Software Testing Report

Sydney Airbnb Data Analysis Tool

s5291506 – Jamil Deris  
s5287914 – Tanish Dhir  
s5295636 – Arjan Dangol

Table of Contents

[1.0 Unit Tests 3](#_Toc49779837)

[2.0 Coverage Report 4](#_Toc49779838)

[3.0 Requirements Acceptance Testing 5](#_Toc49779839)

# Unit Tests

This specific format of the table enables you to make a comparison, between the real outcomes of every test case. This assists the developer in spotting any inconsistencies or problems within their code. If required the developer can expand this table by incorporating test cases that are relevant, to their project and provide supplementary information like the date when the tests were performed the name of the tester and any remarks or comments regarding the test results.

(In this table you fill out details about what unit tests you have done using the unittest module)

| **No** | **Test Case** | **Expected Results** | **Actual Results** | **Status** |
| --- | --- | --- | --- | --- |
| **1.** | **Airbnb Listing Data Retrieval Test Cases** |  |  |  |
| 1.1 | Test List Airbnb Listing with Valid Data | DataFrame with valid data and not empty | DataFrame with valid data and not empty | PASSED |
| 1.2 | Test List Airbnb Listing with Invalid Suburb | Empty DataFrame | Empty DataFrame | PASSED |
| 1.3 | Test List Airbnb Listing with Invalid Date | Empty DataFrame | Empty DataFrame | PASSED |
| 1.4 | Test List Airbnb Listing with Invalid Date Range | Empty DataFrame | Empty DataFrame | PASSED |
| **2.** | **Property Price Distribution Plot Test Cases** |  |  |  |
| 2.1 | Test Show Plot Returns Figure | Result should return a 'matplotlib.figure.Figure' | Returned object is an instance of 'Figure' | PASSED |
| 2.2 | Test Show Plot Has Correct Xlabel | Xlabel of the plot should be 'Average Price' | Xlabel of the plot is 'Average Price' | PASSED |
| 2.3 | Test Show Plot Has Correct Ylabel | Ylabel of the plot should be 'Frequency' | Ylabel of the plot is 'Frequency' | PASSED |
| 2.4 | Test Show Plot Has Non-negative Prices | All price values should be non-negative | All price values are non-negative | PASSED |
| **3.** | **Testing Keyword-Specific Listing Retrieval Functionality** |  |  |  |
| 3.1 | Test Retrieve Keyword Specific Listing Returns DataFrame | The function should return a Pandas DataFrame. | The function returned a Pandas DataFrame. | PASSED |
| 3.2 | Test Retrieve Keyword Specific Listing Contains Keyword | The result DataFrame should contain the specified keyword. | The keyword was found in at least one row of the result DataFrame. | PASSED |
| 3.4 | Test Retrieve Keyword Specific Listing Has At Least One Coulmn | The keyword should appear in at least one of the relevant columns (name, summary, space, description, notes, house\_rules, amenities). | The keyword was found in at least one relevant column of the result DataFrame. | PASSED |
| 3.5 | Test Retrieve Keyword Specific Listing Empty Result | The result should be an empty DataFrame if the specified keyword does not exist. | The function returned an empty DataFrame as expected. | PASSED |
| **4.** | **Testing Cleanliness Analysis Functionality** |  |  |  |
| 4.1 | Test Retrieve Reviews Listing Returns DataFrame | Result should be a Pandas DataFrame. | Result is a Pandas DataFrame. | PASSED |
| 4.2 | Test Retrieve Reviews Listing Contains Cleanliness Mentions | All rows in the result should have cleanliness mentions. | All rows in the result have cleanliness mentions. | PASSED |
| 4.3 | Test Retrieve Reviews Listing within Date Range | Dates in the result should be within the specified range. | Dates in the result are within the specified range. | PASSED |
| 4.4 | Test Retrieve Reviews Listing Empty Result | Result should be empty if no data is found within the range. | Result is empty as there are no reviews within the range. | PASSED |
| **5.** | **Testing Room Usage Analysis Functionality** |  |  |  |
| 5.1 | Test Retrieve Room Usage Listings Returns DataFrame | Result should be a Pandas DataFrame. | Result is a Pandas DataFrame. | PASSED |
| 5.2 | Test Retrieve Room Usage Listings Non-Negative Usage Count | All usage counts should be non-negative integers. | All usage counts are non-negative integers. | PASSED |
| 5.3 | Test Retrieve Room Usage Listings Not Empty | Result should not be empty. | Result is not empty. | PASSED |
| 5.4 | Test Retrieve Room Usage Listings Sorted by Listing Id | The 'listing\_id' column should be in ascending order. | 'listing\_id' column is in ascending order. | PASSED |

