A PROJECT REPORT ON

(Task Management and Pop-Up Reminder System)

BY

Tanaya Prashant Bhavsar

(SP47230018)

IN PARTIAL FULFILLMENT OF

M.Sc.(Computer Science) Part II, Sem IV

SIR PARASHURAMBHAU COLLEGE, PUNE

(2024 - 2025)

Shikshana Prasaraka Mandali's

SIR PARASHURAMBHAU COLLEGE

(EMPOWRED AUTONOMOUS)

TILAK ROAD, PUNE – 411 030.

Department of Computer Science

Certificate

This is to certify that the work in the entitled project of

____

*has been carried out by Mr./Mrs./Ms.*________________________

as the partial fulfillment of the requirements of the M.Sc. (Computer Science) Part II, MSCOSDSC401 - Industrial training at

____

for the last semester starting from _________ *20*

Date : / /20

Head

Teacher Incharge *Department of Science Computer*

1. Internal Examiner *2. External Examiner*

INDEX

ACKNOWLEDGEMENT

I take this opportunity to express my heartfelt gratitude to all those who have supported me directly or indirectly in the successful completion of my project titled **"Task Management and Pop-Up Reminder System"**, undertaken as part of the requirements for the degree of **M.Sc. Computer Science** at **Sir Parshurambhau College, Pune**.

First and foremost, I am deeply grateful to the Almighty for blessing me with the strength, patience, and determination to carry out this project.

I sincerely thank **Mr. Swami Panjala**, my project guide at **Elite Softwares**, for his expert guidance, constructive feedback, and consistent encouragement throughout the development of this project. His mentorship played a vital role in enhancing my practical skills and understanding of full-stack web development.

I extend my special thanks to **Ms. Kirti Garud**, the IT project coordinator at Sir Parshurambhau College, for her support and for providing me with the opportunity to work on this project under academic supervision.

I would also like to express my gratitude to the faculty members of the **Department of Computer Science** at Sir Parshurambhau College, Pune, for their continuous support, knowledge sharing, and motivation during the academic journey.

Finally, I would like to thank my family and friends for their unwavering support, encouragement, and motivation throughout the course of this project.

Thank you all.

INTRODUCTION TO PROBLEM

In today's fast-paced digital world, managing daily tasks and remembering important events has become increasingly challenging. With the growing demands of academic, professional, and personal commitments, individuals often find it difficult to keep track of pending work, deadlines, and special occasions such as birthdays, anniversaries, or appointments. The lack of an efficient system to organize tasks and send timely reminders often leads to missed deadlines, forgotten events, and decreased productivity.

While various task management tools exist, many of them lack integrated reminder systems or user-friendly interfaces that cater to both task tracking and event notifications. Moreover, most solutions are either too complex for everyday users or too limited in functionality for users seeking more control and customization.

The absence of a simple, intuitive, and interactive system that combines task organization with pop-up reminders creates a gap that this project aims to address. Users require a tool that not only helps them manage tasks with features like categorization, priority setting, and due dates but also alerts them with engaging and timely notifications for upcoming events.

Hence, the problem statement revolves around the need for a **comprehensive solution** that allows users to:

- Create, edit, and manage daily tasks efficiently
- Set deadlines and priorities for better task tracking
- Receive real-time pop-up alerts for important events
- Customize reminders based on preferences
- Enjoy a visually appealing and user-friendly interface

The **Task Management and Pop-Up Reminder System** was conceptualized to resolve these pain points by providing a unified platform where users can stay organized, productive, and mindful of important occasions—all within a responsive and accessible web application.

EXISTING SYSTEM

Currently, various task management applications like **Google Tasks**, **Microsoft To-Do**, **Trello**, and **Asana** offer basic functionalities such as creating tasks, setting due dates, and marking completion. However, most of these tools either:

- Lack **personalized reminder pop-ups** for special events like birthdays or anniversaries
- Require **internet connectivity or third-party integrations** to send notifications
- Do not allow users to **customize** the type, timing, or tone of alerts
- Are not built with **simplified interfaces** for non-technical users
- Offer **limited control over data privacy**, especially in free versions
- Are mostly used in a **corporate setting** and are less suited for individual users or students

Moreover, many users find these platforms too complex or overloaded with features they don’t need, especially when they simply want a compact solution to manage their daily life and personal reminders.

