

# **DATA SCIENCE MINOR PROJECT REPORT**

## **CONTENTS OF THE REPORT**

- Cover page

- Declaration

- Certificate

- Table of Content

1. Introduction

2. Source of dataset

3. Analysis on dataset (for each objective)

General Description

Specific Requirements

Analysis results

Visualization

4. Conclusion

5. Future scope

6. References

7. Project Repository

## **Other Conventions**

- i. **Please note the case of letters in the cover page:** The 3<sup>rd</sup> line is 16 pt bold and other lines are 12 pt. The page is centred. Department and Institute names are bold.
- ii. All the matter contained in the report should be typed in MS word (1.5 spacing) Times New Roman, 12 pt or equivalent with other software.
- iii. Figures and tables may be inserted in the text as they appear or may be appended in order.
- iv. Table of Content shall be in well hyperlinked
- v. List of figures and tables shall be maintained with captions in MS word.
- vi. List of references shall be appended at the end.
- vii. References shall be in IEEE format
- viii. Total Number of pages with A4 size paper shall be minimum 30 pages and maximum 80 pages.
- ix. Hard copy of report must be available with each student on the day of evaluation.
- x. In addition to Hard copy of reports e-copy shall also be submitted. An e-copy of the report shall be submitted by the student to respective teacher on their emails.

## COVER PAGE

### **ARTIFICIAL INTELLIGENCE ESSENTIALS**

### **PROJECT REPORT**

(Project Semester January-April 2025)

### ***Smart Weather-Based Outfit Recommendation Chatbot***

Submitted by

DIMPLE YADAV

Registration No.: 12308071

Programme and Section: BTECH CSE K23EV

Course Code: INT428

Under the Guidance of

**Anzar Hussain Lone**

**Discipline of CSE/IT**

**Lovely School of Computer Science and Engineering**

## CERTIFICATE

This is to certify that DIMPLE YADAV bearing Registration no. 12308071 has completed INT428 project titled, "*Smart Weather-Based Outfit Recommendation Chatbot*" under my guidance and supervision. To the best of my knowledge, the present work is the result of her original development, effort and study.



**Signature and Name of the Supervisor: Anzar Hussain Lone**

**Designation of the Supervisor: Assistant Professor**

**School of Computer Science and Engineering**

Lovely Professional University

Phagwara, Punjab.

Date: 10-04-2025

## DECLARATION

I, DIMPLE YADAV, student of DATA SCIENCE under CSE Discipline at Lovely Professional University, Punjab, hereby declare that all the information furnished in this project report is based on my own intensive work and is genuine.



Date: 10-04-2025

Signature

Registration No. 12308071

Name of the student:  
DIMPLE YADAV

# INTRODUCTION

In an era where daily decisions are influenced by convenience and technology, choosing the right outfit based on weather conditions can be streamlined for efficiency and comfort. The "Weather-Based Outfit Recommendation Chatbot" is designed to address this need by providing personalized outfit suggestions tailored to real-time weather data and user preferences. This project utilizes the OpenWeatherMap API to fetch current weather information and employs a rule-based system to recommend outfits based on temperature, weather conditions, gender, and style preferences. The chatbot enhances user interaction by simulating a conversational experience, making it a practical tool for everyday use. The complete source code for this project is available on GitHub at [[https://github.com/Dimple04152/WeatherOutfitChatbot/blob/main/Outfit\\_Chatbot%20\(1\).ipynb](https://github.com/Dimple04152/WeatherOutfitChatbot/blob/main/Outfit_Chatbot%20(1).ipynb)]

## SOURCE OF DATA

The project relies on the OpenWeatherMap API as its primary data source, which provides real-time weather information for cities worldwide. The API is queried using a city name provided by the user, returning data such as temperature (in Celsius) and weather conditions (e.g., Clear, Rain, Snow). This data is accessed via HTTP requests using the Python requests library, with the API key 9be69bd07d67c827b1a499175d021137 embedded in the code. The response is received in JSON format, from which the chatbot extracts the temperature (main.temp) and weather condition (weather[0].main) for further processing.

