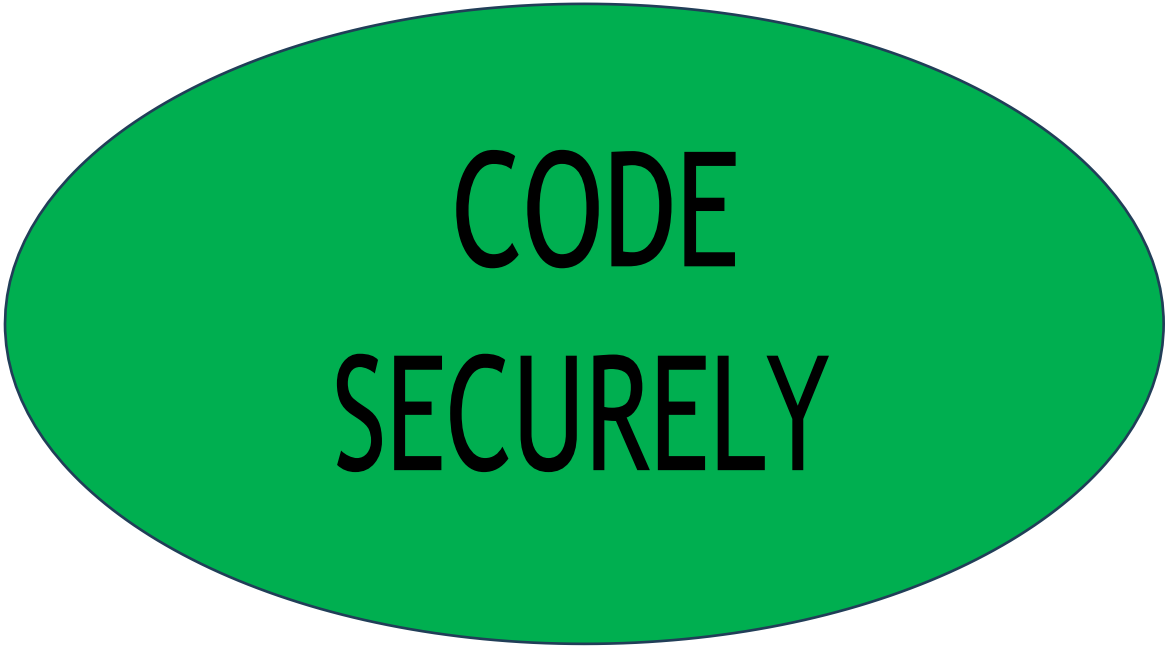


**EVALUATION OF SOFTWARE TOOLS AND PROGRAMMING  
FUNDAMENTALS IN THE PROGRAM DEVELOPMENT**

**REPORT DATE:** JANUARY 7, 2025

**SUBMITTED BY:** OLAWALE FRANCIS ONAOLAPO



**CODE  
SECURELY**

## TABLE OF CONTENTS

	PAGE
ABSTRACT	3
1.0 INTRODUCTION	4
2.0 EVALUATION OF SOFTWARE TOOLS	4
3.0 SECURITY AND RISK IMPLICATIONS	6
4.0 CONTRIBUTION OF PROGRAMMING FUNDAMENTALS	6
5.0 CONCLUSION	8
REFERENCES	9
APPENDIX – BLACKBOX TESTING ENTRIES	10

## **ABSTRACT**

This report evaluates the software tools, libraries, and programming fundamentals employed in the development of an application for the program.

Key tools include Visual Studio Code, SQLite, and Python packages such as Matplotlib, Geopy, and PyInstaller. Their appropriateness, limitations, and security implications are examined, emphasizing risks such as vulnerabilities introduced by third-party modules and SQLite's lack of built-in security features.

The report also underscores the importance of secure coding practices, source verification, and regular updates to mitigate risks in real-world data science applications. Furthermore, it highlights how programming fundamentals, such as functions, robust error handling, efficient data structures, and control flow mechanisms, contributed to the program's development. The findings demonstrate the critical need for careful tool selection, secure practices, and foundational programming principles to ensure effective and secure application development.

## 1.0 INTRODUCTION

This report is part of the program and aims to:

- Highlight the software tools used during the program and evaluate their appropriateness and limitations.
- Assess the security and risk implications associated with using software libraries and tools in real-world Data Science (DS) applications.
- Discuss how the fundamentals of programming and data structures provided by the standard library and environment contributed to the development of the application.

## 2.0 EVALUATION OF SOFTWARE TOOLS

The development of the program utilized several software tools, including:

- **Visual Studio Code (VS Code)**
- **SQLite Database**
- **Python packages/modules outside the standard library** such as Matplotlib, Requests, Geopy, TimezoneFinder, and PyInstaller.

Below is an evaluation of these tools, their appropriateness for the project, and their limitations.

### 2.1 Visual Studio Code (VS Code)

#### Reasons for Use:

- VS Code was chosen for its free, open-source nature and ease of installation.
- It is compatible with the Windows operating system used during development.
- Its extensive support for extensions enhances the development experience, particularly for Python projects and SQLite database integration.

#### Limitations:

- The open-source nature of its extension ecosystem poses potential risks if extensions from unverified sources are installed, introducing vulnerabilities or malicious code.
- Regular updates are required to address bugs and security issues. If not kept up to date, users may be exposed to risks.

