dùng:**

* **Admin/Superuser:** Quản trị viên cao nhất.
* **Creators (Người tạo nội dung):** Có thể tạo và quản lý nội dung thi cử (chủ đề, câu hỏi, đề thi) và xem báo cáo kết quả.
* **Members (Thành viên):** Người dùng có tài khoản, có thể làm các bài thi được chia sẻ và xem lại lịch sử của mình.
* **Guests (Khách):** Người dùng vãng lai, có thể làm các bài thi công khai hoặc được chia sẻ qua mã mà không cần tài khoản.

**I. Các Chức năng Chính và Phương pháp thực hiện**

**A. Hệ thống Quản lý Nội dung**

Đây là nền tảng của ứng dụng, cho phép Creators quản lý toàn bộ ngân hàng câu hỏi của mình.

* **Chức năng:**
  + Tạo, sửa, xóa các **Nhóm chủ đề** (TopicGroup) và các **Chủ đề** (Topic) con.
  + Tạo, sửa, xóa **Câu hỏi** (Question) và các **Đáp án** (Answer) cho từng câu hỏi.
  + Giao diện quản lý được xây dựng theo cấu trúc cây thư mục (Accordion), cho phép thao tác trực tiếp trên trang mà không cần tải lại, nhờ sử dụng **JavaScript (AJAX/Fetch) và HTMX**.
  + Hỗ trợ câu hỏi **một lựa chọn** và **nhiều lựa chọn**, hệ thống tự động xác định dựa trên số đáp án đúng được chọn.
  + Chức năng **Import/Export** hàng loạt câu hỏi từ file Excel, giúp tiết kiệm thời gian nhập liệu. Chức năng này sử dụng thư viện **Pandas** của Python để đọc và ghi file .xlsx.

**B. Hệ thống Đề thi (Quiz)**

Cho phép Creators tạo ra các bài thi từ ngân hàng câu hỏi đã có.

* **Chức năng:**
  + Hỗ trợ 2 loại đề thi: **Đề thi tĩnh** (bộ câu hỏi cố định) và **Đề thi động** (lấy câu hỏi ngẫu nhiên theo quy tắc).
  + **Giao diện tạo đề thi thông minh:**
    - Tự động chuyển đổi giao diện dựa trên loại đề thi được chọn (Tĩnh/Động).
    - Đối với đề tĩnh, giao diện chọn câu hỏi được thiết kế dạng cây thư mục với các tiện ích như **bộ đếm câu hỏi đã chọn** và **checkbox chọn tất cả** theo nhóm/chủ đề.
  + **Cơ chế chia sẻ linh hoạt:**
    - **Công khai (Public):** Admin có thể đánh dấu một đề thi là công khai để tất cả mọi người (kể cả khách) đều có thể thấy và làm bài.
    - **Mã truy cập (Access Code):** Mỗi đề thi có thể được tạo một mã chia sẻ duy nhất. Chỉ những ai có mã này mới có thể tham gia, tăng tính bảo mật và kiểm soát.

**C. Hệ thống Người dùng và Phân quyền**

Nền tảng để quản lý các loại người dùng khác nhau.

* **Chức năng:**
  + Sử dụng hệ thống **Group và Permission** có sẵn của Django để tạo 2 vai trò: Creators và Members.
  + Các chức năng quan trọng (tạo/sửa/xóa) được bảo vệ ở cả backend (dùng decorator @permission\_required) và frontend (dùng thẻ {% if perms %} để ẩn/hiện nút bấm).
  + Xây dựng luồng **Đăng ký / Đăng nhập / Đăng xuất** hoàn chỉnh cho người dùng. Sử dụng UserCreationForm và các view có sẵn LoginView, LogoutView của Django.
  + Cho phép **Khách** (người dùng không có tài khoản) làm bài thi thông qua các đề thi công khai hoặc mã truy cập. Kết quả của khách được lưu lại với tên do họ cung cấp.

**D. Trải nghiệm Làm bài thi**

Tập trung vào việc tạo ra một môi trường thi cử chuyên nghiệp và công bằng.