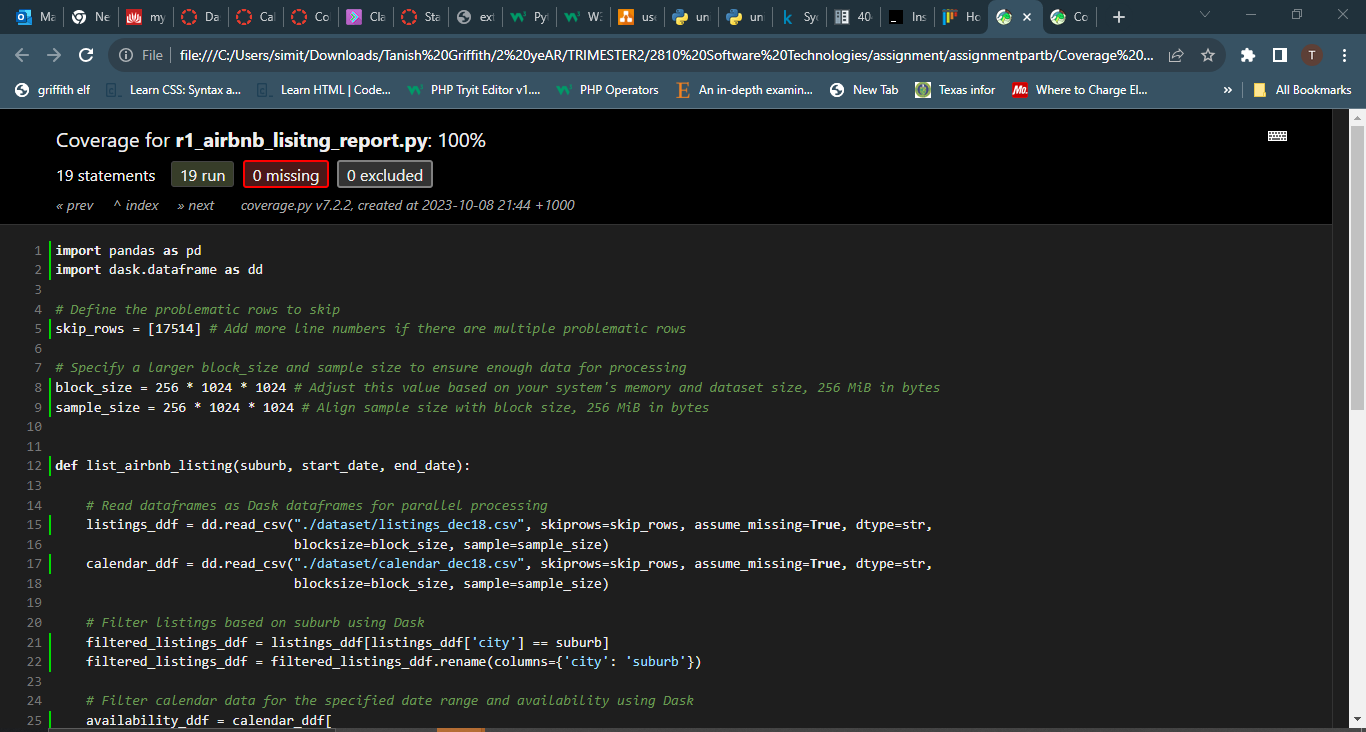
# Coverage Report

A description of the coverage of your unit tests, including how you evaluated coverage (function, statement, branch, condition)

