VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JNANA SANGAMA", BELGAUM-590014, KARNATAKA

An Internship Project Report on

"Smart Weather Forecasting System"

Submitted in partial fulfilment of the requirement for the VII semester course of

BACHELOR OF ENGINEERING In

COMPUTER SCIENCE AND ENGINEERING

Submitted By

Alfiya Hassan	**1AP21CS006**
Soumya S Patil	**1AP21CS009**
Deeksha N	**1AP22CS401**

Under the guidance of

Akhil Sai

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

APS COLLEGE OF ENGINEERING

Anantha Gnana Gangothri

26thk.m, N.H-209, Kanakapura Road, Bangalore–560 082

Project Completion Certificate

I, **Alfiya Hassan** (Roll No: 1AP21CS006), hereby declare that the material presented in the Project Report titled **"Smart Weather Forecasting"** represents original work carried out by me in the **Department of Computer Science** at the **APS college of Engineering, Bangalore** during the tenure **2 October, 2024 – 12, December, 2024**.

With My signature, I certify that:

- I have not manipulated any of the data or results.
- I have not committed any plagiarism of intellectual property and have clearly indicated and referenced the contributions of others.
- I have explicitly acknowledged all collaborative research and discussions.
- I understand that any false claim will result in severe disciplinary action.
- I understand that the work may be screened for any form of academic misconduct.

Date: **Student Signature**:

In my capacity as the supervisor of the above-mentioned work, I certify that the work presented in this report was carried out under my supervision and is worthy of consideration for the requirements of the B.Tech. Internship Work.

Advisor's Name: Dr Shivamurthaiah M **Guide Name:** Akhil Sai

Advisor's Signature **Guide Signature**

Project Completion Certificate

I, **Soumya S Patil** (Roll No: 1AP21CS044), hereby declare that the material presented in the Project Report titled **"Smart Weather Forecasting"** represents original work carried out by me in the **Department of Computer Science** at the **APS college of Engineering, Bangalore** during the tenure **2 October, 2024 – 12, December, 2024**.

With My signature, I certify that:

- I have not manipulated any of the data or results.
- I have not committed any plagiarism of intellectual property and have clearly indicated and referenced the contributions of others.
- I have explicitly acknowledged all collaborative research and discussions.
- I understand that any false claim will result in severe disciplinary action.
- I understand that the work may be screened for any form of academic misconduct.

Date: **Student Signature**:

In my capacity as the supervisor of the above-mentioned work, I certify that the work presented in this report was carried out under my supervision and is worthy of consideration for the requirements of the B.Tech. Internship Work.

Advisor's Name: Dr Shivamurthaiah M **Guide Name:** Akhil Sai

Advisor's Signature **Guide Signature**

Project Completion Certificate

I, **Deeksha N** (Roll No: 1AP22CS401), hereby declare that the material presented in the Project Report titled **"Smart Weather Forecasting"** represents original work carried out by me in the **Department of Computer Science** at the **APS college of Engineering, Bangalore** during the tenure **2 October, 2024 – 12, December, 2024**.

With My signature, I certify that:

- I have not manipulated any of the data or results.
- I have not committed any plagiarism of intellectual property and have clearly indicated and referenced the contributions of others.
- I have explicitly acknowledged all collaborative research and discussions.
- I understand that any false claim will result in severe disciplinary action.
- I understand that the work may be screened for any form of academic misconduct.

Date: **Student Signature**:

In my capacity as the supervisor of the above-mentioned work, I certify that the work presented in this report was carried out under my supervision and is worthy of consideration for the requirements of the B.Tech. Internship Work.