## 2.2 SQLite Database

### Reasons for Use:

- SQLite's free and open-source nature makes it an accessible choice.
- It is lightweight and serverless, requiring only a connection to create databases.
- Its compatibility with SQL syntax and seamless integration with Python enhanced its utility for the program.
- Comprehensive documentation and community resources facilitated its adoption.

### Limitations:

- SQLite has size and quantity constraints, such as a default maximum of 2,000 columns per table and a maximum of 64 tables joined in a single query (SQLite, 2024a).
- Security limitations include a lack of built-in access control or authorization features, as it is not a service-based database. Security is instead dependent on operating system-level file access controls (Liu and Gong, 2013).
- SQLite lacks an auditing mechanism, and backups must be performed manually by copying database files (Liu and Gong, 2013).

## 2.3 Python Packages/Modules Outside the Standard Library

### Key Modules Used:

- **Matplotlib:** Enabled graphical data visualization, making database information more interpretable and presentable.
- **Geopy:** Automated the generation of latitude, longitude, and country names for cities, eliminating manual hardcoding.
- **TimezoneFinder:** Automatically determined time zones for cities, reducing manual effort.
- **PyInstaller:** Converted the Python-based graphical user interface into a standalone application, allowing users to run the program without accessing or altering the Python script.

### Limitations:

- Third-party modules may introduce vulnerabilities if sourced from unverified repositories or left unpatched. Regular updates and source verification are essential to mitigate these risks.

### 3.0 SECURITY AND RISK IMPLICATIONS

The use of software libraries and tools in the development of data science application deployed to the real-world introduces potential security risks, including:

- **Broken Access Control:** Identified as the most prevalent vulnerability by the Open Web Application Security Project (OWASP, 2021).
- **Other Security Risks:** Cryptographic failures, injection attacks, insecure design, and related vulnerabilities (OWASP, 2021).
- **SQLite Vulnerabilities:** SQLite (2024b) highlights several risks such as arbitrary SQL statement injection, which can lead to application crashes or denial of service attacks.
- SecurityScorecard (2025) recorded that there were 3970 vulnerabilities impacted due to code execution in 2024, 2655 vulnerabilities due to SQL injection among other vulnerabilities impact types.

If exploited, these vulnerabilities in data science applications could enable attackers to steal sensitive information, such as location data via the geopy module or database records through SQL injection attacks on SQLite, potentially leading to financial or reputational losses.

To mitigate these risks, developers should adopt secure coding practices, verify the sources of libraries, and regularly update tools and dependencies. For example, the **secrets** module is recommended for security-related tasks, such as generating cryptographically secure random numbers, instead of the **random** module (Python Software Foundation, 2025).

### 4.0 CONTRIBUTION OF PROGRAMMING FUNDAMENTALS

The fundamentals of programming and data structures, provided by the standard library and environment, played a critical role in the application's development. Key contributions include:

#### 4.1 Functions

- Functions enhanced code reusability and modularity. Examples:
  - Database operations such as establishing connections and retrieving data were encapsulated in dedicated functions.
  - Separate functions were used for data validation and API data retrieval, improving maintainability and scalability.

## 4.2 Error Handling with Try-Except

- Robust error handling improved program stability by managing runtime errors. Examples:
  - `sqlite3.OperationalError` managed database-related issues.
  - `ValueError` prevented crashes due to invalid user inputs, enhancing user-friendliness.

## 4.3 Data Structures

- **Dictionaries:** Structured data, such as city details, were passed between functions for efficient manipulation and access using key-value pairs.
- **Lists:** Facilitated batch processing, such as handling multiple city names for weather data retrieval and storing iteration results.

## 4.4 Iteration and Control Flow

- **Loops:** Simplified repetitive tasks like iterating through database results or validating user inputs.
- **Conditional Statements:** Ensured decision-making, such as validating user-provided dates and IDs, was efficient and accurate.

## 4.5 Input and Output

- **Input Functions:** Allowed dynamic customization of queries, such as selecting specific cities or date ranges.
- **Output Functions:** Used to display feedback and results to users, improving interactivity and user engagement.

## 4.6 Data Types

- Proper handling of data types was crucial for accurate data manipulation. Examples:
  - Strings and integers were validated during user input and database queries.
  - Float values were formatted to display temperatures and precipitation data clearly.

#### 4.7 Standard Library and Environment

- **SQLite Module:** Enabled seamless integration for storing and querying weather data.
- **Matplotlib:** Generated visualizations like bar charts and histograms to display temperature and precipitation trends.
- **Datetime Module:** Provided robust date validation and manipulation, ensuring user inputs adhered to required formats.

#### 5.0 CONCLUSION

This report evaluated the software tools and libraries used in the project, highlighting their suitability, limitations, and security implications. Additionally, it demonstrated how programming fundamentals and data structures facilitated the program's development. Hence, it highlights the need to always ensure to employ best practices in tool selection and maintaining secure coding while utilizing the fundamental programming principles to develop applications.



## REFERENCES

Liu, H. and Gong, Y. (2013) 'Analysis and Design on Security of SQLite', in *Proceedings of the International Conference on Computer, Networks and Communication Engineering (ICCNCE 2013)*, China: Atlantis Press. Available at: <https://doi.org/10.2991/iccnce.2013.145>.