Our group made the following report by connecting the functionality files and testing them using various functions. The screenshots below state that:

1. Listings function

Following are screenshots of our listings function file(Figure 1.1 to 1.3)



**Figure1.1**

A screenshot of a computer

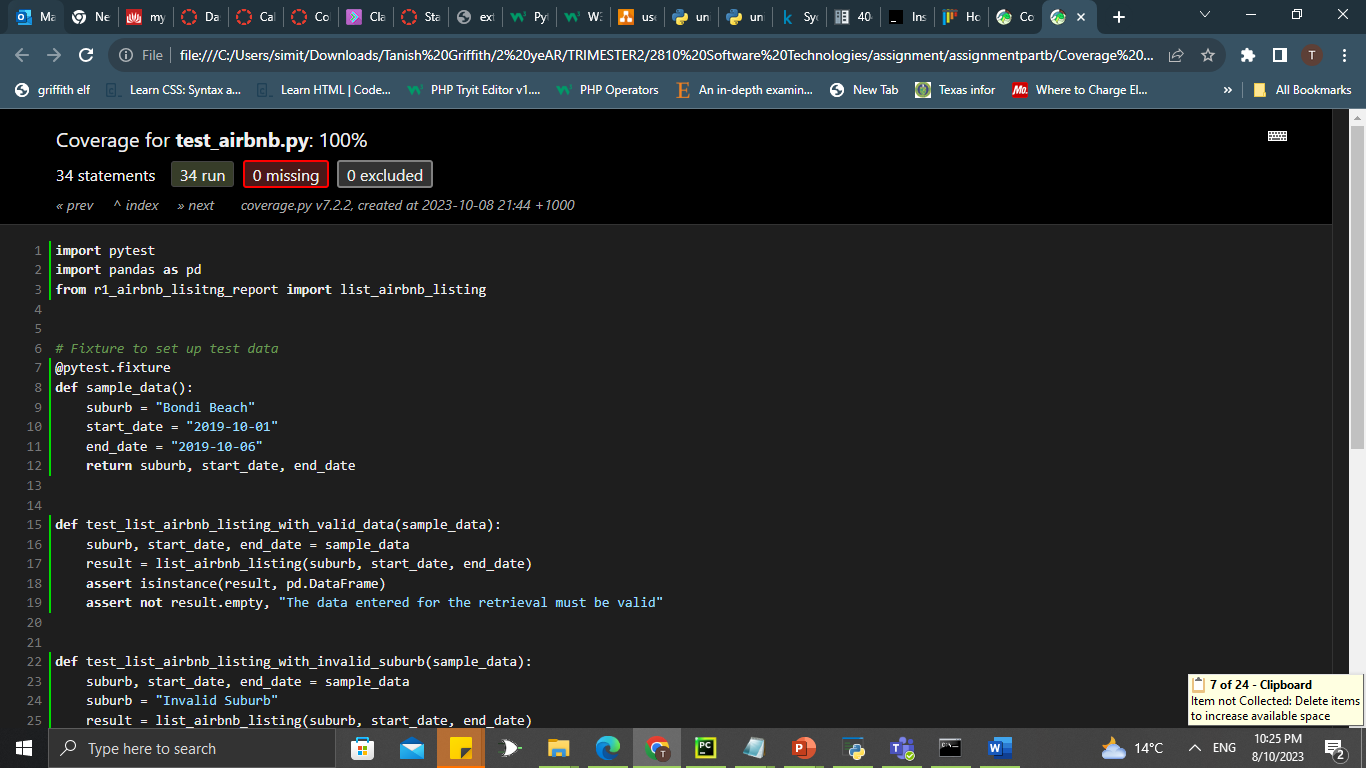
Description automatically generated **Figure1.2**