Advisor's Name: Dr Shivamurthaiah M **Guide Name:** Akhil Sai

Advisor's Signature **Guide Signature**

Evaluation Sheet

Title of the Project: Smart Weather Forecasting

Name of the Students: Alfiya Hassan
Soumya S Patil
Deeksha N

External Supervisor:

_____

Internal Supervisor:

_____

_____

Date:

Place:

1.ABSTRACT

The Smart Weather Forecasting System is a weather-monitoring application designed to provide real-time weather data through a user-friendly web interface. Using a Raspberry Pi as the processing unit, the system integrates sensors or APIs to collect weather information and displays it on a responsive dashboard. With automated updates and a visually appealing design, this project aims to enhance weather awareness for users, providing critical metrics such as temperature, humidity, wind speed, and condition descriptions. The proposed system addresses common challenges faced by college libraries, including long queues, manual book tracking, and difficulty in managing reservations. By leveraging technology, it provides students and faculty with a user-friendly platform to browse, reserve, and track the availability of library books remotely.

Weather plays a crucial role in daily activities, and having accurate, real-time weather data is essential for effective decision-making. The Smart Weather Forecasting System is a cutting-edge project designed to provide continuous weather updates via a web-based dashboard. This system leverages a Raspberry Pi as the control unit and combines it with either weather sensors or an external weather API for data collection. The fetched data is processed and displayed dynamically, ensuring users receive timely and accurate information.

The project integrates advanced technologies, including a Flask backend and a responsive frontend built with HTML, CSS, and JavaScript. Key weather metrics such as temperature, humidity, wind speed, and conditions are visualized in an easy-to-read format. By automating data refresh and incorporating a visually appealing design, this system caters to individual users, industries, and public services that rely on real-time weather monitoring.

2.INTRODUCTION

2.1 Overview

Weather forecasting has become an indispensable tool for modern living, influencing decisions in agriculture, transportation, disaster management, and personal planning. Despite the abundance of weather forecasting services, many fail to deliver real-time updates or provide a user-friendly experience. The Smart Weather Forecasting System aims to bridge these gaps by offering a localized, interactive, and real-time solution.

This project uses a Raspberry Pi as its core processing unit to gather weather data either from connected sensors or through a weather API. A Flask-based backend processes the data, which is then displayed on a responsive web dashboard. Users can view critical weather parameters such as temperature, humidity, and wind speed in real-time, with updates occurring every two seconds. The intuitive interface, combined with automated updates, makes this system highly efficient and practical for various applications.

By leveraging the capabilities of the Raspberry Pi and modern web technologies, the system delivers accurate weather data seamlessly. Its potential applications extend beyond personal use, benefiting industries and organizations that rely on weather forecasting for operations and decision-making**.**

2.2 Objective

The main objective of this project is to develop a smart system that:

- Collects real-time weather data.
- Displays weather conditions dynamically on a web dashboard.
- Provides an interactive and automated user interface for continuous updates.
- Offers scalability for future enhancements, including advanced weather metrics.

2.3 Problem Statement

Accurate and timely weather information is essential for daily decision-making, from planning outdoor activities to ensuring safety during severe weather conditions. However, current solutions are often fragmented, inaccessible, or lack real-time data visualisation. This project addresses these gaps by creating a centralised, automated, and user-friendly weather forecasting system.

3. Application

The Smart Weather Forecasting System can be utilised in various fields, including:

1. **Agriculture**: Monitoring weather conditions to optimise irrigation and crop management.
2. **Transportation**: Providing real-time updates for safe navigation and scheduling.
3. **Disaster Management**: Alerting communities about adverse weather conditions.
4. **Personal Use**: Daily weather updates for individuals planning outdoor activities.
5. **Smart Homes**: Integrating the weather system with a smart home.
6. **Weather Stations**: Provides accurate, local weather data for both personal and community use.
7. **Urban Planning**: Helps urban planners in smart city initiatives by providing real-time environmental data.
8. **Climate Research**: Collecting long-term weather data for research in climate change and environmental science.

4. Components

Hardware

1. Raspberry Pi (as the central processing unit).
2. Power supply and network connectivity.
3. Buzzer.
4. Jumper wires and breadboard.
5. Micro SD Card (16GB or more).
6. External storage.

