****

**Department of Computer Science and Engineering**

**BENGALURU, KARNATAKA, INDIA.**

**B. TECH. (CSE)**

**V SEMESTER**

**Aug. – Dec. 2023**

**UE21CS341A – SOFTWARE ENGINEERING**

**PROJECT REPORT**

**ON**

**Weather Forecasting**

**SUBMITTED BY**

**PES1UG21CS465     -      R Sumedh**

**PES1UG21CS484     -      Reehan Shaveez**

**PES1UG21CS471     -      Rahul Kalekar**

**PES1UG21CS470     -      Raghavendra SB**

**SOFTWARE ENGINEERING PROJECT REPORT**

1. **Identify the lifecycle to be followed for the execution of your project and justify why you have chosen the model**

The lifecycle that we have chosen for our model goes as follows:

**Planning and Requirement Analysis**: Understand the objectives of the app, define features, and gather requirements such as the specific weather data to display, user interface elements, and functionality.

**Development**: Implement the frontend using HTML, CSS, and JavaScript (index.html in your case) for displaying data. Develop the backend (server.js) to fetch data from the weather API, process it, and provide it to the frontend.

**Testing:** Perform unit tests for both frontend and backend to ensure functionality, handle edge cases, and validate data accuracy. Conduct integration tests to check if the frontend interacts correctly with the backend.

**Deployment:** Deploy the app on a server or a cloud platform to make it accessible to users. Configure the environment and set up any necessary server infrastructure.

**Monitoring and Maintenance**: Continuously monitor the app for performance, security, and user feedback. Implement updates, fixes, and enhancements as needed to ensure the app runs smoothly.

I chose this model because it follows a structured approach, allowing for systematic development while addressing potential issues at different stages. It ensures that requirements are met, errors are minimized through testing, and the app is maintained effectively post-deployment.

1. **Identify the tools which u want to use it throughout the lifecycle like planning tool, design tool, version control, development tool, bug tracking, testing tool.**

**Development Tools:**

**IDEs (Integrated Development Environments):** Such as Visual Studio Code for coding HTML, CSS, JavaScript, and Node.js.

Node.js: For backend JavaScript execution.

**Testing Tools**: Unit Testing was done by comparing the results with various available sources across the internet. Further refinements to the project were made based on it.

**Planning Tool:** GitHub

1. **Determine all the deliverables and categorize them as reuse/build components and justify the same.**

**Reuse Components:**

**Third-party Weather API Integration**: The code and configuration that fetch weather data from an external API (e.g., OpenWeatherMap, WeatherAPI). This can be considered a reusable component as it abstracts the functionality of retrieving weather information, and it's likely to remain unchanged across various projects or versions of your app.

**UI Components or Libraries**: Any reusable UI components or libraries used in the frontend (HTML, CSS, JavaScript) that serve general purposes, such as custom-styled buttons, navigation bars, or reusable design elements. These components can be reused in other parts of the app or in future projects, saving development time and ensuring consistent design.

**Build Components**:

**App-Specific Frontend Code (index.html):** The HTML, CSS, and JavaScript files that constitute the specific user interface and functionality of your weather forecasting app. This includes components like the layout, interaction elements, and data presentation tailored for this particular app.

**Backend Code (server.js):** The backend logic responsible for handling requests, processing data from the weather API, and sending the necessary information to the frontend. This code is specific to the app's functionality and is built to serve the purpose of retrieving and formatting weather data for display.

**Justification for Reuse Components:**

These components are categorized as reusable because they encapsulate functionalities that are independent of the specific application logic. They are designed to be versatile and applicable across different projects or versions without substantial modification.

**Justification for Build Components:**

These components are specific to the functionalities and design of your weather forecasting app. They are crafted to meet the unique requirements and specifications of this particular project and are less likely to be directly reusable in other applications without significant modifications.

1. **Create a WBS for the entire functionalities in detail.**

1**. Planning Phase:**

Objective Identification

Requirements Gathering

Task Identification and Assignment

2. **Design Phase:**

Architecture Design

Frontend Structure

Backend Structure

Database Design (if applicable)

3**. Development Phase:**

Frontend Development

HTML Structure

CSS Styling

JavaScript Functionality

Backend Development

Setting up Server (Node.js/Express.js)

API Integration (Weather API)

Data Processing and Formatting

4. **Testing Phase:**

Unit Testing

Frontend Logic

Backend Logic

Integration Testing

Frontend-Backend Interaction

API Response Testing

User Acceptance Testing (UAT)

