|  |
| --- |
| Tôi đang làm dự án cá nhân "Daily.dev challenge": Viết code backend nhằm tạo trang tổng hợp thông tin(tương tư như trang daily.dev,...) từ nhiều nguồn khác nhau. Mô tả công việc: Tạo các controller để quản lý, tìm source (source kiếm từ RSS hoặc API ví dụ: Báo mới, tuổi trẻ, dân trí,...) của các news, đọc source đó và lưu source xuống database và hiển thị lên trang tổng hợp. Đăng ký source, mỗi khi mở trang tổng hợp thì phần mềm sẽ đi tìm từng source và insert từng source đó vào database. Sau đó từ database lấy những source của những news mà người dùng thích nhất lên trang tin chính. Hoặc hiển thị lên những news mà chủ đề người dùng lựa chọn. cập nhật tin tức mỗi ngày hoặc mỗi giờ, ưu tiên hiện những news mà người dùng yêu thích hoặc theo top like,... Hiện tại chỉ dừng lại ở mức độ backend và project nhỏ nên chưa quan tâm đến bảo mật và phần front end giao diện. Công nghệ: SQL Server, C#, các framework của C#. |
| **Quản lý nguồn tin (source):**   * Tìm và thu thập nguồn tin từ RSS hoặc API của các báo (Báo mới, Tuổi trẻ, Dân trí,...). * Lưu thông tin nguồn tin vào cơ sở dữ liệu. * Hiển thị các nguồn tin đã lưu lên trang tổng hợp.   **Đăng ký nguồn tin:**   * Cho phép người dùng đăng ký các nguồn tin yêu thích. * Mỗi lần mở trang, hệ thống sẽ kiểm tra và lấy tin mới từ các nguồn đã đăng ký.   **Lưu và quản lý bài báo:**   * Sau khi lấy được nguồn tin, hệ thống sẽ lưu các bài báo tương ứng xuống cơ sở dữ liệu. * Hiển thị các bài báo dựa trên sở thích của người dùng (ví dụ: những bài báo được yêu thích, chủ đề cụ thể, top like).   **Cập nhật tự động:**   * Hệ thống sẽ tự động cập nhật tin tức hàng ngày hoặc hàng giờ. * Ưu tiên hiển thị những bài báo người dùng yêu thích hoặc theo top bài báo được like nhiều nhất. |
| **1. Thiết kế Cơ sở Dữ liệu**   * **Kiểm tra và hoàn thiện các bảng trong database:**   + Đảm bảo rằng các bảng Provider, Category, Item, Tag, NewTag, User, UserCategory, UserTag, TableConfig đã được thiết kế đúng với cấu trúc cần thiết.   **2. Tạo Mô hình Dữ liệu (Data Models)**   * **Tạo các class tương ứng với các bảng trong cơ sở dữ liệu:**   + Provider, Category, Item, Tag, NewTag, User, UserCategory, UserTag, TableConfig.   **3. Tạo các Repository**   * **Xây dựng các lớp repository để tương tác với cơ sở dữ liệu:**   + Phương thức để thêm, sửa, xóa, và lấy dữ liệu từ các bảng.   **4. Tạo Controllers**   * **Tạo các controller cho từng entity:**   + **ProviderController**: Quản lý nguồn tin.   + **CategoryController**: Quản lý danh mục.   + **ItemController**: Quản lý tin tức.   + **TagController**: Quản lý thẻ.   + **UserController**: Quản lý người dùng và sở thích của họ.   **5. Tạo Logic Đọc và Lưu Tin từ RSS/ API**   * **Tạo phương thức trong controller hoặc service để:**   + Lấy dữ liệu từ các nguồn RSS hoặc API (Báo mới, Tuổi trẻ, Dân trí,...).   + Phân tích và lưu dữ liệu vào cơ sở dữ liệu.   **6. Xây dựng Logic Cập nhật Tin Tức**   * **Tạo một cron job hoặc background service:**   + Cập nhật tin tức từ các nguồn định kỳ (hàng giờ hoặc hàng ngày).   + Chỉ lấy những tin tức mới hoặc đã được cập nhật.   **7. Tạo Endpoint cho Frontend**   * **Tạo các endpoint để lấy thông tin cho trang tổng hợp:**   + Lấy danh sách các nguồn tin, tin tức theo chủ đề yêu thích của người dùng.   + Lấy danh sách tin tức theo top like, most read,...   **8. Tạo Endpoint cho Đăng Ký Nguồn**   * **Cung cấp endpoint cho người dùng đăng ký các nguồn tin:**   + Cho phép người dùng chọn các nguồn tin yêu thích và lưu chúng vào cơ sở dữ liệu.   **9. Tạo Logic Lọc Tin Tức**   * **Phát triển logic để lọc và hiển thị tin tức theo sở thích của người dùng:**   + Dựa vào các thông tin từ bảng UserCategory, UserTag, và TableConfig. |