Software

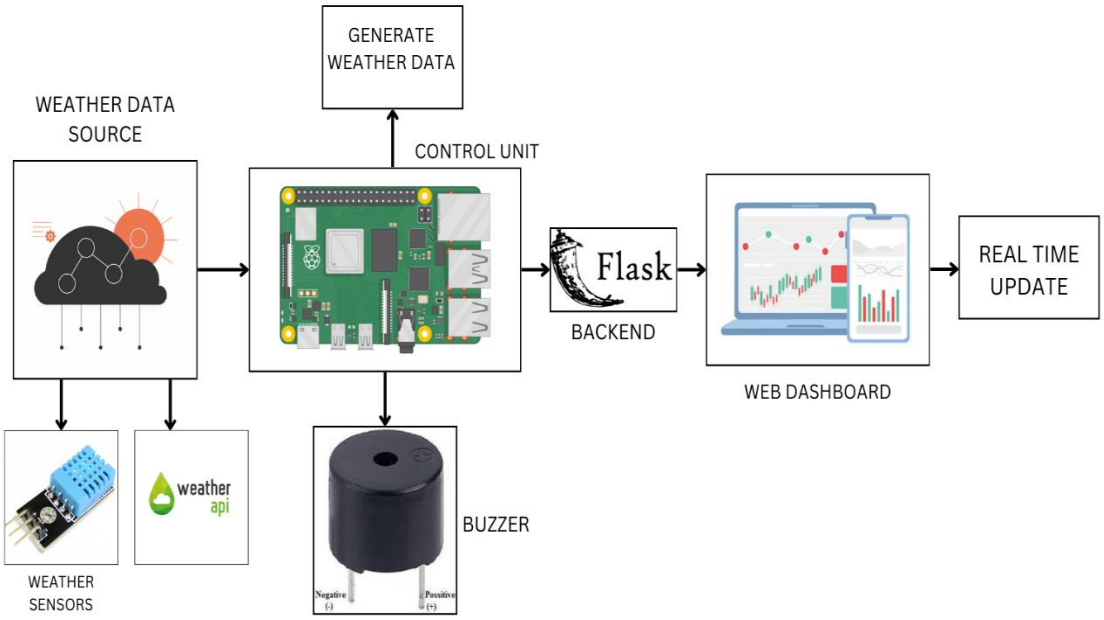
1. Python -Flask, Twilio RPi.GPIO libraries
2. HTML, CSS, JavaScript: Frontend for the web interface.
3. Weather API : For fetching external weather data.
4. Raspberry Pi OS.
5. Twilio Account -For SMS alert.
6. Code editor- VS Code or PyCharm .

Sensor

1. Temperature and humidity sensors.
2. Pressure sensors.
3. Light sensors.
4. Rain sensors.

5.Flowchart

Flowchart Process:



6. Conclusion

The Smart Weather Forecasting System effectively provides real-time weather data in a structured and user-friendly manner. With its automated updates and visually appealing design, it serves as a valuable tool for weather monitoring across diverse applications. The project demonstrates the seamless integration of hardware and software technologies, laying the foundation for future enhancements.

7.Future work

Enhanced Features: Add advanced metrics like air quality, precipitation probability, and UV index.

1. **Mobile Application:** Develop a mobile app for broader accessibility.
2. **Voice Assistance:** Integrate voice commands for weather queries using Raspberry Pi's GPIO capabilities.
3. **Predictive Analytics:** Incorporate machine learning models for weather predictions.
4. **Global Accessibility:** Deploy the system on cloud platforms for wider access beyond local networks.
5. **Advanced Metrics**: Incorporate features like air quality index, precipitation probability, and UV index.
6. **Mobile App**: Extend functionality to mobile platforms for wider accessibility.
7. **Machine Learning Integration**: Use predictive algorithms for advanced weather forecasting.
8. **Global Accessibility**: Deploy the system on a cloud-based platform to support users worldwide.

8.Appendix

A. System Architecture Overview

The system architecture of the Smart Weather Forecasting project consists of several key components:

IoT Device: This consists of a Raspberry Pi (RPi) with connected sensors such as temperature, humidity, and wind speed sensors. The Raspberry Pi is used to collect real-time data and pass it to the Flask backend via a REST API.