Feedback Collection

Bug Fixes

5. **Deployment Phase:**

Server Setup

Deployment of Frontend and Backend

Configuration and Optimization

6. **Maintenance Phase:**

Monitoring and Performance Optimization

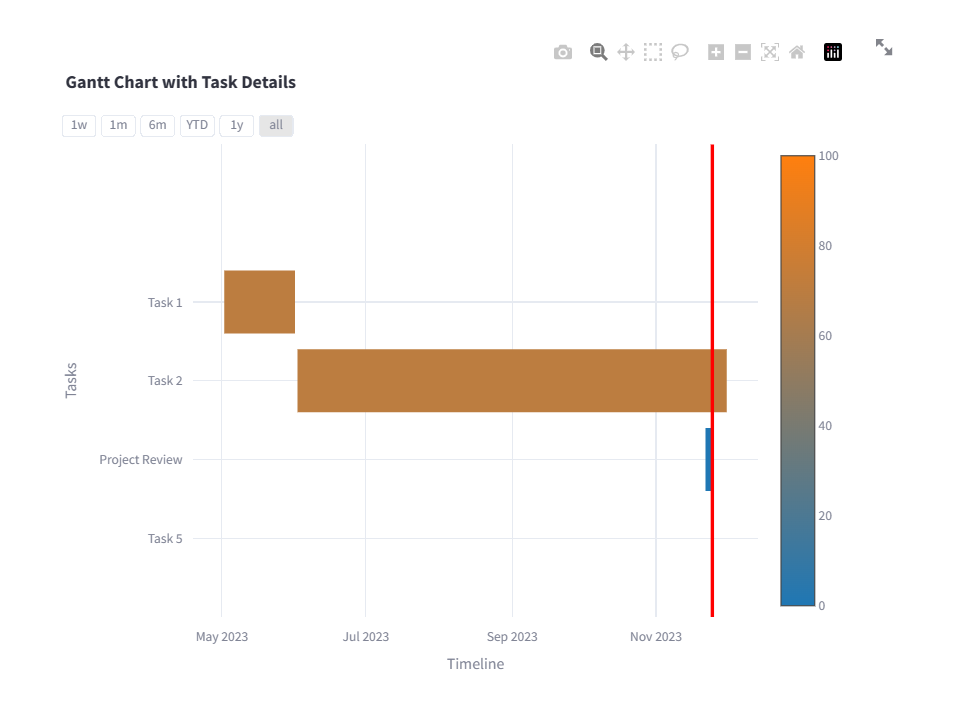
Bug Fixes and Updates

User Support and Feedback Analysis

1. **Do a rough estimate of effort required to accomplish each task in terms of person months.**

The general requirements of effort for each task is not huge enough if the development is carried out in a small scale and restricted only to local devices. However, if the app that we make to predict weather is to be hoisted on the web or is to be used for commercial purposes, then significant effort for interface dev is required. Can take roughly 2- 2.5 months.

1. **Create the Gantt Chart for scheduling using any tool**



**TEST PLAN DOCUMENT**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Name of Module** | **Test Case Description** | **Preconditions** | **Test Steps** | **Test Data** | **Expected Results** | **Actual Result** | **Test Results** |
| 1 | Alert menu | To provide alerts in case of severe weather | Access to Edge or Chrome Browser | Navigate to the server where the app is running.  Search for weather info of an area | Delhi | Severe stormy weather  Highly Humid conditions | Severe stormy weather  Highly Humid conditions | **Pass** |
| 2 | Alert menu | To provide an error alert in case of invalid area | Access to Edge or Chrome Browser | Navigate to the server where the app is running.  Search for weather info of an invalid area | xyz | Invalid Area | Valid Area, Gave some random weather | **Fail** |
| 3 | Alert menu | To provide an error alert in case of invalid area | Access to Edge or Chrome Browser | Navigate to the server where the app is running.  Search for weather info of an invalid area | xyz | Invalid Area | Invalid Area | **Pass** |
| 4 | Search Bar | To provide searching for weather info of a particular area | Access to Edge or Chrome Browser | Navigate to the server where the app is running.  Search for weather info of an area | Delhi | Temp: 19 deg | Temp: 19 deg | **Pass** |
| 5 | Weather API | To provide searching for weather info of a particular area | Access to Edge or Chrome Browser | Navigate to the server where the app is running. | Kolkata | Temp: 23 deg | Temp: 24 deg | **Fail** |
| 6 | Weather API | To provide searching for weather info of a particular area | Access to Edge or Chrome Browser | Navigate to the server where the app is running. | Banashankari | Temp: 19 deg | Temp: 23 deg | **Fail** |
| 7 | Weather API | To provide searching for weather info of a particular area | Access to Edge or Chrome Browser | Navigate to the server where the app is running. | Chennai | Temp: 28 deg | Temp: 28 deg | **Pass** |
| 8 | Weather API | To provide searching for weather info of a particular area | Access to Edge or Chrome Browser | Navigate to the server where the app is running. | Trivandrum | Temp: 28 deg | Temp: 28 deg | **Pass** |
| 9 | Weather API | To provide searching for weather info of a particular area | Access to Edge or Chrome Browser | Navigate to the server where the app is running. | Madrid | Temp: 16 deg | Temp: 16 deg | **Pass** |