OWASP - The Open Worldwide Application Security Project (2021) *Introduction: Welcome to the OWASP Top 10 – 2021*. Available at: <https://owasp.org/Top10/> (Accessed: 03 January 2025).

Python Software Foundation (2025) *Security Considerations*. Available at: [https://docs.python.org/3/library/security\\_warnings.html](https://docs.python.org/3/library/security_warnings.html) (Accessed: 05 January 2025).

SecurityScorecard (2025) *Vulnerabilities By Types/Categories*. Available at: <https://www.cvedetails.com/vulnerabilities-by-types.php> (Accessed: 05 January 2025).

SQLite (2024a) *Limits In SQLite*. Available at: <https://www.sqlite.org/limits.html> (Accessed: 02 January 2025).

SQLite (2024b) *Vulnerabilities*. Available at: <https://www.sqlite.org/cves.html> (Accessed: 02 January 2025).

## **APPENDIX – BLACKBOX TESTING ENTRIES**

## PHASE 1 - 4 BLACKBOX TESTING ENTRIES

LEGEND:						
HEADINGS		FAIL	PASS	PASS		
ID	DESCRIPTION	STEPS	EXPECTED	ACTUAL	PASS / FAIL	COMMENT
PHASE 1 BLACKBOX TESTING ENTRIES						
1	<u>SET-UP TESTING</u> Ensure the user is notified in case there is issue connecting to the database	1. Removed the database to another location 2. Run the functions	unable to open database file Failed to connect to the database.	unable to open database file Failed to connect to the database.	PASS	
2	<u>USER INPUT VALIDATION</u> Ensure the select_all_countries(connection) function runs as expected	1. Start the select_all_countries(connection) code 2. Entered the database connection variable, connection, inside the function and click the run code button.	Country Id: 1 -- Country Name: Great Britain -- Country Timezone: Europe/London	Country Id: 1 -- Country Name: Great Britain -- Country Timezone: Europe/London	PASS	
3	<u>USER INPUT VALIDATION</u> Ensure the select_all_cities(connection) function runs as expected	1. Start the select_all_cities(connection) code 2. Entered the database connection variable, connection, inside the function and click the run code button.	City Id: 1 -- City Name: Middlesbrough -- City Longitude: -1.2344047 -- City Latitude: 54.5760419 -- Country ID: 1	City Id: 1 -- City Name: Middlesbrough -- City Longitude: -1.2344047 -- City Latitude: 54.5760419 -- Country ID: 1	PASS	

4	<u>USER INPUT VALIDATION</u> Ensure the user is notified when wrong parameter is inputted into the average_annual_temperature(connection, city_id, year) function	1. Entered e as input for the city_id	Enter a valid numeric City ID.	Enter a valid numeric City ID.	PASS	
		2. Entered 7 as input for the city_id	Invalid City ID. Please choose from the following IDs: [1, 2, 3, 4, 5, 6].	Invalid City ID. Please choose from the following IDs: [1, 2, 3, 4, 5, 6].	PASS	
		3. Entered 3 as the input for the city_id and tu as input for the year	Invalid year format. Enter a 4-digit year.	Invalid year format. Enter a 4-digit year.	PASS	
		4. Entered 2 as the input for the city_id and 2019 as input for the year	Invalid year. Available years: ['2020', '2021', '2022', '2023', '2024'].	Invalid year. Available years: ['2020', '2021', '2022', '2023', '2024'].	PASS	
		5. Entered 3 as the input for the city_id and 2026 as input for the year	Invalid year. Available years: ['2020', '2021', '2022', '2023', '2024'].	Invalid year. Available years: ['2020', '2021', '2022', '2023', '2024'].	PASS	
		6. Entered 3 as the input for the city_id and 2022 as input for the year	City Id: 3 -- City Name: Paris -- Year: 2022 -- Annual Mean Temperature: 13.14°C	City Id: 3 -- City Name: Paris -- Year: 2022 -- Annual Mean Temperature: 13.14°C	PASS	
5	<u>CALCULATION TEST</u> Ensure the the average_annual_temperature(connection, city_id, year) function output the correct value	1. Entered 3 as the input for the city_id and 2022 as input for the year	Annual Mean Temperature: 13.14°C	Annual Mean Temperature: 13.14°C	PASS	
6	<u>USER INPUT VALIDATION</u> Ensure the user is notified when	1. Entered gh as input for the city_id	Invalid input: gh. Please enter a valid number.	Invalid input: gh. Please enter a valid number.	PASS	