SCOPE OF WORK

The **Task Management and Pop-Up Reminder System** aims to bridge the gap by delivering a **lightweight**, **user-centric**, and **feature-rich** application suitable for students, individuals, and small teams. The scope includes:

✅ **Task Management:**

- Allow users to **create**, **read**, **update**, and **delete** tasks (CRUD operations)
- Enable **priority setting**, **due dates**, and **status tracking**
- Provide filters to sort tasks by urgency, category, or deadline

✅ **Pop-Up Reminder System:**

- Implement browser-based **real-time pop-up notifications**
- Support reminders for **recurring events** like birthdays and anniversaries
- Offer **default and custom images/sounds** for event types

✅ **User Experience (UX):**

- Design a clean, responsive, and intuitive interface using **Bootstrap**
- Use **color-coded visuals** to indicate deadlines and task importance
- Include **interactive calendar** to view events and task status

✅ **Customization & Accessibility:**

- Allow users to set **alert frequency**, **event repetition**, and **reminder tones**
- Support basic **multi-user functionality** and role management (Admin/User)
- Ensure **cross-device accessibility** via responsive design

✅ **Technology Stack Implementation:**

- **Frontend**: HTML5, CSS3, Bootstrap, JavaScript
- **Backend**: Python with Django Framework
- **Database**: SQLite
- **Tools**: Visual Studio Code, Django Admin Panel

This system provides a **smart, efficient, and elegant** solution for managing everyday responsibilities while never missing a celebration or commitment.

FEASIBILITY STUDY

A feasibility study assesses the practicality and viability of developing and deploying the proposed system. The goal is to ensure that the project is achievable within the defined constraints—technical, economic, operational, and legal.

1. Technical Feasibility

- The system is developed using widely adopted, open-source technologies such as **Python**, **Django**, **HTML5**, **CSS3**, **JavaScript**, **Bootstrap**, and **SQLite**.
- The tools and platforms used (e.g., **Visual Studio Code**, **Django Admin**) are compatible with most modern operating systems and are accessible to developers.
- Browser-based **pop-up alerts** and real-time task interaction are technically feasible using JavaScript.
- No need for specialized hardware—runs on any standard computer with a browser.

➞ **Conclusion**: Technically feasible due to the availability of reliable, scalable, and user-friendly technology.

2. Economic Feasibility

- The technologies used are **free and open-source**, eliminating licensing costs.
- No third-party services are required, minimizing operational expenses.
- Development, testing, and deployment can be done with minimal infrastructure and cost.

➞ **Conclusion**: Economically feasible, especially for educational institutions, students, and small organizations.

3. Operational Feasibility

- The application is **easy to use**, with an intuitive UI and minimal training required.
- Provides **real-world benefits** by helping users organize their tasks and remember important dates.
- Addresses a common operational need for individuals, students, and small teams.

- Works effectively in a **single-user or small multi-user environment**.

➞ **Conclusion**: Operationally feasible; the system supports the desired workflow and improves productivity.

4. Legal Feasibility

- The system uses only open-source technologies and frameworks, avoiding any legal complications.
- It handles user data securely and does not involve sensitive information requiring compliance with complex data protection laws (e.g., GDPR).

➞ **Conclusion**: Legally feasible and safe for personal, educational, or small business use.

5. Schedule Feasibility

- The project was successfully completed within the allocated time frame:

  📅 **6th January 2025 – 6th June 2025**.

- Planned modules such as task management, pop-up reminder logic, UI/UX design, and testing were completed on time.

➞ **Conclusion**: The timeline for development was realistic and achievable.

SYSTEM REQUIREMENT

The system requirements define the necessary hardware and software configurations needed to develop, deploy, and use the application effectively. The requirements are categorized into functional and non-functional needs, as well as hardware and software specifications.

1. Functional Requirements

These are the specific features and functionalities the system must provide:

Requirement ID	**Description**
FR-01	Users should be able to **create, read, update, and delete (CRUD)** tasks.
FR-02	System must support **setting deadlines, priorities**, and **categories** for tasks.
FR-03	Application must send **browser-based pop-up reminders** for events like birthdays and meetings.
FR-04	Users should be able to **schedule reminders** (daily, weekly, custom).
FR-05	System must provide a **dashboard** showing active tasks and upcoming reminders.
FR-06	Users must be able to **customize** notification tone, alert type, and display time.
FR-07	Tasks and reminders should be **searchable and filterable** by date or category.