Backend (Flask Server): The Flask application serves as the server that provides weather data, forecasts, and SMS alerts. It fetches data from the Raspberry Pi or simulated weather data and sends it to the frontend.

Frontend (Web Dashboard): A website built using HTML, CSS, and JavaScript displays real-time weather data and a 7-day forecast. It fetches data from the Flask server via API calls and dynamically updates the page with the data.

External Services (Twilio): For SMS alerts, Twilio is used to send weather alerts (e.g., storm or heavy rain) to a mobile number.

B. Flask API Code Breakdown

The backend of the system is developed using Flask. Below is a breakdown of the Flask app code:

Import Dependencies: time, random, logging: Used for generating data and logging information.

Flask, Jsonify, request: From the Flask library to create and manage the web server and endpoints.

CORS: To enable cross-origin resource sharing for communication between the frontend (HTML/JS) and backend (Flask).

twilio.rest.Client: To interact with Twilio API for SMS sending.

RPi.GPIO: For controlling a buzzer connected to the Raspberry Pi GPIO pins.

Twilio Configuration:

TWILIO_ACCOUNT_SID, TWILIO_AUTH_TOKEN, TWILIO_PHONE_NUMBER, and TO_PHONE_NUMBER are configuration settings for sending SMS messages.

send_sms(message) is a function that sends an SMS with the provided message content to the predefined phone number.

GPIO Setup:

RPi.GPIO is used to set up a buzzer that will be activated when certain weather conditions like rain, storm, or strong winds are detected.

Weather Data Generation:

The system uses random values to simulate weather data for temperature, humidity, wind speed, and weather conditions (e.g., clear, cloudy, rainy).

The generate_weather_readings() function generates random values within normal and high ranges and logs them.

API Routes:

weather-data: This endpoint returns the current simulated weather data.

7-day-forecast: This endpoint returns a simulated 7-day weather forecast. Each day includes a temperature and weather condition.

C. Frontend (HTML, CSS, and JavaScript)

The frontend of the system is a dynamic webpage created using HTML, CSS, and JavaScript. Below is a breakdown of the frontend implementation:

HTML Structure:

The HTML contains a navigation bar, header, weather dashboard, and a table for the 7-day weather forecast.

Each weather parameter (e.g., temperature, humidity) is displayed inside a div element with class weather-box. The data is inserted dynamically using JavaScript.

CSS Styling:

The styling is designed to make the dashboard visually appealing with background images, hover effects, and responsive design.

A combination of fixed and relative positioning, padding, and font styling is used to ensure the page is well-organized and user-friendly.

JavaScript (AJAX requests):

Fetch Weather Data(): This function makes a request to the Flask /weather-data endpoint to get the current weather data and dynamically updates the HTML page.

fetchSevenDayForecast(): This function fetches data from the /7-day-forecast endpoint and populates the 7-day forecast table with the corresponding temperature and weather condition.

Automatic Update: The weather data is updated every 2 seconds using setInterval(fetchWeatherData, 2000) to show live weather updates.

D. Detailed Workflow

Weather Data Collection:

The IoT device (Raspberry Pi) collects weather data from sensors.

This data is either simulated or pulled from the device and sent to the Flask server.

API Calls:

The frontend makes requests to the Flask backend API to retrieve the current weather data and 7-day forecast.

When weather conditions such as storm, rain, or strong winds are detected, the server triggers SMS alerts via Twilio and activates the buzzer using the Raspberry Pi GPIO.

Displaying Data:

Once the weather data is received from the Flask API, it is displayed on the webpage under various sections (e.g., temperature, humidity).

The 7-day forecast table is populated dynamically with the forecast data.

Alerts:

The system sends an SMS notification if certain weather conditions (storm, rain, wind) are detected.

A buzzer connected to the Raspberry Pi is triggered to alert the user of impending adverse weather.

8.1 Pseudo code