



# Full-Stack Interview Task: Personalized News Aggregator

## Objective

Build a Personalized News Aggregator Application that allows users to:

- Register and securely log in.
- Receive a tailored news feed based on individual preferences.
- Customize news settings: country, sources, and keywords.
- Browse, search, and filter their news feed.
- Access original articles via source links.

## Technical Requirements

### Backend

- Framework: Python 3.x, Django, Django REST Framework
- Database: PostgreSQL
- Scheduler: Lightweight background task scheduler for periodic news fetching
- News API: Integrate with [NewsAPI.org](https://newsapi.org/) free plan

## Frontend

- Framework: React or Vue 3
- Styling: Tailwind CSS, Bootstrap (use any), or custom CSS

## Functional Specifications

### 1. User Authentication

- Implement user registration and login via email or username with password.
- Secure all API endpoints, ensuring proper access control to user data and news.

### 2. User News Preferences

- Allow users to manage preferences including:
  - Countries (e.g., `us`, `nz`)
  - Sources (e.g., `bbc-news`, `cnn`)
  - Keywords (e.g., `car`, `climate`, `sports`)
- Set default preferences on registration:
  - Countries: `["nz"]`
  - Sources: `["bbc-news", "cnn"]`
  - Keywords: `["car", "automobile"]`
- Provide APIs and frontend interfaces to view and update preferences anytime.

### 3. News Retrieval and Storage

- Schedule a background task every 10 minutes to:
  - Fetch the latest articles from NewsAPI.org based on all users' preferences.
  - Store matching articles in the database linked to each user.
  - Prevent duplicate storage using uniqueness constraints.

### 4. Personalized News Feed

- Deliver a paginated news feed tailored to the authenticated user's preferences.
- Display for each article:
  - Title
  - Summary
  - Source name
  - Source URL (opens in a new tab)
  - Publication date
  - Image thumbnail (if available)
- Support:
  - Pagination
  - Search by keyword or title
  - Filtering by source and publication date

## Deliverables

- Well-structured database models and backend APIs.
- Frontend application with responsive UI for registration, login, preferences management, and news browsing.
- Background scheduler integration for periodic news updates.
- Clean, readable, and maintainable code following best practices.

You should push your code to a public git repository like GitHub with clear instructions about how to run your program. Share the repo link in the mail.

We encourage you to complete this task independently, without the assistance of AI tools, as our goal is to assess your skills and understanding. Thank you!