# Local News Aggregator Project

This project contains a Python backend server and an HTML/JavaScript frontend to create a local news aggregator that fetches, stores, and displays news from various sources.

## Screenshot

## How It Works

The project is composed of two main parts that work together: a Python backend and an HTML frontend.

1. **Backend (local\_server.py):**
   * **Web Server:** It runs a lightweight Flask web server to listen for requests from the frontend.
   * **Database:** It uses SQLite to create a local database file (news.db) to store articles. This prevents re-fetching the same articles and makes loading fast.
   * **News Fetching:** When the /fetch-news API is called, the server uses the feedparser library to download and parse RSS feeds from the configured news sources.
   * **API:** It provides API endpoints (/get-news, /fetch-news, /track-view) for the frontend to interact with.
2. **Frontend (index.html):**
   * **User Interface:** This is the file you open in your browser. It contains the HTML structure and is styled with Tailwind CSS.
   * **Client-Side Logic:** It uses JavaScript to communicate with the backend. When you click "Fetch Latest News Now," it sends a request to the backend's /fetch-news endpoint.
   * **Rendering:** After fetching the news data from the /get-news endpoint, JavaScript dynamically creates the news cards and displays them on the page, allowing you to filter by category and language.

## Files

1. **local\_server.py**: The Flask web server that handles fetching news from RSS feeds, storing it in a local database, and providing it to the frontend via an API.
2. **index.html**: The frontend of the application that you open in your browser. It displays the news and allows you to filter by category and language.

## How to Run

Follow these steps to get the project running on your computer:

### Step 1: Prerequisites

Make sure you have Python installed. Then, open your terminal or command prompt and install the required Python libraries:

pip install Flask Flask-Cors feedparser

### Step 2: Save the Files

Save both local\_server.py and index.html in the **same folder** on your computer.

### Step 3: Run the Backend Server

Navigate to the folder where you saved the files using your terminal and run the following command:

python local\_server.py

You should see output indicating that the server is running, like this:

\* Running on http://127.0.0.1:5000

**Important:** Keep this terminal window open. Closing it will stop the server.

### Step 4: Open the Frontend

Now, simply find the index.html file in your file explorer and double-click it. It will open in your default web browser.

### Step 5: Fetch the News

The page will initially show a loading spinner. The first time you run it, the database (news.db) will be empty.

Click the **"Fetch Latest News Now"** button at the top of the page. This will tell the backend to start downloading articles from all the news sources. This might take a minute. Once it's done, the news articles will appear on the page.

You are all set! You can now browse the news, filter by categories, and switch between languages in the "Indian News" section.