wrong parameter is inputted into the average_seven_day_precipitation(connection, city_id, start_date) function	2. Entered 0 as input for the city_id	Invalid City ID. Please choose from the following IDs: [1, 2, 3, 4, 5, 6].	Invalid City ID. Please choose from the following IDs: [1, 2, 3, 4, 5, 6].	PASS	
	3. Entered 4 as the input for the city_id and xx as input for the start_date	Invalid dates. Please choose dates within the range: 2020-01-01 - 2024-12-27.	Invalid dates. Please choose dates within the range: 2020-01-01 - 2024-12-27.	PASS	
	4. Entered 4 as the input for the city_id and 2017 as input for the start_date	Invalid dates. Please choose dates within the range: 2020-01-01 - 2024-12-27.	Invalid dates. Please choose dates within the range: 2020-01-01 - 2024-12-27.	PASS	
	5. Entered 4 as the input for the city_id and 2027 as input for the start_date	Invalid dates. Please choose dates within the range: 2020-01-01 - 2024-12-27.	Invalid dates. Please choose dates within the range: 2020-01-01 - 2024-12-27.	PASS	
	6. Entered 4 as the input for the city_id and 2022-01-2 as input for the start_date	Invalid dates. Please choose dates within the range: 2020-01-01 - 2024-12-27.	Invalid dates. Please choose dates within the range: 2020-01-01 - 2024-12-27.	PASS	
	7. Entered 4 as the input for the city_id and 2026-01-22 as input for the start_date	Invalid dates. Please choose dates within the range: 2020-01-01 - 2024-12-27.	Invalid dates. Please choose dates within the range: 2020-01-01 - 2024-12-27.	PASS	
	8. Entered 4 as the input for the city_id and 2018-01-20 as input for the start_date	Invalid dates. Please choose dates within the range: 2020-01-01 - 2024-12-27.	Invalid dates. Please choose dates within the range: 2020-01-01 - 2024-12-27.	PASS	

		9. Entered 4 as the input for the city_id and 2021-02-02 as input for the start_date	City Id: 1 -- City Name: Middlesbrough -- Seven Days Starting From: 2021-02-02 -- Average Seven Day Precipitation: 7.73mm	City Id: 1 -- City Name: Middlesbrough -- Seven Days Starting From: 2021-02-02 -- Average Seven Day Precipitation: 7.73mm	PASS	
7	<u>CALCULATION TEST</u> Ensure the the average_seven_day_precipitation(connection, city_id, start_date) function output the correct value	1. Entered 4 as the input for the city_id and 2021-02-02 as input for the start_date	Average Seven Day Precipitation: 7.73mm	Average Seven Day Precipitation: 7.73mm	PASS	
8	<u>USER INPUT VALIDATION</u> Ensure the user is notified when wrong parameter is inputted into the average_mean_temperature_by_city(connection, date_from, date_to) function	1. Entered s as input for the date_from and y as the input for the date_to	Invalid dates. Please choose dates within the range: 2020-01-01 - 2024-12-27.	Invalid dates. Please choose dates within the range: 2020-01-01 - 2024-12-27.	PASS	
		2. Entered z as input for the date_from and 2020-01-04 as the input for the date_to	Invalid dates. Please choose dates within the range: 2020-01-01 - 2024-12-27.	Invalid dates. Please choose dates within the range: 2020-01-01 - 2024-12-27.	PASS	
		3. Entered 2020-04-06 as input for the date_from and d as the input for the date_to	Invalid dates. Please choose dates within the range: 2020-01-01 - 2024-12-27.	Invalid dates. Please choose dates within the range: 2020-01-01 - 2024-12-27.	PASS	
		4. Entered 2020 as input for the date_from and 2024 as the input for the date_to	Invalid dates. Please choose dates within the range: 2020-01-01 - 2024-12-27.	Invalid dates. Please choose dates within the range: 2020-01-01 - 2024-12-27.	PASS	

		5. Entered 2023-03-21 as input for the date_from and 2024-06 as the input for the date_to	Invalid dates. Please choose dates within the range: 2020-01-01 - 2024-12-27.	Invalid dates. Please choose dates within the range: 2020-01-01 - 2024-12-27.	PASS	
		6. Entered 2024-09-10 as input for the date_from and 2024-01-01 as the input for the date_to	Start date cannot be after the end date. Please re-enter the dates.	Start date cannot be after the end date. Please re-enter the dates.	PASS	
		7. Entered 2024-09-10 as input for the date_from and 2026-01-01 as the input for the date_to	Invalid dates. Please choose dates within the range: 2020-01-01 - 2024-12-27.	Invalid dates. Please choose dates within the range: 2020-01-01 - 2024-12-27.	PASS	
		8. Entered 2020-09-08 as input for the date_from and 2023-04-09 as the input for the date_to	City Id: 1 -- City Name: Middlesbrough -- Date From: 2020-09-08 -- Date To: 2023-04-09 -- Average Mean Temperature: 9.81 °C	City Id: 1 -- City Name: Middlesbrough -- Date From: 2020-09-08 -- Date To: 2023-04-09 -- Average Mean Temperature: 9.81 °C	PASS	
9	<u>CALCULATION TEST</u> Ensure the the average_mean_temperature_by_city(connection, date_from, date_to) function output the correct value	1. Entered 2020-09-08 as input for the date_from and 2023-04-09 as the input for the date_to	Average Mean Temperature: 9.81 °C	Average Mean Temperature: 9.81 °C	PASS	
10	<u>USER INPUT VALIDATION</u> Ensure the user is	1. Entered atg as input for the year	Invalid year format. Enter a 4-digit year.	Invalid year format. Enter a 4-digit year.	PASS	