A screenshot of a computer

Description automatically generated

**Figure1.3**

Next screenshots are based on coverage report of the listings file (Figure 1.4 to 1.5):



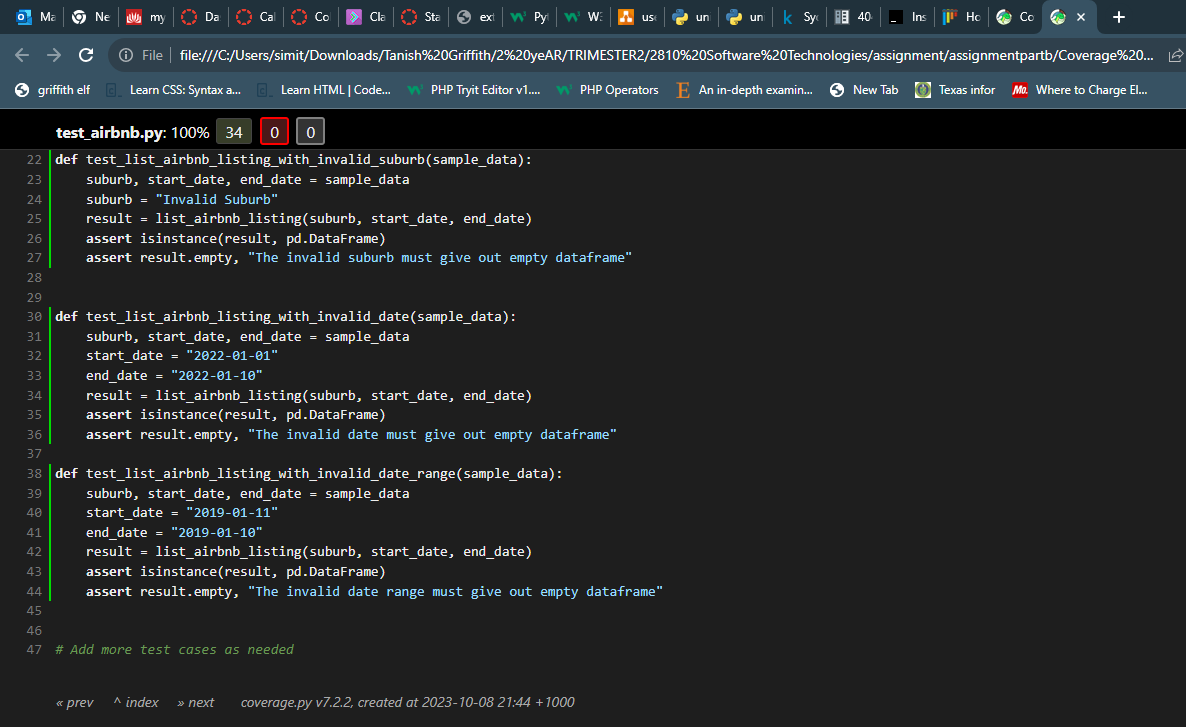
**Figure 1.4**

In the above file, sample data is provided for inputs which is suburb, start, end date. This data is used to test functions which show different kinds of user input. Test\_list\_airbnb\_lisiting\_with\_valid\_data function checks whether sample data by user is valid or not by using assert isinstance function.

Test\_list\_airbnb\_listing\_with\_invalid\_data function checks if entered suburb is invalid or not. If invalid, then it returns nothing.

Test\_list\_airbnb\_listing\_with\_invalid\_date makes sure if any one of the given dates is not in between 7 December 2018 to 6 December 2019. If, it is out of range it returns nothing.

Test\_list\_airbnb\_listing\_with\_invalid\_date\_range makes sure that the given dates are in between 7 December 2018 to 6 December 2019. If any one of them is out of range, it returns nothing.



