EDGE PROJECT: Department of ICT, MBSTU

Deliverable

Project title: Weather forecast project

Submitted by:Mst.Ronita Akter

Supervisor Name: Dr.Ziaur Rohman

September 20, 2024

Date	Revision	Release Notes
20-09-2014	Rev 01	Initial Release

Contents

1 Abstract

The Weather Forecast Project using Django is a web application designed to provide accurate and up-to-date weather information for users. The system integrates with weather APIs to deliver real-time data such as temperature, humidity, wind speed, and forecasts for specific locations. By using Django as the backend framework, the application ensures efficiency, scalability, and easy management of data.

2 Introduction

Weather forecasting has become an essential part of modern life, helping individuals plan their daily activities and prepare for severe weather conditions. This project aims to create a user-friendly weather forecasting web application using Django. Django's Model-View-Template (MVT) architecture will be used to develop the application, with the help of third-party APIs like OpenWeatherMap or WeatherAPI to fetch real-time data.

Key Features:

- Real-time weather updates.
- Four-day weather forecasts.
- Location search to display weather data for specific regions.
- Simple, user-friendly interface.

3 Completion Plan

3.1 Phase 1:Project Setup

- Install Django.
- Set up the Django project and environment.

3.2 Phase 2: API Integration

- Connect to a weather API (like OpenWeatherMap or WeatherAPI).
- Fetch data from the API using Django's request system.

3.3 Phase 3: Database Design

- Create models for storing location data and user preferences (optional).
- Set up migrations and manage the database.

3.4 Phase 4:Front-End Development

- Build simple views using Django templates.
- Style the application with basic CSS and JavaScript for user interaction

3.5 Phase 5: Testing Debugging

- Test the API integration.
- Handle edge cases (invalid locations, no internet).

3.6 Phase 6: Deployment

- Deploy the application on a server (Heroku or AWS).
- Test the live application.

4 Proposed Application Details

This Django project will rely on several components:

- Backend (Django Framework): This will manage the routing, API calls, and database handling.
- Frontend (HTML/CSS/JavaScript): For rendering the weather data and providing user input forms.
- API Integration (Weather API): Real-time weather data will be fetched using an external API like OpenWeatherMap.
- Database (SQLite or PostgreSQL): To store search history and user preferences (if necessary).

5 Discussion

This project offers an excellent opportunity to combine both frontend and backend web development skills. Django's flexibility allows for rapid development and easy integration with third-party APIs. By focusing on a user-friendly interface, the application will be simple for end users, making it valuable for everyday weather checking. Additionally, the ability to scale this project in the future, such as adding user authentication or notifications, ensures its potential for long-term improvement.

5.1 Challenges:

- Dealing with API limits (most weather APIs have daily call limits).
- Handling large amounts of user traffic (optimization may be necessary).
- Ensuring the application is responsive and works across all devices.

6 Conclusion

The Weather Forecast Project using Django is a practical and achievable application that serves a vital purpose in daily life. With careful planning and execution, it will result in an effective and user-friendly platform for delivering real-time weather information. The use of Django simplifies backend management while providing flexibility for future enhancements, making this project a valuable learning experience and useful tool for users.