| **1. Guest (Người dùng chưa đăng nhập)**   * **Xem danh sách bài viết**:   + **Mục tiêu**: Người dùng có thể xem danh sách các bài viết mới nhất hoặc phổ biến từ nhiều nguồn.   + **Mô tả**: Hiển thị các bài viết theo mặc định mà không yêu cầu người dùng đăng nhập.   + **Hành động**: Xem, Lọc bài viết theo thẻ hoặc nguồn. * **Tìm kiếm bài viết**:   + **Mục tiêu**: Người dùng có thể tìm kiếm bài viết theo từ khóa.   + **Mô tả**: Cho phép người dùng không cần đăng nhập để tìm kiếm.   + **Hành động**: Nhập từ khóa và tìm kiếm bài viết. * **Đăng ký tài khoản**:   + **Mục tiêu**: Cho phép người dùng tạo tài khoản để có thêm tính năng cá nhân hóa.   + **Hành động**: Đăng ký tài khoản qua email hoặc đăng nhập qua OAuth (Google, GitHub, etc.). * **Xem các nguồn tin phổ biến**:   + **Mục tiêu**: Khuyến khích người dùng khám phá các nguồn tin chất lượng.   + **Hành động**: Xem danh sách các nguồn tin và số lượng bài viết từ mỗi nguồn.   **2. Registered User (Người dùng đã đăng ký)**   * **Xem và quản lý Feed cá nhân**:   + **Mục tiêu**: Cung cấp một luồng tin tức cá nhân hóa dựa trên sở thích của người dùng.   + **Mô tả**: Người dùng đã đăng ký có thể theo dõi nguồn tin và thẻ yêu thích.   + **Hành động**:     - Xem feed được cá nhân hóa.     - Theo dõi và bỏ theo dõi nguồn tin.     - Theo dõi và bỏ theo dõi thẻ.     - Lọc bài viết theo thẻ hoặc nguồn đã theo dõi. * **Tương tác với bài viết**:   + **Mục tiêu**: Người dùng có thể tương tác với bài viết qua các hành động như like, comment, bookmark.   + **Hành động**: Like, Comment, Bookmark bài viết để lưu lại xem sau. * **Quản lý danh sách bookmark**:   + **Mục tiêu**: Người dùng có thể lưu lại các bài viết để xem sau.   + **Hành động**:     - Xem danh sách các bài viết đã bookmark.     - Thêm/xóa bài viết khỏi danh sách bookmark. * **Quản lý lịch sử đọc**:   + **Mục tiêu**: Người dùng có thể xem lại lịch sử các bài viết đã đọc.   + **Hành động**:     - Xem danh sách các bài viết đã đọc.     - Xóa lịch sử đọc nếu cần. * **Submit bài viết**:   + **Mục tiêu**: Người dùng có thể gửi bài viết hoặc liên kết để xem xét và đăng tải.   + **Hành động**: Gửi liên kết bài viết mới để hệ thống kiểm duyệt. * **Chỉnh sửa thông tin cá nhân và sở thích**:   + **Mục tiêu**: Cho phép người dùng quản lý thông tin tài khoản và sở thích về nguồn tin và thẻ.   + **Hành động**:     - Cập nhật thông tin tài khoản.     - Cập nhật sở thích (thêm/xóa thẻ và nguồn tin).   **3. Admin (Quản trị viên)**   * **Quản lý bài viết**:   + **Mục tiêu**: Quản trị viên có thể duyệt và quản lý các bài viết được đăng tải.   + **Hành động**:     - Duyệt bài viết do người dùng submit.     - Xóa bài viết nếu vi phạm.     - Chỉnh sửa nội dung bài viết nếu cần thiết. * **Quản lý người dùng**:   + **Mục tiêu**: Quản trị viên có thể quản lý thông tin người dùng.   + **Hành động**:     - Cập nhật hoặc khóa tài khoản người dùng vi phạm.     - Xem danh sách và thông tin người dùng. * **Quản lý nguồn tin**:   + **Mục tiêu**: Admin có thể thêm, chỉnh sửa, hoặc xóa các nguồn tin.   + **Hành động**:     - Thêm nguồn tin mới vào hệ thống.     - Cập nhật thông tin nguồn tin hiện có.     - Xóa nguồn tin khỏi hệ thống. * **Quản lý thẻ**:   + **Mục tiêu**: Admin có thể thêm, chỉnh sửa, hoặc xóa các thẻ trong hệ thống.   + **Hành động**: Thêm, Cập nhật, Xóa thẻ. * **Quản lý tương tác và báo cáo vi phạm**:   + **Mục tiêu**: Admin có thể theo dõi các báo cáo vi phạm về bài viết, bình luận từ người dùng.   + **Hành động**:     - Kiểm tra các báo cáo vi phạm.     - Xử lý các bài viết hoặc người dùng vi phạm. |
|  |
|  |