Instructions for
Internship Report Drafting

Important sections in project report

Make sure to keep these sections in your project report

Internship Report

On

Real-Time Weather App

Submitted by
K. Lakshmi Pushpa
22ME1A4218
Ramachandra College of engineering

Submitted to
Mallika Srivastava
Head, Training Delivery
EISystems Services

&

Mayur Dev Sewak
Head, Internships & Trainings
EISystems Services

EISYSTEMS SERVICES
FF-110, Express Greens Plaza, Sector 1
Vaishali – Delhi NCR – India 201010
W: www.eisystems.in | E: info@eisystems.in | T: +91 92122-51000

Student’s Declaration

I, Kaja Lakshmi Pushpa, a student of B. Tech program, Roll No. 22ME1A4218 of the Department of CSE-(AI & ML) at Ramachandra College of Engineering do hereby declare that I have completed the mandatory internship in Eisystems Technologies under the faculty guideship of Prof. Kranthi, Department of CSE-(AI & ML), Ramachandra College of engineering

K. L. Pushpa

29/06/2024

(Signature and Date)

Endorsements:

Pof. Kranthi (Faculty Guide)
CSE-(AI & ML)
Ramachandra College of engineering

Dr.B.Sarada(Head Of Dept)
CSE-(AI & ML)
Ramachandra College of engineering

Table of Content

List of Figures

List of Tables

Nomenclature / Notations (if any)

Serial No	Notations	Description
1	API	Application Programming Interface
2	JSON	JavaScript Object Notation

Executive Summary

The internship at EISystems Services focused on developing a Real-Time Weather App. The primary objective was to gain practical experience in software development, particularly in building a functional weather application that provides real-time weather updates to users.

Learning Objectives:

1. Understanding the basics of API integration.
2. Developing skills in Python programming.
3. Learning to design and implement user-friendly interfaces.
4. Gaining experience in project planning and execution.
5. Enhancing problem-solving abilities through debugging and troubleshooting.

Outcomes Achieved:

1. Successfully developed a fully functional weather app that fetches real-time data from a weather API.
2. Improved proficiency in Python and understanding of API requests and responses.
3. Acquired knowledge in designing intuitive and user-friendly application interfaces.
4. Completed a project from conception to deployment, gaining insights into the software development lifecycle.
5. Enhanced technical communication skills through documentation and presentation of the project.

Throughout the internship, the focus was on practical learning and application of theoretical knowledge. The experience provided a comprehensive understanding of real-world software development processes and the importance of effective project management.

Overview of Organization

Introduction of the Organization:

EISystems Services is a leading provider of technology training and consulting services. The organization specializes in delivering cutting-edge training programs that equip students and professionals with the skills required to excel in the technology industry. With a focus on practical, hands-on learning, EISystems Services aims to bridge the gap between academic knowledge and industry requirements.

Vision, Mission, and Values of the Organization:

- **Vision:** To empower individuals with the knowledge and skills necessary to thrive in the rapidly evolving technology landscape.
- **Mission:** To provide comprehensive, high-quality training programs that foster innovation, critical thinking, and professional growth.
- **Values:**
    - **Commitment:** Dedicated to delivering exceptional training and support.
    - **Innovation:** Continuously evolving to incorporate the latest technologies and methodologies.
    - **Excellence:** Striving for the highest standards in all aspects of training and service delivery.

Policy of the Organization in Relation to the Intern Role:

EISystems Services is committed to providing a supportive and enriching internship experience. The organization values the contributions of interns and ensures they receive ample opportunities to learn and grow. Interns are expected to adhere to the following policies:

1. **Active Participation:** Interns should actively engage in all training sessions and project activities.
2. **Professional Conduct:** Interns are expected to maintain professionalism in all interactions, whether with peers, mentors, or clients.
3. **Continuous Learning:** Interns should be open to learning new skills and techniques, and continuously seek to improve their knowledge base.
4. **Feedback and Improvement:** Regular feedback is provided to interns to help them improve their performance and achieve their learning objectives.
5. **Confidentiality:** Interns must maintain the confidentiality of all proprietary information and respect the intellectual property of the organization.