2. Non-Functional Requirements

These are quality attributes that the system must meet:

Requirement ID	**Description**
NFR-01	The system should have a **responsive and user-friendly UI** using Bootstrap.
NFR-02	The application should be **platform-independent** and work across major browsers.
NFR-03	It must be **lightweight** and optimized for performance.
NFR-04	Data should be stored and retrieved securely using Django’s ORM.
NFR-05	Should support **scalability** for future multi-user or team collaboration features.

✅ 3. Software Requirements

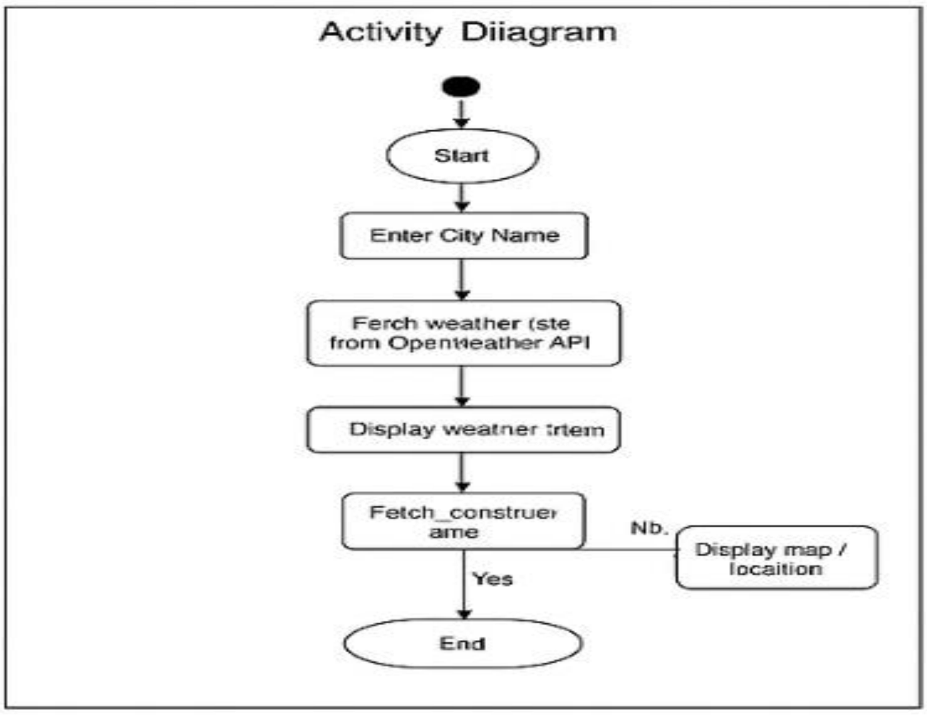
Software Component	**Version / Description**
Operating System	Windows 10 or higher / Linux / macOS
Backend	Python 3.x, Django Framework
Frontend	HTML5, CSS3, JavaScript, Bootstrap 4/5
Database	SQLite (default Django database)
IDE / Code Editor	Visual Studio Code
Browser	Google Chrome, Mozilla Firefox, Edge (latest versions)
Additional Tools	Django Admin, Git (optional)

✓ 4. Hardware Requirements

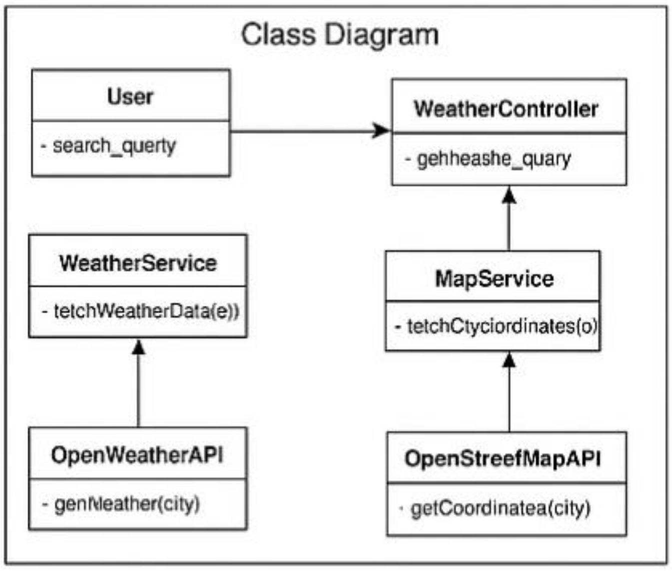
Component	**Minimum Requirement**
Processor	Intel Core i3 or equivalent
RAM	4 GB (8 GB recommended)
Storage	At least 1 GB free space
Display	13" screen with 1366x768 resolution (responsive for mobile too)
Internet	Required for initial setup and testing

