NEWS WEBSITE USING WEB SCRAPING

PROJECT OVERVIEW

The news website is built using a combination of web scraping, backend development, and frontend design. The key components include:

- A web scraper to fetch news articles from reliable sources.
- A backend system to store and serve the scraped data.
- A frontend interface to present news articles in an intuitive manner.

METHODOLOGY

1. **Web Scraping Component**

- Identified reliable news sources such as BBC.
- Used Selenium in Node.js for extracting relevant news data in real-time.
- Process:
    1. Open target news websites using Selenium.
    2. Locate article urls from the html and save them in an array.
    3. Parse through the array. For each article, open the url and locate article elements (headlines, author, published date, content, images, source link).
    4. Extract and store the data in a structured format.
    5. Save the extracted data in Postgres for easy access and retrieval.
    6. Perform this for each category including the home page.

2. **Data Storage**

- Database: PostgreSQL
- Schema: Seven tables are present in the schema: 1. News, 2. Business, 3. Culture, 4. Arts, 5. Earth, 6. Travel, 7. Innovation

- Why Postgres?
    - Stores data in well-structured tables.
    - Easily scalable for future expansion.
    - Easily accessible from the backend.

3. **Backend Development**

- Developed the backend using Express.js to fetch and serve real-time news data via a REST API.
- Used PostgreSQL as the database to store the extracted news articles.
- Created separate tables for different categories:
    - news
    - travel
    - earth
    - culture
    - art
    - business
- Routes Implemented:
    - GET /news → Fetch all stored news articles.
    - GET /business → Fetch articles from business category.
    - GET /culture → Fetch articles from culture category.
    - GET /arts → Fetch articles from arts category.
    - GET /future-planet → Fetch articles from earth category.
    - GET /innovation → Fetch articles from innovation category.
    - GET /travel → Fetch articles from travel category.
    - GET /article/:id?category=category → Fetch a single article from a category.

- Implemented error handling to manage failed scrapes and incomplete articles.
- Used node-cron to schedule scraping every 3 hours, with the database being truncated at the start of each schedule.

4.**Frontend**

- Used React.js and Tailwind CSS for building the frontend.
- Designed a user-friendly UI with the following pages:
    - Homepage: Displays top news articles and featured articles with images and summaries. Added a Header and Footer for better navigation between pages.
    - Categories: Sections for News, Travel, Earth, Culture, Art, Innovation and Business.
    - Search: Allows users to find specific news articles from the page.
    - Detailed News Page: Displays full article details for each news piece and a more articles section for user convenience.
    - Ensured responsiveness and cross-browser compatibility.
    - Handled articles with improper image urls and unknown authors.
    - Added pagination for better navigation.

CHALLENGES AND SOLUTIONS

Challenge	**Solution**
Frequent structural changes in source websites	Implemented robust error handling and fallback mechanisms.
Handling large volumes of news data efficiently	Optimized the scraping process and used a scalable PostgreSQL database.
Ensuring real-time updates without excessive server load	Implemented scheduled scraping with **node-cron** and automatic database truncation.