## ANALYSIS ON DATA

This section outlines the core functionality of the chatbot: interpreting weather data to provide outfit recommendations. Unlike traditional data analysis, the "analysis" here refers to the rule-based logic that processes temperature, weather conditions, gender, and style preferences to generate practical clothing suggestions.

## GENERAL DESCRIPTION

The chatbot operates in a structured workflow:

1. **User Interaction:** It greets the user, requests their name, city, gender, and style preference (casual, formal, or sporty), and attempts to extract city and gender from free-form input using regular expressions.
2. **Weather Data Retrieval:** It fetches current weather data for the specified city using the OpenWeatherMap API.
3. **Outfit Recommendation:** Based on the retrieved temperature and weather condition, along with user-provided gender and style preference, it suggests an appropriate outfit.
4. **Response Delivery:** The suggestion is presented in a conversational manner, enhanced by a typing simulation for a human-like feel.

## SPECIFIC REQUIREMENTS

The chatbot's functionality is driven by several key functions:

- **get\_weather(city):**

```
def get_weather(city):  
    base_url = "http://api.openweathermap.org/data/2.5/weather"  
    params = {"q": city, "appid": API_KEY, "units": "metric"}  
    response = requests.get(base_url, params=params)  
    if response.status_code != 200 or "main" not in data:  
        return None  
    data = response.json()  
    return {"temp": data["main"]["temp"], "condition": data["weather"][0]["main"]}
```

- This function retrieves weather data, handling errors by returning None if the request fails.

- **recommend\_outfit(temp, condition, gender, style\_pref):**

```
def recommend_outfit(temp, condition, gender, style_pref):  
    outfit = ""  
    if gender == "male":  
        if style_pref == "formal":  
            if temp < 15:  
                outfit = "Wear a formal coat, shirt with trousers, and leather shoes."  
            else:  
                outfit = "Wear a light formal shirt, trousers, and formal shoes."  
            # Additional conditions for sporty and casual styles  
        elif gender == "female":  
            if style_pref == "formal":  
                if temp < 15:  
                    outfit = "Wear a blazer over a blouse with formal trousers or skirt, a  
                else:  
                    outfit = "Wear a formal dress or top with slacks and heels or flats."  
            # Additional conditions for sporty and casual styles  
    if "rain" in condition.lower():  
        outfit += " 🌦 Carry an umbrella or wear a raincoat."  
    # Additional weather-based adjustments  
    return outfit
```

- This function uses conditional logic to determine outfits based on temperature thresholds, weather conditions, gender, and style preferences.

- **weather\_chatbot():** Orchestrates the user interaction, data retrieval, and recommendation process, incorporating a type\_print function to simulate typing:

```

def type_print(text, delay=1):
    for char in text:
        print(char, end=' ', flush=True)
        time.sleep(delay / 100)
    print()