* **Chức năng:**
  + Giao diện làm bài hiện đại, hiển thị mỗi lần một câu hỏi.
  + **Cơ chế chống gian lận:**
    - **Thời gian được quản lý ở server:** Đồng hồ đếm ngược được tính toán dựa trên thời gian bắt đầu lưu trong CSDL, việc làm mới (F5) trang không reset lại thời gian.
    - **Thứ tự câu hỏi cố định:** Câu hỏi được xáo trộn một lần duy nhất khi bắt đầu và thứ tự này được lưu lại cho suốt lượt làm bài, đảm bảo tính nhất quán khi F5.
    - Hiển thị cảnh báo khi người dùng cố gắng rời khỏi hoặc tải lại trang thi.
  + **Tiện ích cho người dùng:**
    - **Tự động lưu tiến trình:** Mọi câu trả lời và đánh dấu đều được lưu tự động vào localStorage của trình duyệt, giúp khôi phục lại bài làm nếu người dùng lỡ tay đóng tab.
    - Chức năng **"Đánh dấu để xem lại"** (cờ) giúp người dùng dễ dàng quay lại các câu hỏi khó.
    - Hiển thị ghi chú cho biết câu hỏi là một hay nhiều đáp án đúng.

**E. Báo cáo và Thống kê**

Cung cấp cho Creators công cụ để đo lường hiệu quả của đề thi.

* **Chức năng:**
  + Trang tổng quan liệt kê các đề thi cùng số liệu tóm tắt (số lượt làm, điểm trung bình).
  + Trang báo cáo chi tiết cho từng đề thi với:
    - Biểu đồ phân bổ điểm số (sử dụng thư viện **Chart.js**).
    - Bảng danh sách kết quả của từng người làm bài (bao gồm cả khách).

**II. Các Công nghệ và Kiến trúc chính**

* **Backend:** Django.
* **Frontend:** HTML, CSS, Bootstrap 5, JavaScript (Vanilla JS).
* **Thư viện:** Pandas (xử lý Excel), Chart.js (vẽ biểu đồ), HTMX (cho một số giao diện động ban đầu).
* **Cơ sở dữ liệu:** SQLite cho môi trường phát triển. Trường TextField được dùng để lưu trữ chuỗi JSON cho các cấu trúc dữ liệu phức tạp (như thứ tự câu hỏi).
* **Deployment:** Toàn bộ ứng dụng đã được cấu hình để triển khai lên **PythonAnywhere**, với các cài đặt bảo mật cho môi trường production (DEBUG=False, ALLOWED\_HOSTS, SECRET\_KEY được quản lý bằng biến môi trường).

**III. Quy trình Làm việc**

Dự án tuân thủ quy trình làm việc chuyên nghiệp: **Local (máy cá nhân) -> Git/GitHub -> PythonAnywhere (Production)** Mọi thay đổi đều được thực hiện và kiểm thử ở local trước, sau đó được đẩy lên kho chứa GitHub, và cuối cùng được triển khai lên server bằng lệnh git pull.

