COVID-19 Data Tracker Documentation

Introduction

This document outlines the plan and key details for the development of the COVID-19 Data Tracker application. The purpose of this project is to create a web-based application that extracts, stores and visualizes COVID-19 data from reliable sources. The project will be implemented using JavaScript, with Express.js and MongoDB for the backend, React, Tailwind CSS, and Framer Motion for the front end.

Tools and Technologies

Backend

- **JavaScript**: The primary programming language for backend development.
- **Express.js**: A fast and minimalist web framework for building server-side applications.
- **MongoDB**: A NoSQL database for storing COVID-19 data efficiently.
- **Helmet**: A middleware for securing the application by setting various HTTP headers.
- **Docker**: Containerization tool for easy deployment.

Fronted

- **React:** A popular JavaScript library for building user interfaces.
- **Tailwind CSS**: A utility-first CSS framework for responsive and stylish UI design.
- **Framer Motion**: A library for creating smooth animations and interactive UI elements.

Schedule

Week 1: Backend Development and Data Extraction

1. **Day 1-2:** *Project Setup and Backend Skeleton*
    A. Set up your project structure, install dependencies, and create a basic server setup using Express.
    B. Integrate Helmet and other security tools for enhanced security measures.
2. **Day 3-4:** *Data Extraction and MongoDB Integration*
    A. Implement the data extraction endpoint to fetch COVID-19 data from the chosen API.
    B. Integrate MongoDB for data storage, design a schema, and set up models.
3. **Day 5-6:** *Advanced Data Handling and Filtering*
    A. Enhance the data extraction endpoint to allow more advanced filtering options (e.g., date range, country).
    B. Implement server-side validation and sanitize user input.
4. **Day 7-8:** *Data Visualization and Backend Finalization*
    A. Begin working on the integration of data visualization components using Chart.js.
    B. Ensure that the backend code is well-structured and follows best practices.

Week 2: Frontend Development, Data Visualization, and Deployment

5. **Day 9-10:** *Frontend Setup and UI Skeleton*
    A. Set up a React frontend project.
    B. Design a basic UI layout using Tailwind CSS.
6. **Day 11-12:** *Interactive Data Visualization*
    A. Integrate Chart.js to create interactive graphs and charts for COVID-19 data.
    B. Implement a date picker component for custom date range selection.

7. **Day 13-14:** *Animations and Responsive Design*
    A. Incorporate animations using Framer Motion to enhance user experience.
    B. Ensure the user interface is fully responsive across different devices and screen sizes.
8. **Day 15-16:** *Advanced Features (Optional)*
    A. Implement any additional features you consider important for the project within the available time.
9. **Day 17-18:** *Testing, Deployment, and Documentation*
    A. Thoroughly test the application, including data extraction, visualization, security, and responsiveness.
    B. Deploy the full-stack application to a cloud platform (e.g., Heroku) for online accessibility.
    C. Write comprehensive documentation covering setup, usage, and any advanced features.

Features and Functionalities

The COVID-19 Data Tracker application will include the following key features and functionalities:

1. **Data Extraction and Storage:**
    - Implement an endpoint to fetch COVID-19 data from a reliable API source.
    - Store the extracted data in a MongoDB database for efficient retrieval.
2. **Data Visualization:**
    - Create interactive visualizations using Chart.js to display COVID-19 statistics.
    - Present data such as daily new cases, total cases, and total deaths.
3. **Filtering and Search:**
    - Allow users to filter data based on parameters like country and date range.
4. **Responsive UI:**
    - Design a responsive user interface using Tailwind CSS, ensuring compatibility across devices.
5. **Animations:**
    - Enhance user experience with smooth animations using Framer Motion.
6. **Documentation:**
    - Provide clear and concise documentation for setting up, running, and using the application.