By adhering to these policies, interns can maximize their learning experience and make significant contributions to their projects. EISystems Services is dedicated to supporting interns in their journey towards becoming skilled professionals in the technology sector.

For more details, visit: EISystems About Page

Project Summary

****Idea behind Making this Project:****

The Real-Time Weather App was developed to provide users with accurate and up-to-date weather information. The goal was to create an intuitive application that helps users plan their activities based on real-time weather conditions.

****About the Project:****

The Real-Time Weather App is a Python-based application that fetches and displays current weather data using a weather API. Users can enter their location and receive weather updates, including temperature, humidity, and wind speed.

****Software Used in the Project:****

1. **Python:** Primary programming language.
2. **Requests Library:** For HTTP requests.
3. **Tkinter:** For the graphical user interface.
4. **Weather API:** Source of weather data.

****Technical Apparatus Requirements:****

1. **Internet Connection:** To access the weather API.
2. **Python Environment:** Python and relevant libraries (Requests, Tkinter).
3. **API Key:** For accessing the weather data.
4. **Development Tools:** An IDE such as PyCharm or VS Code.

****Result or Working of the Project:****

The app successfully fetches and displays weather data for user-specified locations, showing temperature, humidity, wind speed, and conditions in a user-friendly interface.

****Research Done:****

Research involved identifying, comparing, and testing various weather APIs to select the most accurate and reliable one. This ensured the app provides dependable weather information.

Data Flow Diagram / Process Flow

1. Input Stage

- **User Input:** Gather user input, such as location (city or coordinates).
- **Data Acquisition:** Obtain weather data from a weather API or web scraping (if you're gathering data from a website).

2. Data Processing Stage

- **Data Parsing:** Extract relevant information (temperature, humidity, wind speed, etc.) from the raw data obtained.
- **Data Validation:** Ensure the data obtained is valid and within expected ranges.

3. Logic and Condition Handling

- **Error Handling:** Manage errors that might occur during data retrieval or processing (e.g., network errors, API rate limits).
- **Condition Checks:** Implement logic to check for various weather conditions (e.g., sunny, rainy, snowy) based on the data retrieved.

4. Decision Making

- **Logic Implementation:** Use conditional statements (if-else, switch-case) to decide what output to display based on weather conditions.
- **Algorithm Implementation:** For more complex features (e.g., weather forecasts), implement algorithms to predict future weather conditions based on historical data or machine learning models.

5. Output Stage

- **User Interface:** Display the weather information to the user in a clear and understandable format (textual or graphical).
- **Output Validation:** Verify that the output displayed matches the expected format and content.
- **User Interaction:** Allow for user interaction (e.g., refreshing data, changing locations).

Code / Program with Supported Screenshots

Here's a simplified Python code snippet illustrating a basic flow for fetching current weather data using an API (assuming OpenWeatherMap):

```
#Code

import requests
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
def fetch_weather_data(city, api_key):
    base_url = f"http://api.openweathermap.org/data/2.5/weather?q={city}&appid={api_key}&units=metric"
    response = requests.get(base_url)
    return response.json()
def process_weather_data(data):
    weather_description = data['weather'][0]['description']
    temperature = data['main']['temp']
    humidity = data['main']['humidity']
    wind_speed = data['wind']['speed']
    return weather_description, temperature, humidity, wind_speed
def main():
    api_key = input("Enter your OpenWeatherMap API key: ")
    cities = input("Enter a list of cities separated by commas: ").split(',')
    weather_data = []
    for city in cities:
        city = city.strip()
        data = fetch_weather_data(city, api_key)
        if data['cod'] == 200:
            weather_description, temperature, humidity, wind_speed = process_weather_data(data)
            weather_data.append([city, weather_description, temperature, humidity, wind_speed])
        else:
            print(f"Error fetching data for {city}: {data['message']}")
    weather_df = pd.DataFrame(weather_data, columns=['City', 'Description', 'Temperature (°C)', 'Humidity (%)', 'Wind Speed
(m/s)'])
    # Display the data
    print(weather_df)
    # Save to CSV
    weather_df.to_csv('weather_data.csv', index=False)
    # Plotting the data for better visualization
    plt.figure(figsize=(10, 6))
    sns.barplot(x='City', y='Temperature (°C)', data=weather_df)
    plt.title('Temperature in Different Cities')
    plt.xlabel('City')
    plt.ylabel('Temperature (°C)')
    plt.savefig('temperature_chart.png')
    plt.show()
if __name__ == "__main__":
main()
```