**Figure 1.5**

1. Price Chart function: The following screenshots are based on price chart function on the tool.(Figure 2.1 to 2.2)

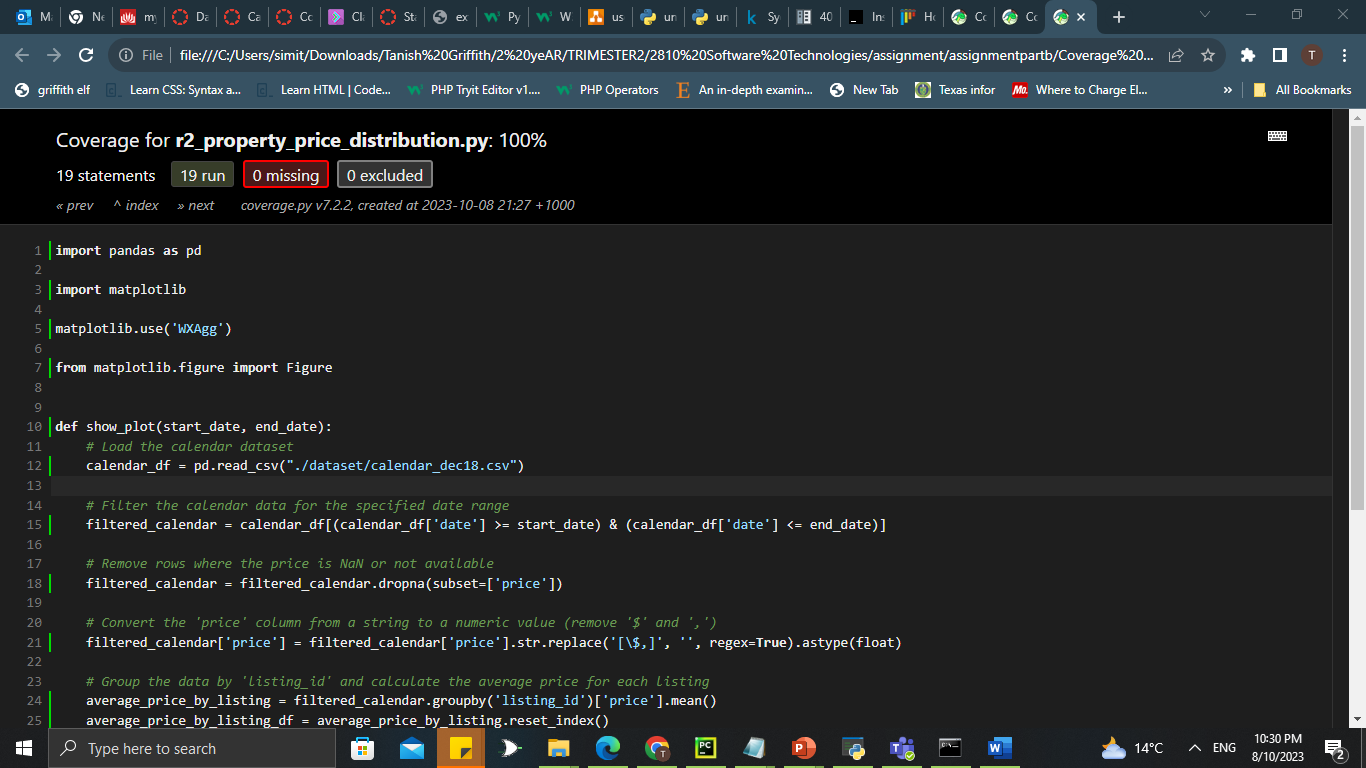


Figure 2.1

A screenshot of a computer

Description automatically generated

Figure 2.2

The following screenshots are of testing file for price chart function Figure 2.3 to 2.4 .

