LOVELY PROFESSIONAL UNIVERSITY

Real Time Weather-Based Outfit Suggestion Platform

Course code:- **INT219/INT220**

Submitted to: **Ankit Saurav / Kuldeep Kumar Kushwaha**

Submitted by:

Name- **Md Arif Hussain**

Registration Number- **12309394**

Section- **K23NT**

Roll Number- 61

Prepared for
Continuous Assessment 3
Spring 2025

Project Documentation

Real-Time Weather-Based Outfit Suggestion Platform

Table of Contents

1. Introduction:-

1.1 Purpose

The primary goal of this project is to create a user-friendly web application that provides outfit suggestions based on live weather data. In today's fast-paced life, many people do not get the time to check the weather forecast and dress accordingly. This platform bridges that gap by providing automated outfit suggestions, helping users make better dressing decisions effortlessly.

1.2 Scope

The platform is designed to serve users across different regions by accessing weather data using APIs. It is accessible via browsers as it's built with HTML, CSS, and JavaScript for the frontend and PHP/MySQL for backend operations. The platform is hosted on a local XAMPP server. Future upgrades can include features such as login authentication, wardrobe customization, machine learning for personalized suggestions, and integration with fashion e-commerce sites.

2. General Description:-

2.1 Problem Statement

Many individuals step out inappropriately dressed due to not checking the weather. This can lead to discomfort, health issues (e.g., cold or heatstroke), and reduced confidence. Weather unpredictability has made it harder to plan outfits in advance, especially during transition seasons.

2.2 Proposed Solution

This project presents a smart, real-time web platform that fetches live weather data based on the user's location and suggests the most appropriate outfit for the conditions. It uses pre-defined logic that maps temperature, humidity, and condition (rainy, sunny, snowy, etc.) to clothing items that are comfortable and suitable for the environment.

2.3 Objectives

- Automate outfit recommendations based on live weather conditions.
- Provide an interactive and responsive user interface.
- Integrate a reliable weather API.
- Log user queries and preferences into the database for tracking.
- Enhance daily decision-making with minimal effort.

2.4 Target Audience

- General public
- Office-goers and students
- Travelers and commuters
- Elderly individuals who need weather-conscious dressing
- Tech-savvy users interested in automation

2.5 Societal Impact

The application has the potential to improve public health by reducing exposure-related illnesses. It promotes sustainable fashion choices and encourages mindful dressing. Additionally, it can contribute to reducing decision fatigue by automating routine tasks like selecting outfits based on weather.

3. Specific Requirements:-

3.1 Functional Requirements

- The system must access a reliable weather API (e.g., OpenWeatherMap).
- Automatically detect or allow user to input location.
- Fetch current weather parameters: temperature, humidity, condition.
- Display outfit suggestions accordingly.
- Store past queries and responses in a MySQL database.
- Allow users to refresh or check updated data.

3.2 Non-Functional Requirements

- Cross-browser compatibility.
- The application should be responsive and work on mobile/tablet devices.
- Security should be maintained while handling data.
- Must provide feedback or error message if API fails or no data is found.
- Must run seamlessly on XAMPP local server environment.

3.3 User Interface Requirements

- A clean, visually appealing homepage.
- Weather information panel displaying current data.
- Outfit suggestion section with images, text, or icons.
- Option to refresh or check other locations (optional feature).
- Loading animation or status while fetching data.

3.4 Performance Requirements

- Must load weather and outfit suggestions within 2–3 seconds.
- Should handle multiple queries without crashing.
- Efficient mapping logic to reduce backend processing time.
- Minimal memory and CPU usage on client-side.

4. System Architecture:-

4.1 Frontend Technologies

The frontend of the platform is developed using HTML, CSS, and JavaScript. HTML structures the content, CSS styles the user interface for a visually appealing layout, and JavaScript adds interactivity. The weather data is fetched and dynamically displayed using JavaScript.

4.2 Backend Technologies

The backend uses PHP to handle server-side logic, such as fetching data from the weather API and interacting with the MySQL database. PHP scripts are responsible for sending requests, processing the response, and preparing the data for display on the frontend.

4.3 Database Design

MySQL is used to store user data, location queries, and historical weather lookups. Tables are structured to maintain records efficiently, supporting future analytics and user personalization.

4.4 Data Flow Diagram

1. User opens the web page.
2. JavaScript fetches user's location or takes manual input.
3. PHP backend makes a call to the weather API.
4. Weather data is returned and processed.
5. PHP logic matches weather conditions with predefined outfit suggestions.
6. Suggestions are displayed on the UI.
7. All relevant data is logged into the MySQL database.

4.5 System Workflow

The system workflow follows a client-server model where the client (browser) initiates the interaction, and the server processes the data and sends back the appropriate response. Weather information is obtained in real-time, making the platform dynamic and user-centric.

5. Implementation:-

5.1 Modules Overview

- Weather Module: Fetches and processes weather data.
- Outfit Logic Module: Maps weather parameters to suitable outfit suggestions.
- User Interface Module: Handles layout, display, and interactivity.
- Database Module: Logs queries and manages user-related data.

5.2 Integration Process

The integration process involved connecting the frontend with the backend PHP scripts using form submissions and AJAX where necessary. The weather API was integrated via CURL in PHP, and the JSON response was parsed to extract relevant parameters.

5.3 Weather API Integration

An API like OpenWeatherMap is used. The PHP backend sends a request to the API using CURL with parameters like city name or coordinates. The response includes temperature, weather condition, humidity, etc., which are used in logic for outfit suggestion.

5.4 Outfit Logic Design

A set of predefined rules and thresholds (e.g., temperature < 10°C = coat, > 30°C = shorts) was implemented using PHP conditions. Outfits are categorized and associated with weather patterns to ensure comfort and relevance.

6. Testing and Validation:-

6.1 Test Cases

- Test if weather data loads correctly for a known city.
- Check if outfit suggestion changes with weather condition.
- Test responsiveness on different devices.
- Validate error handling for API failures.

6.2 Bug Tracking

During development, issues like API rate limits, incorrect outfit mapping, and frontend display bugs were encountered and resolved. Basic logging and manual testing were used to trace bugs.

6.3 Result Analysis

The application performed reliably in local server environments. Suggestions were consistent and timely. User feedback showed high usability and practicality of the platform.

7. Conclusion and Future Scope:-

The Real-Time Weather-Based Outfit Suggestion Platform successfully addresses the gap between weather awareness and daily outfit decisions. With real-time data and automated suggestions, users benefit from enhanced comfort and decision-making efficiency.

Future enhancements may include:

- User login and profile-based suggestions
- Integration with fashion retailers
- AI-based personalized outfit recommendations
- Voice assistant compatibility

8. References:-

- OpenWeatherMap API Documentation
- PHP and MySQL official docs
- Mozilla Developer Network (MDN) for HTML/CSS/JavaScript
- W3Schools and GeeksforGeeks tutorials

9. Appendix:-

Screenshots, database schema, sample code snippets, and UI mockups can be included here as supporting materials.

10. Github Link:-

https://github.com/mdarifh786/WebDev-Project