UML DIAGRAMS

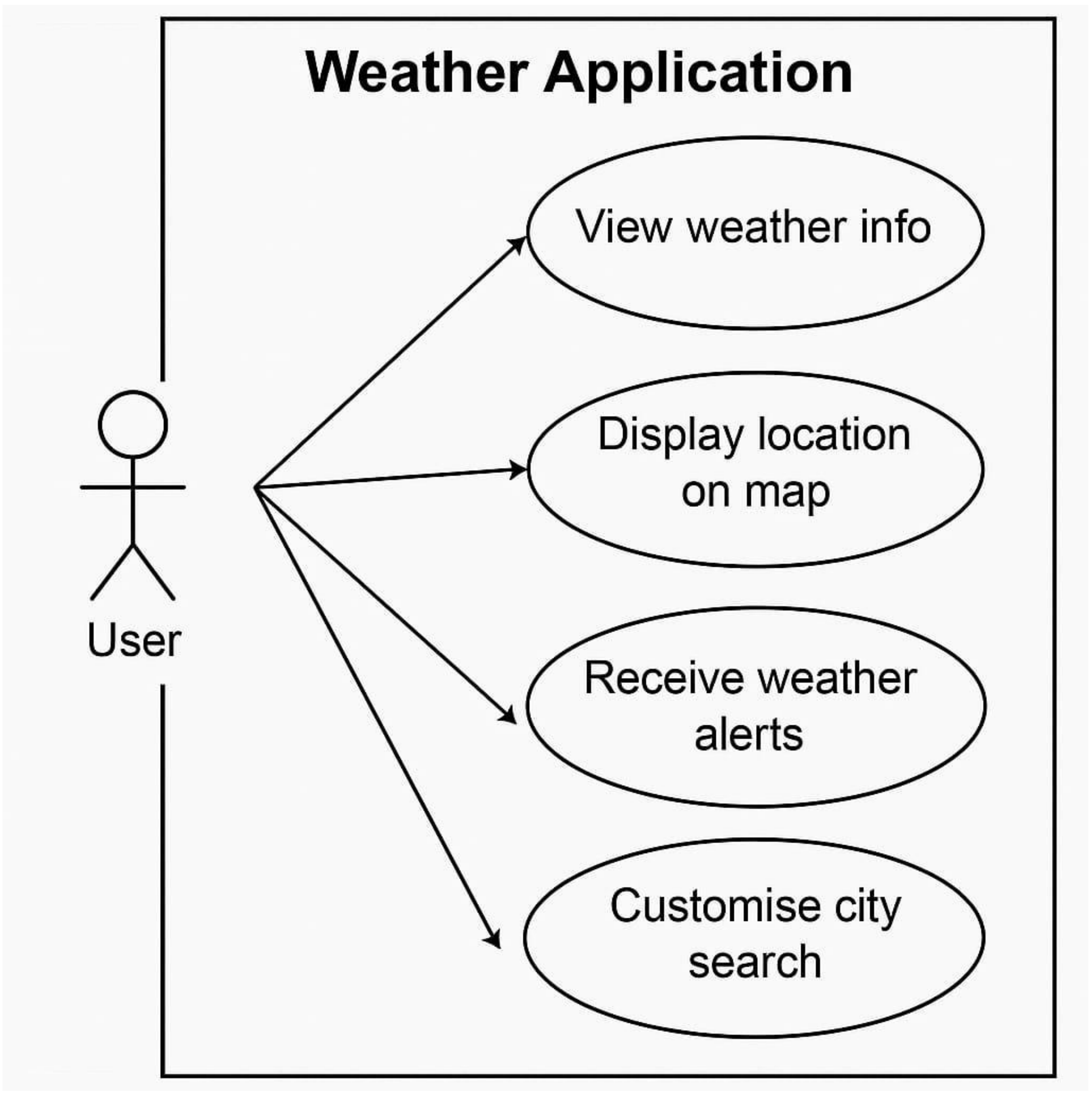
- **Activity Diagram**



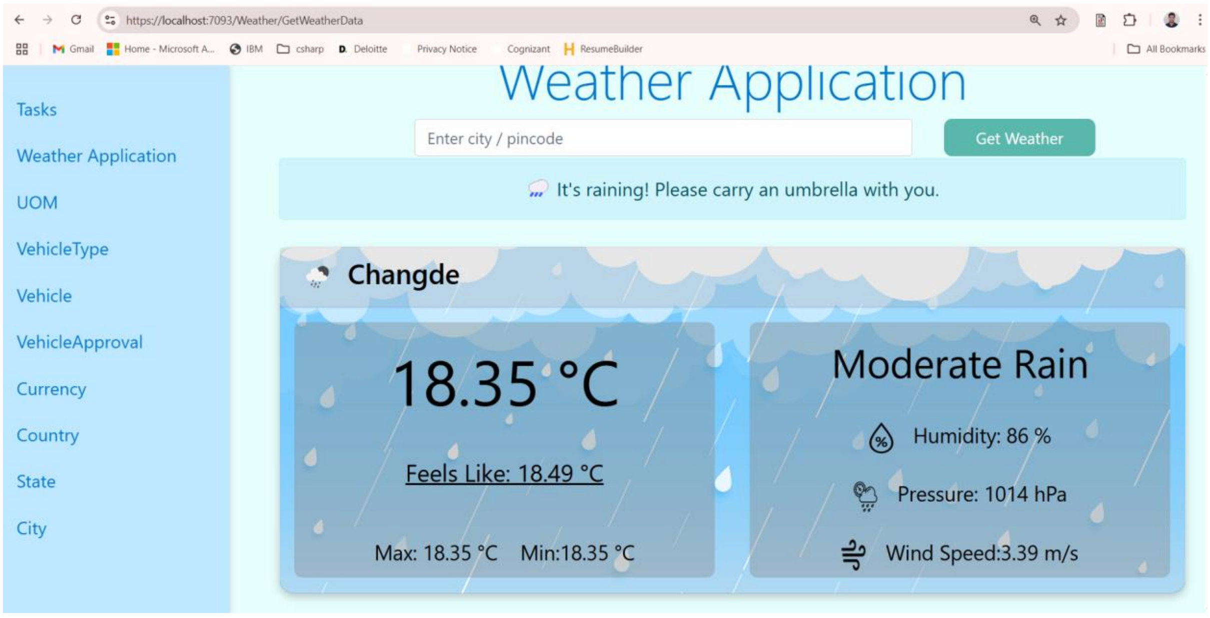
- **Class Diagram**



- **Use Case**



USER INTERFACE


https://localhost:7093/Weather/GetWeatherData

Tasks
Weather Application
UOM
VehicleType
Vehicle
VehicleApproval
Currency
Country
State
City

Weather Application

Enter city / pincode | Get Weather

It's raining! Please carry an umbrella with you.