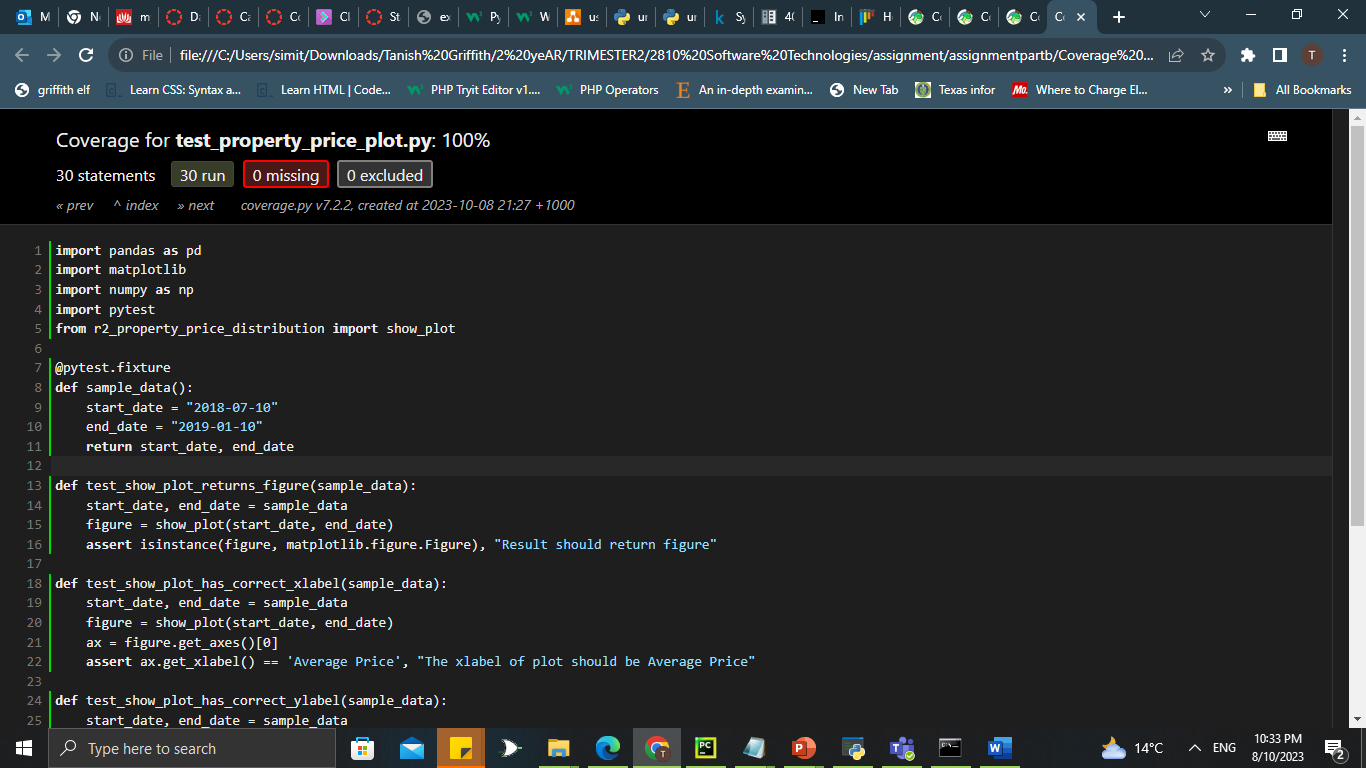


Figure 2.3

A screenshot of a computer

Description automatically generated

Figure 2.4

In the above screenshots sample data is taken of dates within the specified date range. Test\_show\_plot\_returns\_figure displays chart based on start and end date.

Test\_show\_plot\_has\_correct\_xlabel makes sure x axis of figure has label of average price.

Test\_show\_plot\_has\_correct\_ylabel makes sure y axis of figure has label of frequency.

Test\_show\_plot\_has\_non\_negative\_prices checks all values of plot are greater than 0.

