|  |
| --- |
| 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#, ADO và các framework của C#. |
| **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 ngày.   + Chỉ lấy những tin tức mới hoặc đã được cập nhật.   **7. Tạo Endpoint cho Frontend**   * + Lấy thông tin về người dùng và các sở thích của họ (đã đăng ký các nguồn nào).   + Endpoint lấy danh sách các tin tức mà người dùng yêu thích (dựa trên lượt thích, hoặc các danh mục/tag mà họ đã theo dõi).   **8. Tạo Endpoint cho Đăng Ký Nguồn**   * Endpoint cho phép người dùng đăng ký, hủy đăng ký nguồn tin yêu thích. * **Bổ sung:**   + Lưu thông tin về hành vi người dùng (lượt xem, tương tác) để phân tích.   + Tạo tính năng "thích" hoặc "bỏ thích" (like/dislike) các tin tức, và lưu thông tin đó vào bảng thích hợp.   **9. Tạo Logic Lọc Tin Tức**   * **Phát triển logic để hiển thị tin tức dựa trên sở thích của người dùng:**   + Lọc theo **UserCategory**, **UserTag**, và **TableConfig** để đề xuất tin tức liên quan.   + **Bổ sung:** Tính toán điểm (scoring) cho các bài viết dựa trên hành vi người dùng, như lượt xem, số lượt thích, và thời gian tương tác.   **10. Tạo Module Quản Lý Người Dùng (User Management)**   * Tạo các endpoint để quản lý thông tin cá nhân của người dùng:   + Đăng ký tài khoản, đăng nhập, và cập nhật thông tin người dùng.   + Xử lý việc lưu trữ thông tin về sở thích (danh mục, thẻ, nguồn tin).   + **Bổ sung:** Quản lý các cấu hình người dùng (ví dụ: cài đặt tần suất cập nhật tin tức, thông báo).   **11. Cải thiện hiệu năng và mở rộng khả năng cập nhật**   * **Bổ sung:**   + Cải tiến hệ thống caching để tăng tốc độ lấy tin tức.   + Xây dựng cơ chế phát hiện thay đổi nội dung từ RSS/API để giảm thiểu tải khi cập nhật dữ liệu.   **12. Tích hợp chức năng thông báo (Notification)**   * Xây dựng hệ thống thông báo để gửi thông tin về các tin tức mới hoặc tin nổi bật mà người dùng có thể quan tâm:   + Gửi email hoặc thông báo đẩy (push notifications) khi có tin tức phù hợp với sở thích của người dùng. |
|  |
|  |
|  |