**Title: Remote Weather Monitoring and Forecasting System**

* Abstract: Weather Reporting System is proposed to get live reporting of weather conditions on agriculture used regions. It must monitor temperature, humidity, wind, on large ground fields. The system is aimed to offer an economically efficient solution to monitor weather conditions.
* **INTODUCTION**: The Weather Monitoring System aims to provide users with accurate and up-to-date weather information based on their location or specified location. The system utilizes a weather API to fetch real-time weather data and presents it to users in a user-friendly interface.

• **Features**: A weather app that shows the weather at current location and allows the user to search location of a particular location, following are the features:

Notification System:

Implement a notification system using platform-specific APIs (such as Firebase Cloud Messaging for Android or Apple Push Notification Service for iOS) to send weather alerts to the user's device.Handle user preferences for receiving weather alerts in the app's settings.

Location-based Weather Forecast: Users can receive weather updates based on their current location or by manually entering a location.

Real-time Data: It fetches real-time weather data from a weather API to ensure accuracy and relevance.

Forecast: Users can view weather forecasts for the upcoming hours or days, helping them plan their activities accordingly.

Customization: Users can customize units (e.g., temperature in Celsius or Fahrenheit) and display preferences to suit their preferences.

Alerts and Notifications: It can send alerts and notifications for severe weather conditions, keeping users informed and safe.

* **Technology:**

1. HTML (Hypertext Mark-up Language): HTML is used for structuring the content, including elements such as headers, paragraphs, lists, and links. It provides the foundation for the app's layout and user interface.
2. CSS (Cascading Style Sheets): CSS is used for styling the HTML elements, controlling aspects such as colours, fonts, spacing, and layout. It's essential for creating visually appealing and responsive designs in the app.
3. JavaScript: JavaScript is used for adding interactivity and dynamic behaviour to the it. In this, JavaScript would be used to fetch weather data from APIs, manipulate the DOM (Document Object Model) to update the UI based on the retrieved data, handle user interactions, and implement features such as animations and transitions.
4. API Integration: The system utilizes a weather API to fetch weather data, including current conditions, forecasts, and historical data. Weather data APIs provide the app with access to weather information such as current conditions, forecasts, and historical data. JavaScript is used to make HTTP requests to these APIs, retrieve the data in JSON format, and process it for display in the app.
5. User Interface: The user interface is designed to be intuitive and userfriendly, with easy navigation and clear presentation of weather information.

Home Screen:

Current location weather display Search bar for searching weather in other locations Toggle button for weather alerts

Settings Screen:

Toggle button for enabling/disabling weather alerts. Other settings such as units (e.g., Celsius/Fahrenheit, km/h or mph for wind speed, etc.)

Weather Display:

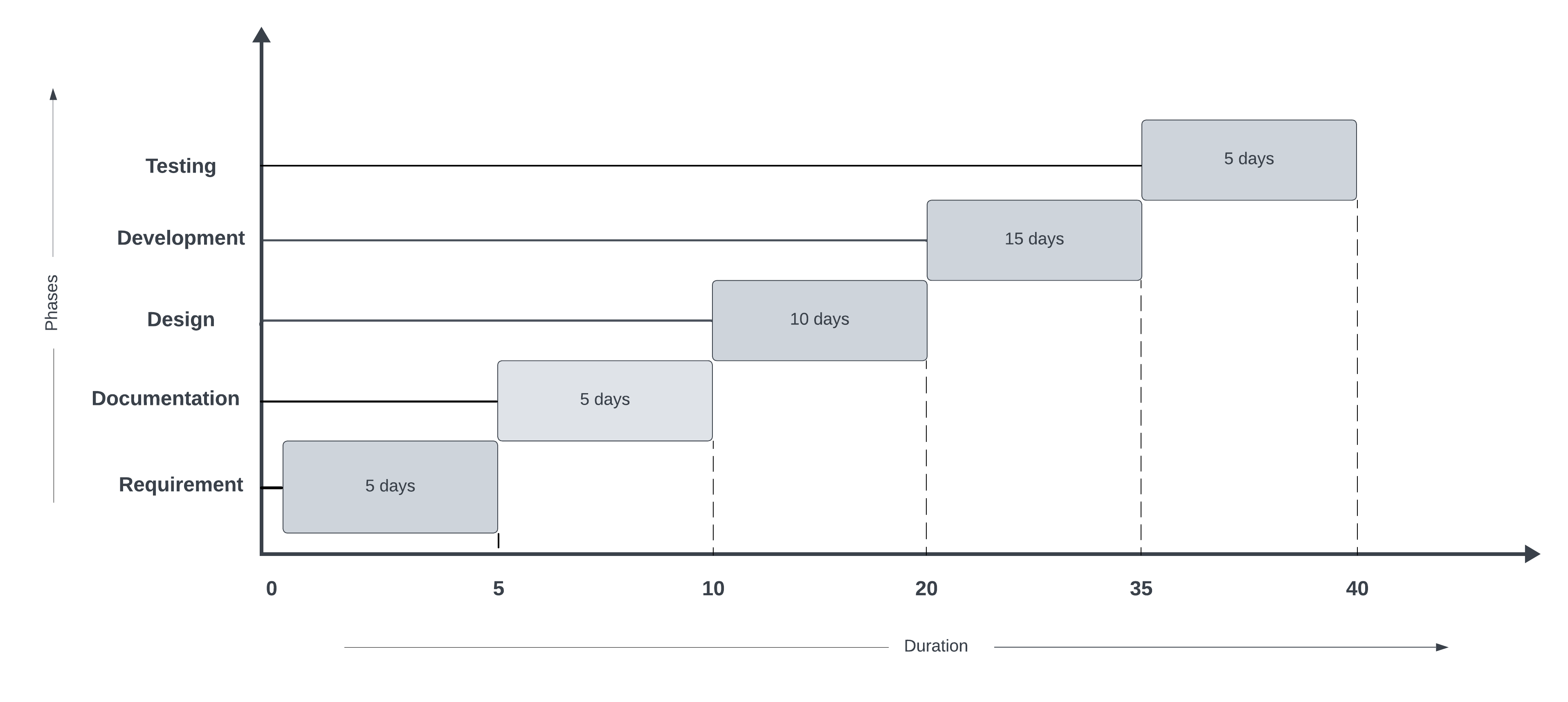
Utilize APIs such as OpenWeatherMap or Weathers tack to fetch weather data. Display current weather conditions for the user's current location.

Allow users to search for weather conditions in other locations by entering the location name or zip code. Display detailed weather information such as temperature, humidity, wind speed, precipitation, etc.

Weather Alerts:

Implement a notification system to push weather alerts to the user at the start of the day. Weather alerts could include severe weather warnings (e.g., storms, hurricanes, heavy rainfall, etc.) or other significant weather events.Allow users to customize which types of weather alerts they want to receive notifications for.

* **Timeline:**



* **Mentor:**

**NAME**: Govind Attarde

**Contact no**.: 7588624436

**Profession**: Professor at Bharti Vidyapeeth Jawaharlal Nehru Institute of Technology

* **Roles & Responsibility:**

1. Ishita Kadam: UI/UX Desing and Documentation work
2. Pradnya Gore: Frontend & layout creation (HTML, CSS and frontend framework like bootstrap)
3. Vedant Banaitkar: Backend (JS & API)

* **Further Enhancement:**

Geolocation-based Weather Alerts: Implement geolocation-based weather alerts, where users receive notifications based on their current location's weather conditions without manually entering a location. This provides personalized alerts and enhances user experience.

Interactive Weather Maps: Integrate interactive weather maps using APIs like Leaflet or Google Maps to visualize weather conditions, forecasts, and patterns. Users can zoom in/out, pan, and overlay different weather data layers for a better understanding of the weather in their area.

Localized Weather Forecasts: Improve localization by providing weather forecasts and alerts tailored to specific regions, cities, or neighborhoods. This enhances the app's usefulness for users in different geographical locations.

Multi-platform Support: Expand the app's availability by developing versions for multiple platforms, such as iOS, Android, desktop, and wearable devices. This allows users to access weather information from their preferred devices seamlessly.