# A Report on

# Weather App(MAUSAM)

**Submitted By:**

**Samkit Sanghvi (N019)**

**Kinjal Maheshwari (N022)**

**Vansh Israni (N025)**


**Shri Vile Parle Kelvani Mandal's**

**Mukesh Patel School of Technology and Management Engineering**

**Department of Computer Engineering**

**Vile Parle (w), Mumbai- 400056**

## Project Overview

**Mausam** is a simple weather application built using Flask (a Python web framework) that allows users to check the current weather conditions for any location using the WeatherAPI. Along with its main feature of weather information retrieval, the website includes a **FAQ** page, an **About** page, and a **Contact** page for customer support. The project also integrates a **form submission feature** using SQLite3 as the database, allowing users to submit their contact details and messages. The key objective of this project is to create an intuitive and user-friendly web application where users can check the weather and communicate with the website administrators via the form submission feature.

## Software Requirements

For the Mausam project, the following software and tools are used:

- **IDE:** Visual Studio Code (or any Python-supporting IDE)
- **Backend Framework:** Flask (Python-based web framework)
- **Database:** SQLite3 (for storing form submissions)
- **RESTful API:** WeatherAPI (for fetching real-time weather data)
- **Web Server:** Flask built-in development server
- **Authentication:** Flask-HTTPAuth (for basic authentication)
- **Template Engine:** Jinja2 (Flask's default template engine)
- **Session Management:** Flask-Session (for session management)
- **HTTP Requests:** requests library (for API calls)
- **Configuration:** A config.py file containing necessary API keys and configuration variables

# Web Pages Overview

 **Home Page ("/")**:

- **Purpose**: The homepage provides a brief introduction to the Mausam weather app and the company. Users can access the weather checking functionality and navigate to other pages.
- **Key Features**: Displays the company name dynamically and provides links to other pages (FAQ, About, Contact).

 **FAQ Page ("/faq/")**:

- **Purpose**: This page answers common questions that users might have about the weather app or other services offered by the website.
- **Key Features**: A simple, static page that loads common questions and answers dynamically using Jinja2 template rendering.

 **About Page ("/about/")**:

- **Purpose**: The about page shares information about the project or company behind Mausam.
- **Key Features**: Provides background information on the app's development and purpose, including the company name and other relevant details.

 **Contact Page ("/contact/")**:

- **Purpose**: Allows users to submit inquiries or messages through a contact form.
- **Key Features**: A form that collects user name, email, and message, and saves this data into an SQLite database. The form also displays a submission number upon successful form submission.

 **Weather Page ("/weather/")**:

- **Purpose**: The main feature of the website, where users can input a location and retrieve real-time weather data.

- **Key Features**: A form that takes in a user query (location), sends it to the WeatherAPI, and displays the current weather conditions. It also uses Google Maps API for displaying location-related data visually.

# Database and Diagrams

Number of tables featured in our project: 2

### Users Table

**Purpose**: Stores user login credentials for authentication.

**Attributes**:

- **id** (Primary Key): Unique identifier for each user, auto-incremented.
- **username**: Stores the username chosen by the user. Must be unique and is required for login.
- **password**: Stores the user's password. Required for authentication.

**Notes**: This table is used to authenticate users when they attempt to log in. It ensures secure access to other features of the app by requiring login credentials.

### Submissions Table

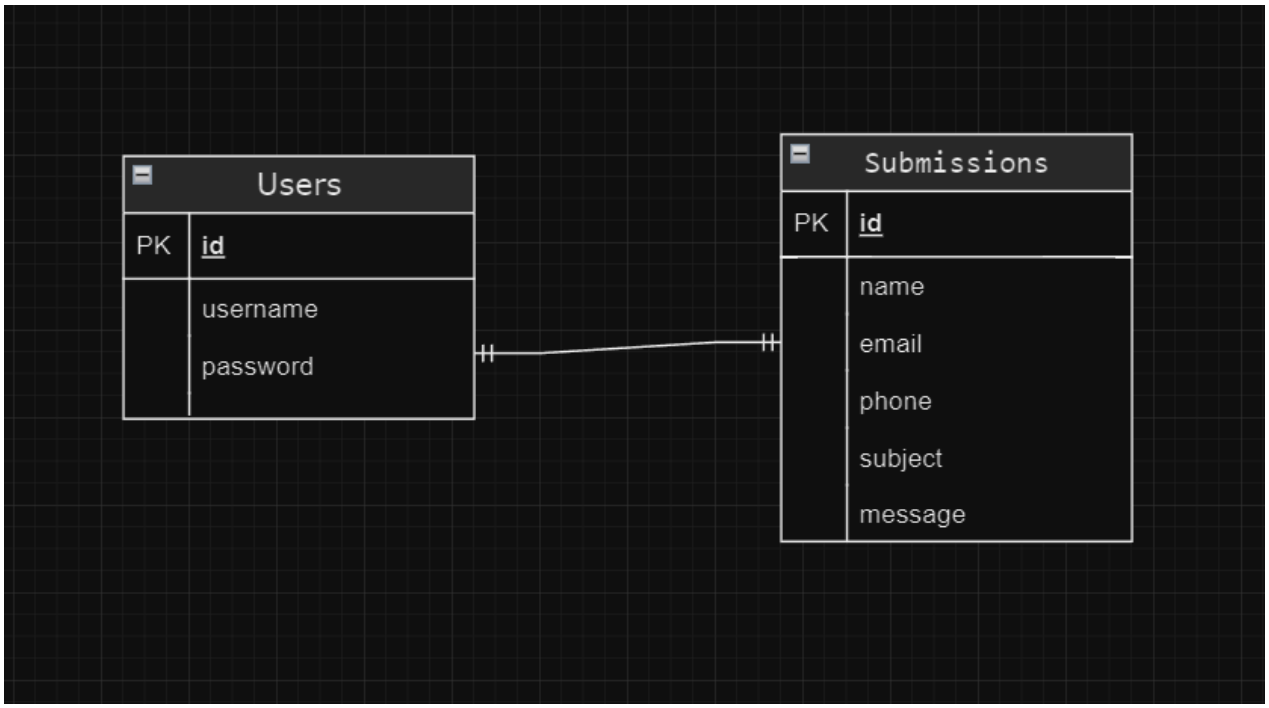
**Purpose**: Captures user submissions from the contact form on the website.

**Attributes**:

- **id** (Primary Key): Unique identifier for each form submission, auto-incremented.
- **name**: The name of the person submitting the form. Required field.
- **email**: Contact email of the submitter. Required field.
- **phone**: Contact phone number of the submitter. Optional field.
- **subject**: Subject line of the message.
- **message**: The actual message or inquiry submitted by the user. Required field.

This table helps store user inquiries and messages for future reference and response. Each submission is saved with a unique identifier for easy tracking.

# ENTITY - RELATIONSHIP DIAGRAM



# SEQUENCE DIAGRAM