Assignment 2: API

Objective

The objective of this assignment is to learn how to work with APIs to retrieve, process, and display data. Enhance your project by integrating multiple APIs and presenting the information effectively. Develop clean, responsive, and user-friendly applications using best practices in coding and project design.

Requirements

- ❖ These APIs should run only on the server side and offer valuable data to users. Display this data in frontend of your work.
- ❖ Your work logic must be implemented inside your core js file, not in html file.

1. **Random User Generator API - https://randomuser.me/api/**

The application must have a button to get a random user from the Random User API in Node.js backend and extract specific personal and location details. The following data must be extracted and displayed on the frontend:

- First name,
- Last name,
- Gender,
- Profile picture,
- Age,
- Date of birth,
- City,
- country,
- Full address (street name and number).

User information should be presented clearly using: cards or structured sections, profile image and labeled data fields.

Note: your work logic must be implemented inside your core js file, not in html file.

2. **Integrate Countrylayer API based on user's country - https://manage.countrylayer.com/signup/free**

The Countrylayer API provides RESTful endpoints that return structured data about countries worldwide.

Using the *country name* obtained from the RandomUser API in task 1, the application retrieves these attributes: ***Country name, Capital city, Official language(s), Currency, National flag.***

When integrating the Countrylayer API, the server must:

- Use an API key stored securely in environment variable
- Parse the response to extract required fields
- Handle missing or unavailable data gracefully
- Send only relevant, cleaned data to the frontend

3. **Add Exchange Rate API based on user's currency - https://www.exchangerate-api.com/**

Using the *currency* obtained from the Countrylayer API in task 2, show how the user's local currency compares to: United States Dollar (USD) and Kazakhstani Tenge (KZT).

Example Output: 1 EUR = 1.08 USD, 1 EUR = 495.20 KZT.

The exchange rate section should be displayed near the country information to maintain logical grouping of related data.

4. **Show headlines with News API based on the user's country - https://newsapi.org/**

Get news headlines related to the random user's *country* from task 1. News data must be fetched on the server side. Retrieve five headlines in English language and the headline must contain the user's country name.

Each news article must display (get this information from API response):

- Headline title
- Image (if available)
- Short description
- Source URL (link to full article)

User information should be presented clearly using: cards or structured sections, a profile image, and labeled data fields.

5. **Project Organization and Design**
   - **Clean Code and Project Structure:** Keep your code clean, well-documented, and organized. Follow best practices for coding and

maintain a clear project structure that you learned from previous assignments.

- **Design and User Interface:**
    i. Enhance the user interface with thoughtful design elements, making the application visually appealing.
- **Server Configuration and Dependencies:**
    i. The server should run on port 3000.
    ii. Ensure that the submission includes a package.json file with all dependencies listed.

Submission:

- Submit the following:
    - A .zip file or a GitHub repository link containing your project.
    - Additionally, provide comprehensive documentation, including setup instructions, API usage details, and explanations of key design decisions in your README.md. *Your work logic must be implemented inside your core js file, not in html file.*

Grading Criteria:

Criteria	Description	Weight
1. Core requirements	The application implements the fundamental features as specified: Random User API	25%
2. Additional APIs Integration	The project integrates News API, CountryLayer API, ExchangeRate API. These APIs must run server-side and provide meaningful data displayed on the frontend.	30%
3. Project Organization and Design	The project demonstrates clean code practices, follows an organized structure, and implements a responsive and visually appealing user interface.	10%
4. Defense	The student clearly explains the functionality, structure, and decisions behind the code during the defense. They should also answer theoretical questions related to the project and the concepts used.	35%