```

## ANALYSIS RESULTS

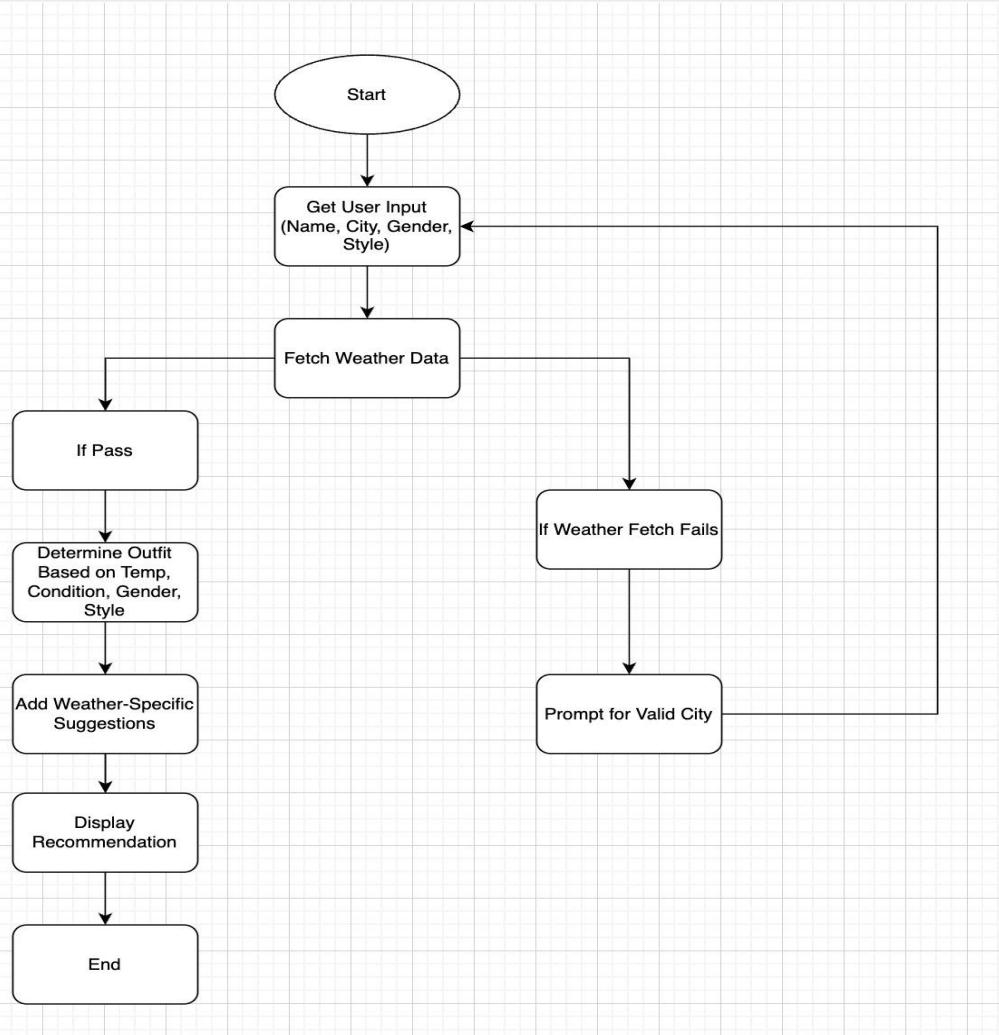
- Example output for a user in Srinagar:

- **Input:** City: Srinagar, Gender: Female, Style: Formal
- **Weather:** Temperature: 26.11°C, Condition: Clear
- **Output:** "Here's what I'd recommend: Wear a formal dress or top with slacks and heels or flats. 🌞 Wear sunglasses and consider a hat or sunscreen."

This demonstrates the chatbot's ability to tailor suggestions to specific weather conditions and user preferences.

## VISUALISATION

A flow chart illustrating the decision-making process:



## CONCLUSION

The Weather-Based Outfit Recommendation Chatbot effectively combines real-time weather data from the OpenWeatherMap API with user preferences to deliver personalized outfit suggestions. Its conversational interface, enhanced by typing simulation, provides an engaging user experience. This project showcases the practical application of API integration and rule-based logic in creating a domain-specific tool that simplifies daily decision-making.

## FUTURE SCOPE

Potential enhancements include:

- **Integration of Large Language Models (LLMs):** To enable more natural language processing and handle out-of-scope queries (e.g., "This prompt is out of scope").
- **Expanded Weather Parameters:** Incorporating humidity, wind speed, or forecasts for more precise recommendations.
- **User Feedback Mechanism:** Allowing users to rate suggestions to refine the logic over time.
- **Multi-Platform Deployment:** Extending the chatbot to web or mobile applications.

## REFERENCES

- OpenWeatherMap API Documentation: <https://openweathermap.org/api>
- Python Requests Library Documentation: <https://docs.python-requests.org/>
- Python Official Documentation: <https://www.python.org/doc/>

## Project Repository

The complete source code for this project is available on GitHub at

[[https://github.com/Dimple04152/WeatherOutfitChatbot/blob/main/Outfit\\_Chatbot%20\(1\).ipynb](https://github.com/Dimple04152/WeatherOutfitChatbot/blob/main/Outfit_Chatbot%20(1).ipynb)]