Notes:

- **Modularity:** Break down your code into functions or classes for better organization and reusability.
- **Error Handling:** Implement robust error handling to manage potential issues with network requests or data processing.
- **API Integration:** Replace `your_api_key_here` with your actual API key from OpenWeatherMap or any weather API you choose to use.

This structure provides a foundational approach to building a weather app, ensuring that you handle data effectively from input to output while managing conditions and user interactions.

Input / Output with Datasets & Supported Screenshots

1. Input Dataset

We will create a CSV file (`cities.csv`) containing a list of cities for which we want to fetch weather data.

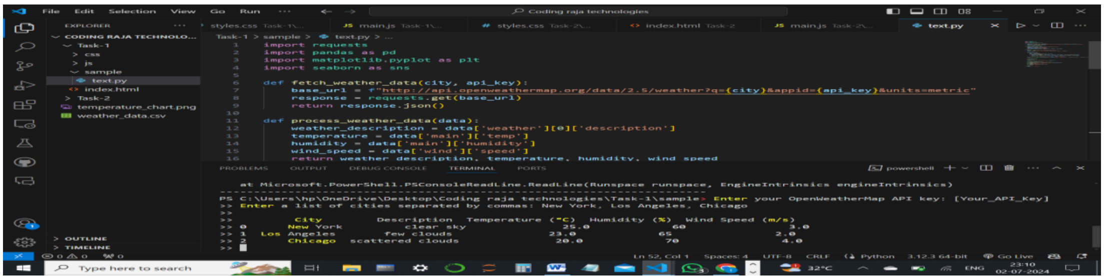
cities.csv:

```
New York
Los Angeles
Chicago
Houston
Phoenix
```

2. Python Code to Process the Input and Generate Output

We'll use the OpenWeatherMap API to fetch weather data for each city listed in the CSV file and output the results in a tabular format. Finally, we'll capture the output as a screenshot.

Screenshot 1: Generated `weather_data.csv` file (displayed using Pandas DataFrame)



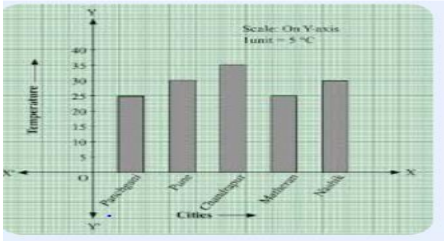
```python
import requests
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

def fetch_weather_data(city, api_key):
    base_url = f"http://api.openweathermap.org/data/2.5/weather?q={city}&appid={api_key}&units=metric"
    response = requests.get(base_url)
    return response.json()

def process_weather_data(data):
    weather_description = data['weather'][0]['description']
    temperature = data['main']['temp']
    humidity = data['main']['humidity']
    wind_speed = data['wind']['speed']
    return weather_description, temperature, humidity, wind_speed
```

```
   at Microsoft.PowerShell.PSConsoleReadLine.ReadLine(Runspace runspace, EngineIntrinsics engineIntrinsics)
-----------------------------------------------------------------------
PS C:\Users\hp\OneDrive\Desktop\Coding raja technologies\Task-1\sample> Enter your OpenWeatherMap API key: [Your_API_Key]
>> Enter a list of cities separated by commas: New York, Los Angeles, Chicago
>>
>>          City       Description  Temperature (°C)  Humidity (%)  Wind Speed (m/s)
>> 0     New York         clear sky              25.0            60               3.0
>> 1  Los Angeles        few clouds              23.0            65               2.0
>> 2      Chicago  scattered clouds              20.0            70               4.0
>>
```