1. Keyword Search function: The following screenshots show keyword search function of the program Figure 3.1 to 3.2 .

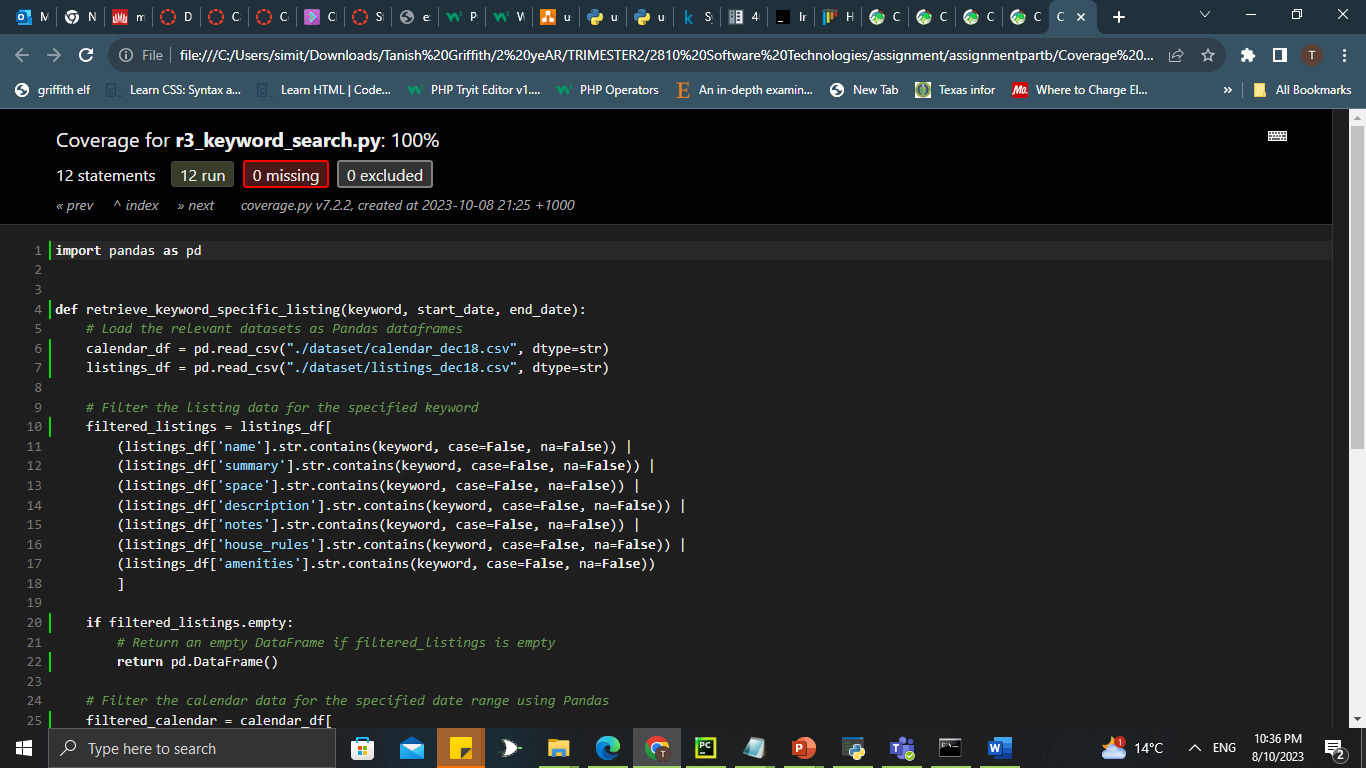


Figure 3.1

A screenshot of a computer

Description automatically generated

Figure 3.2

The following screenshots show test file for keyword search function in Figure 3.3 to 3.4 .