12	<u>USER INPUT VALIDATION</u> Ensure the user is notified when wrong parameter is inputted into the plot_annual_temperatures_grouped_bar(db_city_names, city_annual_temperature_record) plotting function	1.No input is required for the user. The set-up is checked to ensure the database connection variable name is accurate. I clicked on the run code button.	Group bar chart of the average annual mean temperature by city from 2020 to 2024 (plotted for all the years and the cities in the database) - with different bar colours for each year	Group bar chart of the average annual mean temperature by city from 2020 to 2024 (plotted for all the years and the cities in the database) - with some bars for different years having the same colours.	FAIL	Use colormaps in matplotlib dynamically map the dynamic variables, years, in chart.
13	<u>USER INPUT VALIDATION</u> Ensure the user is notified when wrong parameter is inputted into the plot_annual_temperatures_grouped_bar(db_city_names, city_annual_temperature_record) plotting function	1.No input is required for the user. The set-up is checked to ensure the database connection variable name is accurate. I clicked on the run code button.	Group bar chart of the average annual mean temperature by city from 2020 to 2024 (plotted for all the years and the cities in the database) - with different bar colours for each year	Group bar chart of the average annual mean temperature by city from 2020 to 2024 (plotted for all the years and the cities in the database) - with different bar colours for each year	PASS	
14	<u>CALCULATION TEST</u> Ensure the the plot_annual_temperatures_grouped_bar(db_city_names, city_annual_temperature_record) plotting function output the correct value	1. No input is required for the user. The set-up is checked to ensure the database connection variable name is accurate. I clicked on the run code button.	1. Paris average annual mean temperature for 2020 = 13.2° C 2. Lagos average annual mean temperature for 2022 = 26.7° C	1. Paris average annual mean temperature for 2020 = 13.2° C 2. Lagos average annual mean temperature for 2022 = 26.7° C	PASS	

15	<u>USER INPUT VALIDATION</u> Ensure the user is notified when wrong parameter is inputted into the plot_bar_chart_with_mean(data, mean_temperature, date_from, date_to) plotting function	1. Entered f as input for the date_from and u as the input for the date_to	Invalid dates. Please choose dates within the range: 2020-01-01 - 2024-12-27.	Invalid dates. Please choose dates within the range: 2020-01-01 - 2024-12-27.	PASS	
		2. Entered a as input for the date_from and 2023-21-04 as the input for the date_to	Invalid dates. Please choose dates within the range: 2020-01-01 - 2024-12-27.	Invalid dates. Please choose dates within the range: 2020-01-01 - 2024-12-27.	PASS	
		3. Entered 2024-04-06 as input for the date_from and w as the input for the date_to	Invalid dates. Please choose dates within the range: 2020-01-01 - 2024-12-27.	Invalid dates. Please choose dates within the range: 2020-01-01 - 2024-12-27.	PASS	
		4. Entered 2022 as input for the date_from and 2023 as the input for the date_to	Invalid dates. Please choose dates within the range: 2020-01-01 - 2024-12-27.	Invalid dates. Please choose dates within the range: 2020-01-01 - 2024-12-27.	PASS	
		5. Entered 2022-03-21 as input for the date_from and 2022-06 as the input for the date_to	Invalid dates. Please choose dates within the range: 2020-01-01 - 2024-12-27.	Invalid dates. Please choose dates within the range: 2020-01-01 - 2024-12-27.	PASS	
		6. Entered 2023-09-16 as input for the date_from and 2022-01-03 as the input for the date_to	Start date cannot be after the end date. Please re-enter the dates.	Start date cannot be after the end date. Please re-enter the dates.	PASS	
		7. Entered 2022-09-10 as input for the date_from and 2026-05-01 as the input for the date_to	Invalid dates. Please choose dates within the range: 2020-01-01 - 2024-12-27.	Invalid dates. Please choose dates within the range: 2020-01-01 - 2024-12-27.	PASS	

		8. Entered 2022-02-30 as input for the date_from and 2024-03-29 as the input for the date_to	Invalid dates. Please choose dates within the range: 2020-01-01 - 2024-12-27.	Invalid dates. Please choose dates within the range: 2020-01-01 - 2024-12-27.		
		9. Entered 2020-02-03 as input for the date_from and 2024-12-09 as the input for the date_to	Bar chart of the average mean temperature by city from 2020-02-03 to 2024-12-09	Bar chart of the average mean temperature by city from 2020-02-03 to 2024-12-09	PASS	
16	<u>CALCULATION TEST</u> Ensure the the plot_bar_chart_with_mean(data, mean_temperature, date_from, date_to) plotting function output the correct value	9. Entered 2020-02-03 as input for the date_from and 2024-12-09 as the input for the date_to	1. Middlesbrough average mean temperature from 2020-02-03 to 2024-12-09 = 10.5°C 2. London average mean temperature from 2020-02-03 to 2024-12-09 = 11.69°C 3. Overall average mean temperature from 2020-02-03 to 2024-12-09 = 14.61°C	1. Middlesbrough average mean temperature from 2020-02-03 to 2024-12-09 = 10.5°C 2. London average mean temperature from 2020-02-03 to 2024-12-09 = 11.69°C 3. Overall average mean temperature from 2020-02-03 to 2024-12-09 = 14.61°C	PASS	
17	<u>USER INPUT VALIDATION</u> Ensure the user is notified when wrong parameter is inputted into the plot_seven_day_precipitation_chart(city_names,	1. Entered pe as input for the start_date.	Invalid date format. Please use YYYY-MM-DD.	Invalid date format. Please use YYYY-MM-DD.	PASS	
		2. Entered 2020 as input for the start_date.	Invalid date format. Please use YYYY-MM-DD.	Invalid date format. Please use YYYY-MM-DD.	PASS	