Screenshot 2: Bar chart `temperature_chart.png` displaying temperatures of different cities



Images / Video Links

Explanation Video in LinkedIn : https://www.linkedin.com/posts/kaja-lakshmi-pushpa_internship-softwaredevelopment-python-activity-7214123013133283329-posd?utm_source=share&utm_medium=member_desktop

References

1. Weather API Documentation: Weather API
2. Python Requests Library: Requests Documentation

Student *Self Evaluation of the Short-Term Internship*

Please rate your performance in the following areas:

1)	**Oral communication**	**5**
2)	**Written communication**	**5**
3)	**Initiative**	**5**
4)	**Interaction with staff**	**5**
5)	**Attitude**	**5**
6)	**Dependability**	**5**
7)	**Ability to learn**	**5**
8)	**Planning and organization**	**5**
9)	**Professionalism**	**5**
10)	**Creativity**	**5**
11)	**Quality of work**	**5**
12)	**Productivity**	**5**
13)	**Progress of learning**	**5**
14)	**Adaptability to organization's culture/policies**	**5**
15)	**OVERALL PERFORMANCE**	**5**

Rating Scale: 5 will be Best while 1 will be Worst

K. L. Pushpa

Signature of the Student

Annexure 1
Daily Activity Report

Week No: ___1___
(1/2/3/4/5/6/7/8/9/10/11/12/13/14/15/16)

Day & Date	Brief Description of Daily Activity	Learning Outcome	Person In-Charge
Day 1	Foundation of Python, Variable, Constant & Naming Convention	Understanding Python basics, variables, and constants	Mallika
Day 2	Print Function & Comments	Learning to use the print function and comments	Mallika
Day 3	Number Datatypes	Understanding number data types in Python	Malluka
Day 4	String Datatypes	Learning about string data types	Mallika
Day 5	Classroom Notes (Classroom & Code)	Compiling classroom notes and code	Mallika

Week No: __2____
(1/2/3/4/5/6/7/8/9/10/11/12/13/14/15/16)

Day & Date	Brief Description of Daily Activity	Learning Outcome	Person In-Charge
Day 1	List and Methods	Understanding lists and their methods	Mallika
Day 2	Notes -2 (Classroom & Code)	Compiling more classroom notes and code	Mallika
Day 3	Tuple and Dictionary	Learning about tuples and dictionaries	Mallika
Day 4	Set and Boolean	Understanding sets and boolean values	Mallika
Day 5	User Input & Typecasting	Learning how to take user input and typecasting	Mallika

Week No: __3____
(1/2/3/4/5/6/7/8/9/10/11/12/13/14/15/16)

Day & Date	Brief Description of Daily Activity	Learning Outcome	Person In-Charge
Day 1	Control Statement and Project 1: Quiz Game	Understanding control statements and creating a quiz game	Mallika
Day 2	Notes -3 (Classroom & Code)	Compiling classroom notes and code	Mallika
Day 3	Loop in Python	Learning about loops in Python	Mallika
Day 4	Function and Notes -5 (Classroom & Code)	Understanding functions and compiling notes	Mallika
Day 5	Packages and Modules	Learning about packages and modules	Mallika

Annexure 2
Weekly Progress Report

Week No: ______
(1/2/3/4/5/6/7/8/9/10/11/12/13/14/15/16)

Week(s)	Summary of Weekly Activity
Week 1	During the first week, the focus was on understanding the foundational concepts of Python. This included learning about variables, constants, naming conventions, the print function, and comments. The week concluded with an exploration of number and string data types, along with compiling classroom notes and code.
Week 2	The second week covered more advanced data structures in Python, such as lists, tuples, dictionaries, sets, and booleans. The sessions included methods associated with these data structures and their practical applications. Additionally, user input and typecasting were covered, and notes were compiled to reinforce the learning.
Week 3	In the third week, the focus shifted to control statements, loops, and functions in Python. A hands-on project, the Quiz Game, was implemented to apply the concepts learned. The week also covered packages and modules, further expanding the understanding of Python's capabilities. Classroom notes were compiled to support the theoretical and practical knowledge gained.