**Documentation**

**real-time stock market data web application**

Project Timeline:

* **A project timeline is a visual list of tasks or activities placed in chronological order, which lets project managers view the entirety of the project plan in one place.** A project timeline typically takes the form of a horizontal bar chart, where each task is given a name and a corresponding start and end date.
* A project timeline provides an in-depth overview of the entire project from start to finish. You can see when a task starts and when it’s due—and importantly, whether or not it’s dependent on another task.
* A project timeline can be priceless for a project team.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Project Description (2 Months):

Week-Wise distribution of tasks up to 2 Months.

**MONTH-1 Details:**

**Week-1:**

* Introduction to React.js
  + Understanding components, JSX syntax, and state management in React.js.
* Setting up the development environment
  + Installing Node.js, creating a new React.js application, and setting up a basic file structure.

**Week-2:**

* Styling the application
  + Styling the application using CSS and Sass to create a responsive design.
* Implementing user authentication
  + Setting up user authentication using JSON Web Tokens (JWT) and Passport.js.

**Week-3:**

* Pulling data from APIs
  + Understanding the Alpha Vantage API and using it to pull real-time stock market data.
* Displaying data on the frontend
  + Using React.js components to display the real-time stock market data in an easy-to-understand format.

**Week-4:**

* Adding search functionality
  + Adding search functionality to the application to allow users to search for specific stocks.
* Testing and debugging
  + Using testing frameworks such as Jest and Enzyme to test the application and debug errors.

**MONTH-2 Details:**

**Week-1:**

* Implementing data visualization
  + Adding data visualization to the application using charting libraries such as D3.js or Chart.js.
* Handling errors and edge cases
  + Handling errors and edge cases in the application such as incorrect user input or API errors.

**Week-2:**

* Building the backend
  + Building the backend of the application using Node.js and Express.js.
* Creating APIs
  + Creating APIs to handle user authentication, data retrieval, and data manipulation.

**Week-3:**

* Integrating News API
  + Integrating the News API to display relevant news articles related to the selected stocks.
* Implementing real-time updates
  + Implementing real-time updates using Web-Sockets to ensure that the data displayed is always up-to-date.

**Week-4:**

* Optimizing performance
  + Optimizing the performance of the application using techniques such as code splitting, lazy loading, and caching.
* Deploying the application
  + Deploying the application to a web server such as Heroku or AWS.

References:

**1. Introduction to React.js:**

- Official React.js Documentation: <https://reactjs.org/docs/getting-started.html>

**2. Setting up the development environment:**

- Create React App Documentation: <https://create-react-app.dev/docs/getting-started/>

**3. Styling the application:**

- CSS Basics: <https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/CSS_basics>

- Sass Documentation: <https://sass-lang.com/documentation>

**4. Implementing user authentication:**

- JWT.io Introduction: <https://jwt.io/introduction/>

- Passport.js Documentation: <http://www.passportjs.org/docs/>

**5. Pulling data from APIs:**

- Alpha Vantage API Documentation: <https://www.alphavantage.co/documentation/>

- Fetch API Guide: <https://developer.mozilla.org/en-US/docs/Web/API/Fetch_API/Using_Fetch>

**6. Displaying data on the frontend:**

- React.js Documentation: <https://reactjs.org/docs/introducing-jsx.html>

**7. Adding search functionality:**

- React.js Controlled Components: <https://reactjs.org/docs/forms.html#controlled-components>

**8. Testing and debugging:**

- Jest Documentation: <https://jestjs.io/docs/getting-started>

- Enzyme Documentation: <https://enzymejs.github.io/enzyme/>

**9. Implementing data visualization:**

- D3.js Documentation: <https://d3js.org/>

- Chart.js Documentation: <https://www.chartjs.org/docs/latest/>

**10. Building the backend:**

- Node.js Documentation: <https://nodejs.org/en/docs/>

- Express.js Documentation: <https://expressjs.com/>

**11. Integrating News API:**

- News API Documentation: <https://newsapi.org/docs>

**12. Implementing real-time updates:**

- Web-Sockets Guide: <https://developer.mozilla.org/en-US/docs/Web/API/WebSockets_API>

**13. Optimizing performance:**

- React.js Performance Optimization: <https://reactjs.org/docs/optimizing-performance.html>

**14. Deploying the application:**

- Heroku Documentation: <https://devcenter.heroku.com/>

- AWS Documentation: <https://docs.aws.amazon.com/>

Acknowledgment:

Creating a project timeline using a Gantt chart and discussing its various aspects has been an insightful topic to explore. I would like to acknowledge the importance of project managers and teams in utilizing project timelines to effectively plan, track, and manage projects. Additionally, I appreciate the challenges involved in accurate task estimation, managing dependencies, resource allocation, and handling changes. The significance of effective communication, collaboration, and adaptability in overcoming these challenges is duly acknowledged. The use of project management software and tools, such as Microsoft Project, Smartsheet, and Team-Gantt, deserves recognition for their role in simplifying the process of creating project timelines. Overall, understanding the project timeline's role and challenges contributes to successful project execution.

Contact Information:

For any inquiries, questions, or

support, please contact: [2020mcb1241@iitrpr.ac.in](mailto:2020mcb1241@iitrpr.ac.in)

**Milind Rathore**

**From-Web Developer Group**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***END**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***