	precipitation_averages, start_date, end_date) function	3. Entered 2021-01-2 as the input for the start_date.	Invalid date format. Please use YYYY-MM-DD.	Invalid date format. Please use YYYY-MM-DD.	PASS	
		4. Entered 2024-02-30 as the input for the start_date.	Invalid dates. Please choose dates within the range: 2020-01-01 - 2024-12-27.	Invalid dates. Please choose dates within the range: 2020-01-01 - 2024-12-27.	PASS	
		5. Entered 2022-02-02 as the input for the start_date. The end_date is automated calculated as the next 6 days after the date inputted as the start_date	Bar chart of average seven-day precipitation by city from 2022-02-02 to 2022-12-08	Bar chart of average seven-day precipitation by city from 2022-02-02 to 2022-12-08	PASS	
18	<u>CALCULATION TEST</u> Ensure the the plot_seven_day_precipitation_chart(city_names, precipitation_averages, start_date, end_date) plotting function output the correct value	1. Entered 2022-02-02 as the input for the start_date. The end_date is automated calculated as the next 6 days after the date inputted as the start_date	1. Middlesbrough average seven-day precipitation from 2022-02-02 to 2022-12-08 = 2.89 mm 2. Leeds average seven-day precipitation from 2022-02-02 to 2022-12-08 = 3.86 mm	1. Middlesbrough average seven-day precipitation from 2022-02-02 to 2022-12-08 = 2.89 mm 2. Leeds average seven-day precipitation from 2022-02-02 to 2022-12-08 = 3.86 mm	PASS	
19	<u>USER INPUT VALIDATION</u> Ensure the user is notified when wrong parameter is inputted into the plot_precipitation_	1. Entered ih as input for the city_id	Enter a valid numeric City ID.	Enter a valid numeric City ID.	PASS	
		2. Entered # as input for the city_id	Enter a valid numeric City ID.	Enter a valid numeric City ID.	PASS	

	histogram(city_name, city_weather_records, date_from, date_to) function	3. Entered 7 as the input for the city_id	Invalid City ID. Please choose from the following IDs: [1, 2, 3, 4, 5, 6].	Invalid City ID. Please choose from the following IDs: [1, 2, 3, 4, 5, 6].	PASS	
		4. Entered 4 as the input for the city_id and 2017 as input for the date_from and 2020 as the input for the date_to	Invalid dates. Please choose dates within the range: 2020-01-01 - 2024-12-27.	Invalid dates. Please choose dates within the range: 2020-01-01 - 2024-12-27.	PASS	
		5. Maintained 4 as the input for the city_id and 2022-08-08 as input for the date_from and 2026-08-07 as the input for the date_to	Invalid dates. Please choose dates within the range: 2020-01-01 - 2024-12-27.	Invalid dates. Please choose dates within the range: 2020-01-01 - 2024-12-27.	PASS	
		6. Maintained 4 as the input for the city_id and f as input for the date_from and h as the input for the date_to	Invalid dates. Please choose dates within the range: 2020-01-01 - 2024-12-27.	Invalid dates. Please choose dates within the range: 2020-01-01 - 2024-12-27.	PASS	
		7. Maintained 4 as the input for the city_id and 2022-08-08 as input for the date_from and 2021-08-07 as the input for the date_to	Start date cannot be after the end date. Please re-enter the dates.	Start date cannot be after the end date. Please re-enter the dates.	PASS	

		8. Maintained 4 as the input for the city_id and 2020-02-30 as input for the date_from and 2022-02-30 as the input for the date_to	Invalid dates. Please choose dates within the range: 2020-01-01 - 2024-12-27.	Invalid dates. Please choose dates within the range: 2020-01-01 - 2024-12-27.	PASS	
		9. Maintained 4 as the input for the city_id and 2021-03-03 as input for the date_from and 2024-10-01 as the input for the date_to	Histogram chart of the precipitation distribution for Toulouse from 2021-03-03 to 2024-10-01	Histogram chart of the precipitation distribution for Toulouse from 2021-03-03 to 2024-10-01	PASS	
20	<u>CALCULATION TEST</u> Ensure the the plot_precipitation_histogram(city_name, city_weather_records, date_from, date_to) plotting function output the correct value	4. Entered 4 as the input for the city_id and 2021-03-03 as input for the date_from and 2024-10-01 as the input for the date_to	1. Most frequent range of precipitation for Toulouse from 2021-03-03 to 2024-10-01 is 0 - 3 mm 2. The average precipitation for Toulouse from 2021-03-03 to 2024-10-01 = 2.26mm	1. Most frequent range of precipitation for Toulouse from 2021-03-03 to 2024-10-01 is 0 - 3 mm 2. The average precipitation for Toulouse from 2021-03-03 to 2024-10-01 = 2.26mm	PASS	
21	<u>USER INPUT VALIDATION</u> Ensure the user is notified when wrong parameter is inputted into the plot_scatter_plot(c	1. Entered b as input for the city_id	Enter a valid numeric City ID.	Enter a valid numeric City ID.	PASS	
		2. Entered # as input for the city_id	Enter a valid numeric City ID.	Enter a valid numeric City ID.	PASS	