Changde

18.35 °C

Feels Like: 18.49 °C

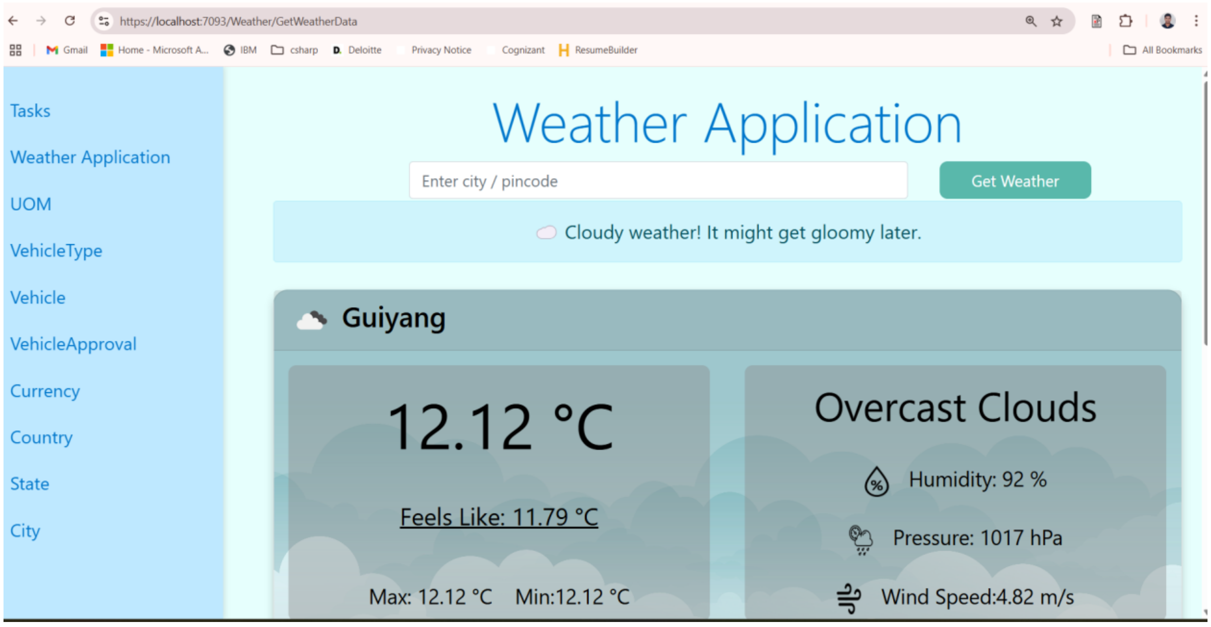
Max: 18.35 °C Min:18.35 °C

Moderate Rain

Humidity: 86 %

Pressure: 1014 hPa

Wind Speed:3.39 m/s


https://localhost:7093/Weather/GetWeatherData

Tasks
Weather Application
UOM
VehicleType
Vehicle
VehicleApproval
Currency
Country
State
City

Weather Application

Enter city / pincode | Get Weather

Cloudy weather! It might get gloomy later.

Guiyang

12.12 °C

Feels Like: 11.79 °C

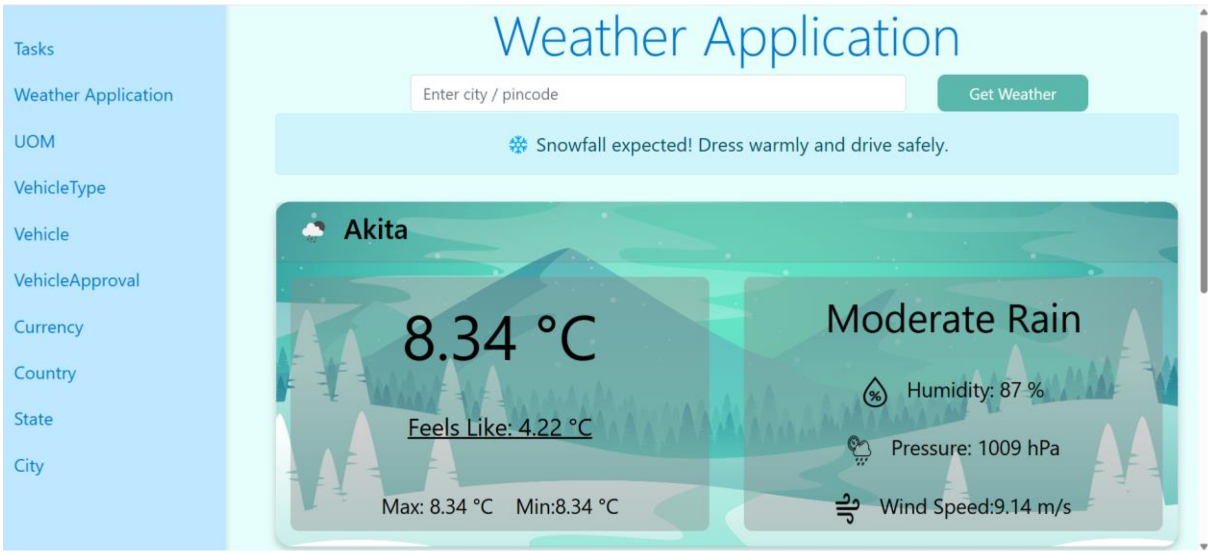
Max: 12.12 °C Min:12.12 °C

Overcast Clouds

Humidity: 92 %

Pressure: 1017 hPa

Wind Speed:4.82 m/s



Weather Application

❄ Snowfall expected! Dress warmly and drive safely.