**A. Cải thiện Bố cục và Luồng làm việc**

1. **Thiết kế lại Trang Bảng điều khiển (Dashboard Redesign):**
   * **Hiện tại:** Trang Dashboard khá đơn giản, chỉ hiển thị 4 con số thống kê.
   * **Ý tưởng:** Biến nó thành một "trung tâm điều khiển" thực sự. Chúng ta có thể thêm các mục như:
     + Một danh sách "Các bài thi đã làm gần đây".
     + Một danh sách "Các đề thi của bạn được làm nhiều nhất" (dành cho Creators).
     + Một form nhỏ để nhập mã tham gia thi ngay tại dashboard.
   * **Lợi ích:** Làm cho Dashboard trở nên hữu ích và có giá trị hơn, là điểm khởi đầu tốt hơn cho người dùng mỗi khi họ đăng nhập.
2. **Cải thiện "Trạng thái rỗng" (Better Empty States):**
   * **Hiện tại:** Khi người dùng mới đăng ký, các trang danh sách (Quản lý nội dung, Lịch sử,...) chỉ hiển thị một dòng chữ thông báo trống.
   * **Ý tưởng:** Thay thế các dòng chữ đơn điệu bằng các khối (panel) được thiết kế đẹp mắt, có icon minh họa và một nút kêu gọi hành động (Call-to-Action) rõ ràng. Ví dụ: *"Chào mừng! Bạn chưa có chủ đề nào cả. Hãy [Tạo Nhóm chủ đề đầu tiên]!"*
   * **Lợi ích:** Hướng dẫn và khuyến khích người dùng mới tương tác với ứng dụng, làm cho ứng dụng có cảm giác hoàn thiện và thân thiện hơn.
3. **Bố cục 2 cột cho Trang Tìm kiếm:**
   * **Hiện tại:** Trang tìm kiếm câu hỏi (question\_search.html) có bộ lọc ở trên và kết quả ở dưới, gây lãng phí không gian hai bên.
   * **Ý tưởng:** Chuyển bộ lọc tìm kiếm sang một cột sidebar bên trái. Kết quả tìm kiếm sẽ hiển thị ở cột lớn bên phải. Sidebar này có thể ẩn/hiện trên các thiết bị di động.
   * **Lợi ích:** Tận dụng không gian màn hình tốt hơn, là một bố cục rất quen thuộc và hiệu quả cho các trang có chức năng lọc/tìm kiếm.

**B. Tăng cường Tương tác và Phản hồi**

1. **Cảnh báo chống mất dữ liệu khi rời trang (Confirm on Unload):**
   * **Hiện tại:** Người dùng đang điền một form dài (như tạo câu hỏi) có thể vô tình đóng tab và mất hết dữ liệu đã nhập.
   * **Ý tưởng:** Dùng JavaScript để phát hiện nếu form đã bị thay đổi. Nếu người dùng cố gắng rời đi, trình duyệt sẽ hiện một hộp thoại xác nhận: "Các thay đổi bạn đã thực hiện có thể không được lưu. Bạn có chắc muốn rời đi?"
   * **Lợi ích:** Một tính năng rất chuyên nghiệp, bảo vệ công sức của người dùng và giảm thiểu sự thất vọng.
2. **Hệ thống thông báo "Toast":**
   * **Hiện tại:** Các thông báo (ví dụ: tham gia đề thi thành công) đang dùng chung hệ thống Message của Django, hiển thị ở đầu trang và làm xô lệch bố cục.
   * **Ý tưởng:** Xây dựng một hệ thống thông báo "Toast" nhất quán. Các thông báo ngắn như "Đã sao chép mã!", "Xóa thành công!" sẽ xuất hiện ở góc trên bên phải màn hình và tự biến mất.
   * **Lợi ích:** Cung cấp phản hồi tinh tế, hiện đại và không làm gián đoạn luồng làm việc của người dùng.

**C. Nâng cao Trải nghiệm Cốt lõi**

1. **Bộ lọc "Chỉ xem câu sai" trên trang kết quả:**
   * **Hiện tại:** Trang kết quả (attempt\_result.html) hiển thị tất cả các câu hỏi.
   * **Ý tưởng:** Thêm các nút lọc (ví dụ: "Tất cả", "Đúng", "Sai") ở đầu trang kết quả. Khi người dùng nhấn vào "Sai", trang sẽ chỉ hiển thị danh sách các câu hỏi mà họ đã trả lời sai.
   * **Lợi ích:** Một tính năng cực kỳ hữu ích cho việc học tập và ôn lại kiến thức. Người dùng có thể nhanh chóng tập trung vào những phần họ còn yếu.
2. **Chế độ Tối/Sáng (Dark/Light Mode):**
   * **Hiện tại:** Giao diện chỉ có một chế độ sáng.
   * **Ý tưởng:** Thêm một nút bấm (thường ở header hoặc sidebar) cho phép người dùng chuyển đổi giữa giao diện sáng và tối. Lựa chọn của người dùng sẽ được lưu lại trong trình duyệt cho các lần truy cập sau.
   * **Lợi ích:** Một tính năng rất phổ biến và được yêu thích. Nó không chỉ làm cho trang web trông "xịn" hơn mà còn giúp bảo vệ mắt người dùng khi sử dụng vào ban đêm.