	ity_name, city_weather_recor ds, date_from, date_to) function	3. Entered 9 as the input for the city_id	Invalid City ID. Please choose from the following IDs: [1, 2, 3, 4, 5, 6].	Invalid City ID. Please choose from the following IDs: [1, 2, 3, 4, 5, 6].	PASS	
		4. Entered 1 as the input for the city_id and 2020 as input for the date_from and 2024 as the input for the date_to	Invalid dates. Please choose dates within the range: 2020-01-01 - 2024-12-27.	Invalid dates. Please choose dates within the range: 2020-01-01 - 2024-12-27.	PASS	
		5. Maintained 1 as the input for the city_id and 2021-08-18 as input for the date_from and 2025-08-02 as the input for the date_to	Invalid dates. Please choose dates within the range: 2020-01-01 - 2024-12-27.	Invalid dates. Please choose dates within the range: 2020-01-01 - 2024-12-27.	PASS	
		6. Maintained 1 as the input for the city_id and f as input for the date_from and h as the input for the date_to	Invalid dates. Please choose dates within the range: 2020-01-01 - 2024-12-27.	Invalid dates. Please choose dates within the range: 2020-01-01 - 2024-12-27.	PASS	
		7. Maintained 1 as the input for the city_id and 2023-02-08 as input for the date_from and 2022-03-07 as the input for the date_to	Start date cannot be after the end date. Please re-enter the dates.	Start date cannot be after the end date. Please re-enter the dates.	PASS	

		8. Maintained 1 as the input for the city_id and 2021-02-30 as input for the date_from and 2023-02-30 as the input for the date_to	Invalid dates. Please choose dates within the range: 2020-01-01 - 2024-12-27.	Invalid dates. Please choose dates within the range: 2020-01-01 - 2024-12-27.	PASS	
		9. Maintained 1 as the input for the city_id and 2020-02-02 as input for the date_from and 2024-09-09 as the input for the date_to	Scatter plot chart of the precipitation Vs mean temperature for Middlesbrough from 2020-02-02 to 2024-09-09	Scatter plot chart of the precipitation Vs mean temperature for Middlesbrough from 2020-02-02 to 2024-09-09	PASS	
22	USER INPUT VALIDATION Ensure the user is notified when wrong parameter is inputted into the plot_temperature_line_chart(city_name, city_weather_records, date_from, date_to) function	1. Entered kp as input for the city_id	Enter a valid numeric City ID.	Enter a valid numeric City ID.	PASS	
		2. Entered ! as input for the city_id	Enter a valid numeric City ID.	Enter a valid numeric City ID.	PASS	
		3. Entered 8 as the input for the city_id	Invalid City ID. Please choose from the following IDs: [1, 2, 3, 4, 5, 6].	Invalid City ID. Please choose from the following IDs: [1, 2, 3, 4, 5, 6].	PASS	
		4. Entered 5 as the input for the city_id and 2021 as input for the date_from and 2024 as the input for the date_to	Invalid dates. Please choose dates within the range: 2020-01-01 - 2024-12-27.	Invalid dates. Please choose dates within the range: 2020-01-01 - 2024-12-27.	PASS	



		5. Maintained 5 as the input for the city_id and 2024-08-20 as input for the date_from and 2025-06-12 as the input for the date_to	Invalid dates. Please choose dates within the range: 2020-01-01 - 2024-12-27.	Invalid dates. Please choose dates within the range: 2020-01-01 - 2024-12-27.	PASS	
		6. Maintained 5 as the input for the city_id and g as input for the date_from and h as the input for the date_to	Invalid dates. Please choose dates within the range: 2020-01-01 - 2024-12-27.	Invalid dates. Please choose dates within the range: 2020-01-01 - 2024-12-27.	PASS	
		7. Maintained 5 as the input for the city_id and 2024-12-08 as input for the date_from and 2023-07-07 as the input for the date_to	Start date cannot be after the end date. Please re-enter the dates.	Start date cannot be after the end date. Please re-enter the dates.	PASS	
		8. Maintained 5 as the input for the city_id and 2020-02-30 as input for the date_from and 2024-02-30 as the input for the date_to	Invalid dates. Please choose dates within the range: 2020-01-01 - 2024-12-27.	Invalid dates. Please choose dates within the range: 2020-01-01 - 2024-12-27.	PASS	

		9.Maintained 5 as the input for the city_id and 2020-01-01 as input for the date_from and 2024-12-27 as the input for the date_to	Line charts for minimum temperature, maximum temperature and the mean temperature for Lagos from 2020-01-01 to 2024-12-27 that will be clear to read and use - without congestion.	Line charts for minimum temperature, maximum temperature and the mean temperature for Lagos from 2020-01-01 to 2024-12-27 that was congested on the chart.	FAIL	The matplotlib.widgets.Slider and matplotlib.backends.MouseButton modules was used to make the chart interactive by allowing users to adjust the visible data range of the chart.
		10. Entered 5 as the input for the city_id and 2020-01-01 as input for the date_from and 2024-12-27 as the input for the date_to	Line charts for minimum temperature, maximum temperature and the mean temperature for Lagos from 2020-01-01 to 2024-12-27 that will be clear to read and use - without congestion.	Line charts for minimum temperature, maximum temperature and the mean temperature for Lagos from 2020-01-01 to 2024-12-27 that was clear to read and use - without congestion.	PASS	