Akita

8.34 °C

Feels Like: 4.22 °C

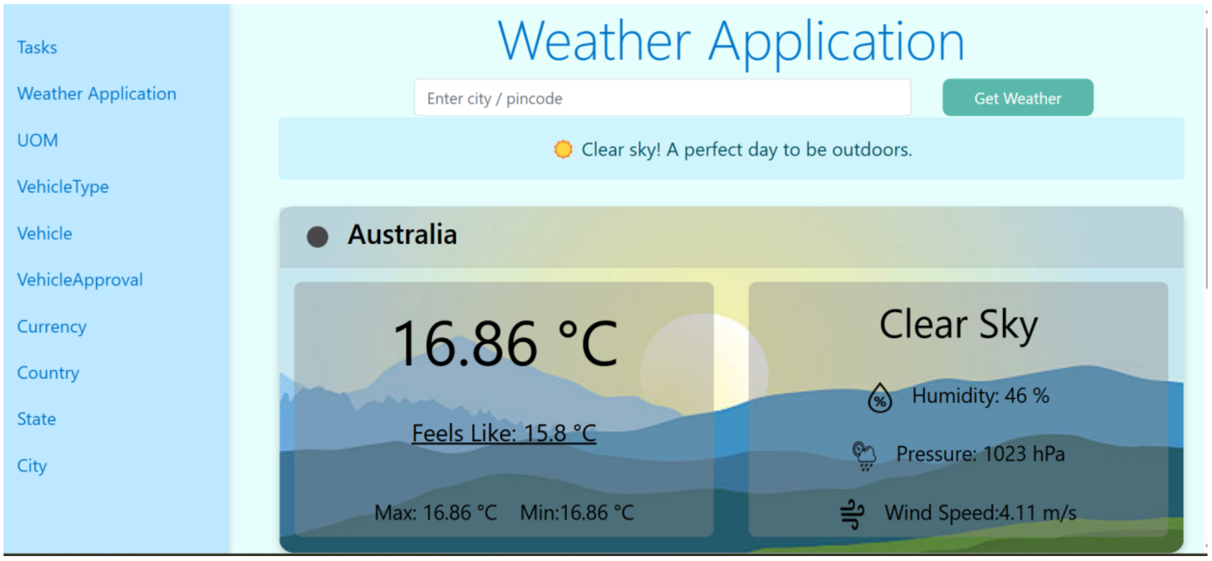
Max: 8.34 °C Min:8.34 °C

Moderate Rain

Humidity: 87 %

Pressure: 1009 hPa

Wind Speed:9.14 m/s



Weather Application

Clear sky! A perfect day to be outdoors.

Australia

16.86 °C

Feels Like: 15.8 °C

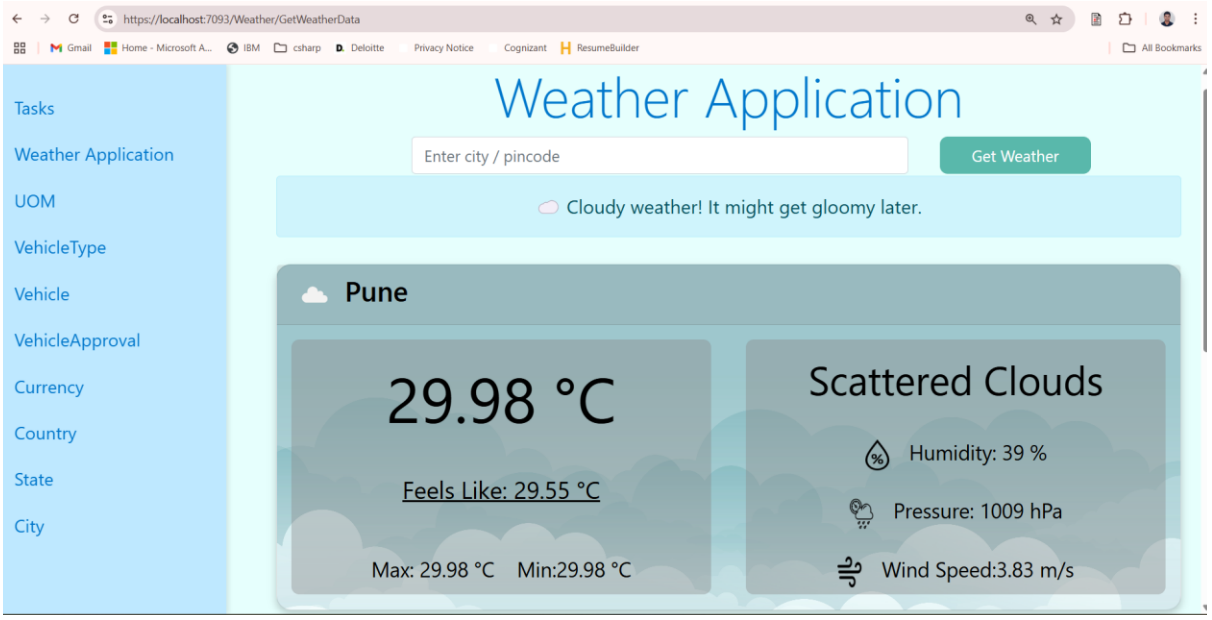
Max: 16.86 °C Min:16.86 °C

Clear Sky

Humidity: 46 %

Pressure: 1023 hPa

Wind Speed:4.11 m/s



Weather Application

Cloudy weather! It might get gloomy later.

Pune

29.98 °C

Feels Like: 29.55 °C

Max: 29.98 °C Min:29.98 °C

Scattered Clouds

Humidity: 39 %

Pressure: 1009 hPa

Wind Speed:3.83 m/s

TESTING STRATEGIES

Testing strategies were essential for ensuring the **Banking ERP System** functioned as expected and met all quality standards. The primary testing strategies used in the project included:

1. **Manual Testing**: This approach was used to identify bugs and defects in the system by executing test cases manually. Test cases were designed based on functional requirements for modules such as **Customer Management**, **Account Handling**, and **Transactions**. It helped identify critical issues like incorrect data processing or broken features.
2. **Unit Testing**: Each individual module was tested for its correctness by isolating and verifying specific functions or components. This ensured that each part of the system was performing as expected before integration.
3. **Integration Testing**: After unit testing, **integration testing** was performed to ensure that different modules within the system interacted correctly, especially for workflows like transaction processing and report generation.
4. **Regression Testing**: This type of testing ensured that new changes or fixes did not negatively affect existing functionalities. It was performed regularly as updates and bug fixes were applied.
5. **User Acceptance Testing (UAT)**: UAT involved testing the system with end-users to verify that it met their expectations and requirements. It was crucial to ensure that the system was user-friendly and aligned with business needs.
6. **Security Testing**: Security testing focused on identifying vulnerabilities such as unauthorized access or data leaks. It aimed to ensure the system’s data protection measures were effective and met industry standards.

These testing strategies collectively helped to ensure the system's functionality, security, and usability.

TEST CASES

Test Case ID	Test Scenario	Test Steps	Expected Result	Actual Result	Status
TC_01	Search for weather by valid city	1. Enter a valid city name (e.g., Pune) 2. Click “Search”	Weather details for Pune should be displayed	Result displayed	Pass
TC_02	Search with empty input	1. Leave the city input field blank 2. Click “Search”	System should show an error message like “City name required”	Displayed error message	Pass
TC_03	Invalid city name	1. Enter an invalid/non-existent city (e.g., xyz123 2.)2. Click “Search”	Application should show “City not found” or a similar alert	Displayed error message	Pass
TC_04	Check API integration	1. Enter a valid city 2. Capture API call	Weather data should be correctly fetched from the OpenWeather API	API error	Pass
TC_05	Display map location	1. Search a valid city 2. Check map display	OpenStreetMap should load showing the searched city	Displayed map	Pass

LIMITATIONS

While the **Banking ERP System** project achieved its key objectives, several limitations impacted its scope and implementation:

1. **Time Constraints**: Due to the limited duration of the internship, certain features, such as **automation testing** and **advanced performance testing**, were not fully implemented.
2. **Manual Testing**: The project primarily relied on **manual testing**, which is time-consuming and prone to human error, limiting the speed and accuracy of bug identification and resolution.
3. **Lack of Mobile Integration**: While the system could benefit from a **mobile application**, this feature was not developed due to time and resource limitations.
4. **Scalability Concerns**: The system was not fully tested under heavy load or stress conditions, which could impact its performance as the number of users increases.
5. **Security Enhancements**: While basic security measures were implemented, advanced features such as **multi-factor authentication (MFA)** and **encryption for sensitive data** were not fully integrated.

These limitations are areas of improvement for future development of the system.