### PHASE 3 BLACKBOX TESTING

23	USER INPUT VALIDATION Ensure the user is notified when wrong parameter is inputted into the weather API request program in phase 3	Entered 2020 as the start date and 2021 as the end date, then clicked the program run code	time data '2020' does not match format '%Y-%m-%d'	time data '2020' does not match format '%Y-%m-%d'	PASS	
		Entered s as the start date and a as the end date, then clicked the program run code	time data 's' does not match format '%Y-%m-%d'	time data 's' does not match format '%Y-%m-%d'	PASS	

		Entered 2023-02-02 as the start date and 2023-01-02 as the end date, then clicked the program run code	The end date must be after the start date.	The end date must be after the start date.	PASS	
		Entered 2022-02-30 as the start date and 2025-01-30 as the end date, then clicked the program run code	day is out of range for month	day is out of range for month	PASS	
		Entered 2022-12-22 as the start date and 2025-02-02 as the end date, then clicked the program run code	The end date cannot be after two days before today's date: 2024-12-31.	The end date cannot be after two days before today's date: 2024-12-31.	PASS	
		Entered 2025-01-24 as the start date and 2025-02-03 as the end date, then clicked the program run code	The start date cannot be after two days before today's date: 2024-12-31.	The start date cannot be after two days before today's date: 2024-12-31.	PASS	

		Entered 2022-02-02 as the start date and 2022-02-05 as the end date, then clicked the program run code	Dates are valid! Database initialized successfully! Weather data for Lagos saved successfully! Weather data for Middlesbrough saved successfully! Weather data for London saved successfully! Weather data for Leeds saved successfully! Weather data for Paris saved successfully! Weather data for Toulouse saved successfully!	Dates are valid! Database initialized successfully! Weather data for Lagos saved successfully! Weather data for Middlesbrough saved successfully! Weather data for London saved successfully! Weather data for Leeds saved successfully! Weather data for Paris saved successfully! Weather data for Toulouse saved successfully!	PASS	
PHASE 4 BLACKBOX TESTING						
24	USER INPUT VALIDATION Ensure the user is notified when wrong parameter is inputted into the weather API request program in phase 4 - python script for the Tkinter GUI Program	Entered 2022 as the start date and 2024 as the end date, then clicked the program run code	time data '2022' does not match format '%Y-%m-%d'	time data '2022' does not match format '%Y-%m-%d'	PASS	
		Entered g as the start date and z as the end date, then clicked the program run code	time data 'g' does not match format '%Y-%m-%d'	time data 'g' does not match format '%Y-%m-%d'	PASS	

		Entered 2022-02-03 as the start date and 2021-01-02 as the end date, then clicked the program run code	The end date must be after the start date.	The end date must be after the start date.	PASS	
		Entered 2021-02-30 as the start date and 2024-02-30 as the end date, then clicked the program run code	day is out of range for month	day is out of range for month	PASS	
		Entered 2023-12-22 as the start date and 2025-05-02 as the end date, then clicked the program run code	The end date cannot be after two days before today's date: 2024-12-31.	The end date cannot be after two days before today's date: 2024-12-31.	PASS	
		Entered 2025-02-24 as the start date and 2026-12-03 as the end date, then clicked the program run code	The start date cannot be after two days before today's date: 2024-12-31.	The start date cannot be after two days before today's date: 2024-12-31.	PASS	
		Entered 2022-02-09 as the start date and 2022-02-12 as the end date, then clicked the program run code	Weather data fetched and saved successfully	Weather data fetched and saved successfully	PASS	
24	USER INPUT VALIDATION Ensure the user is notified when wrong parameter is inputted into the	Entered 2021 as the start date and 2022 as the end date, then clicked the program run code	time data '2021' does not match format '%Y-%m-%d'	time data '2021' does not match format '%Y-%m-%d'	PASS	

	weather API request program in phase 4 - the standalone Tkinter GUI application done with Pyinstaller	Entered w as the start date and x as the end date, then clicked the program run code	time data 'w' does not match format '%Y-%m-%d'	time data 'w' does not match format '%Y-%m-%d'	PASS	
		Entered 2024-09-03 as the start date and 2023-01-12 as the end date, then clicked the program run code	The end date must be after the start date.	The end date must be after the start date.	PASS	
		Entered 2020-02-30 as the start date and 2024-02-30 as the end date, then clicked the program run code	day is out of range for month	day is out of range for month	PASS	
		Entered 2022-11-21 as the start date and 2027-01-02 as the end date, then clicked the program run code	The end date cannot be after two days before today's date: 2024-12-31.	The end date cannot be after two days before today's date: 2024-12-31.	PASS	
		Entered 2025-03-24 as the start date and 2026-02-13 as the end date, then clicked the program run code	The start date cannot be after two days before today's date: 2024-12-31.	The start date cannot be after two days before today's date: 2024-12-31.	PASS	

		Entered 2024-05-10 as the start date and 2024-05-15 as the end date, then clicked the Update SDI-ICA Database button	Weather data fetched and saved successfully	Weather data fetched and saved successfully	PASS	
--	--	--	---	---	------	--