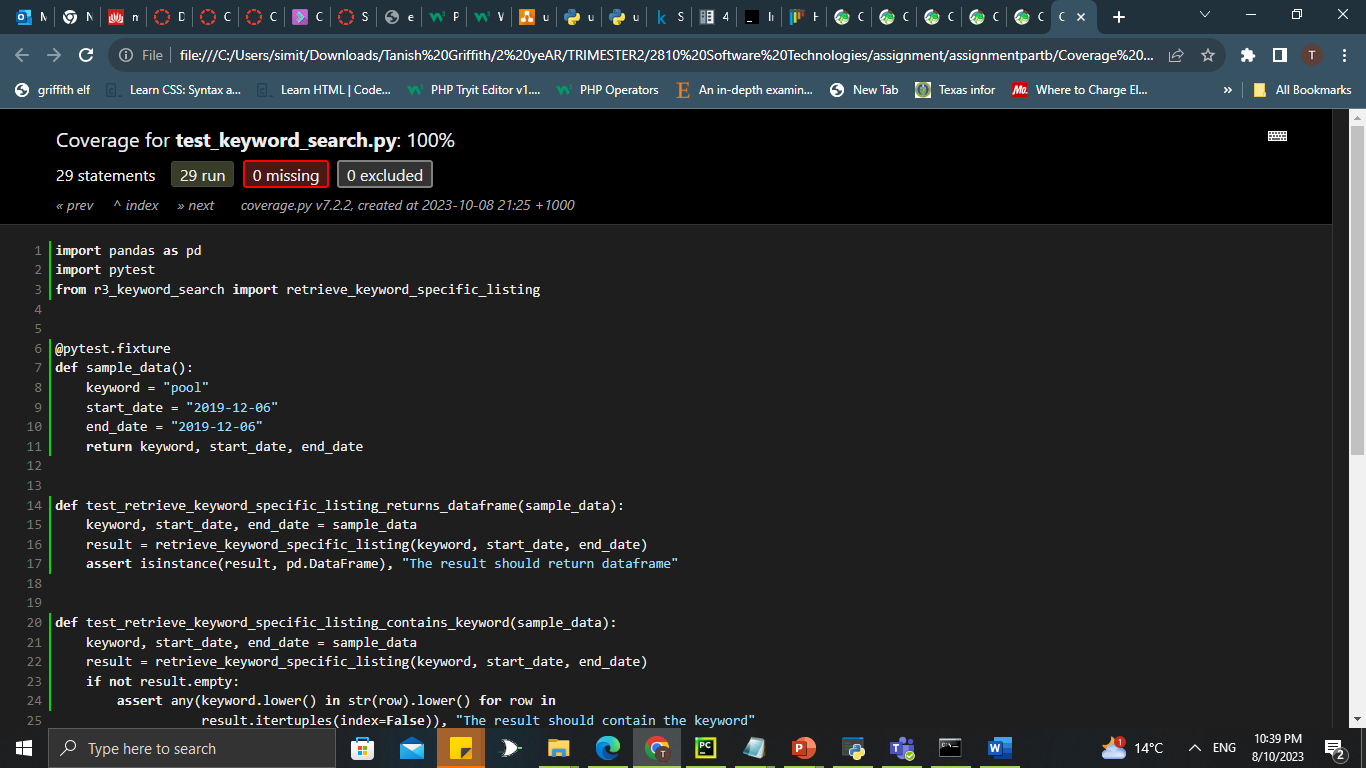


Figure 3.3

A screenshot of a computer

Description automatically generated

Figure 3.4

1. Review Analysis: The following screenshots show reviews analysis function in Figure 4.1 to 4.2 .

A screenshot of a computer program

Description automatically generatedFigure 4.1

A screenshot of a computer

Description automatically generated

Figure 4.2

The following screenshots show test file for review analysis function in figure 4.3 to 4.4 .

A screenshot of a computer

Description automatically generated

Figure 4.3

A screenshot of a computer

Description automatically generated

Figure 4.4

1. Room Usage Analysis: The following screenshots shows room usage function of the program in Figure 5.1 to 5.2 .

A screenshot of a computer

Description automatically generated

Figure 5.1

A screenshot of a computer

Description automatically generated

Figure 5.2

The following screenshots are from test file for room usage analysis function from Figure 5.3 to 5.4 .

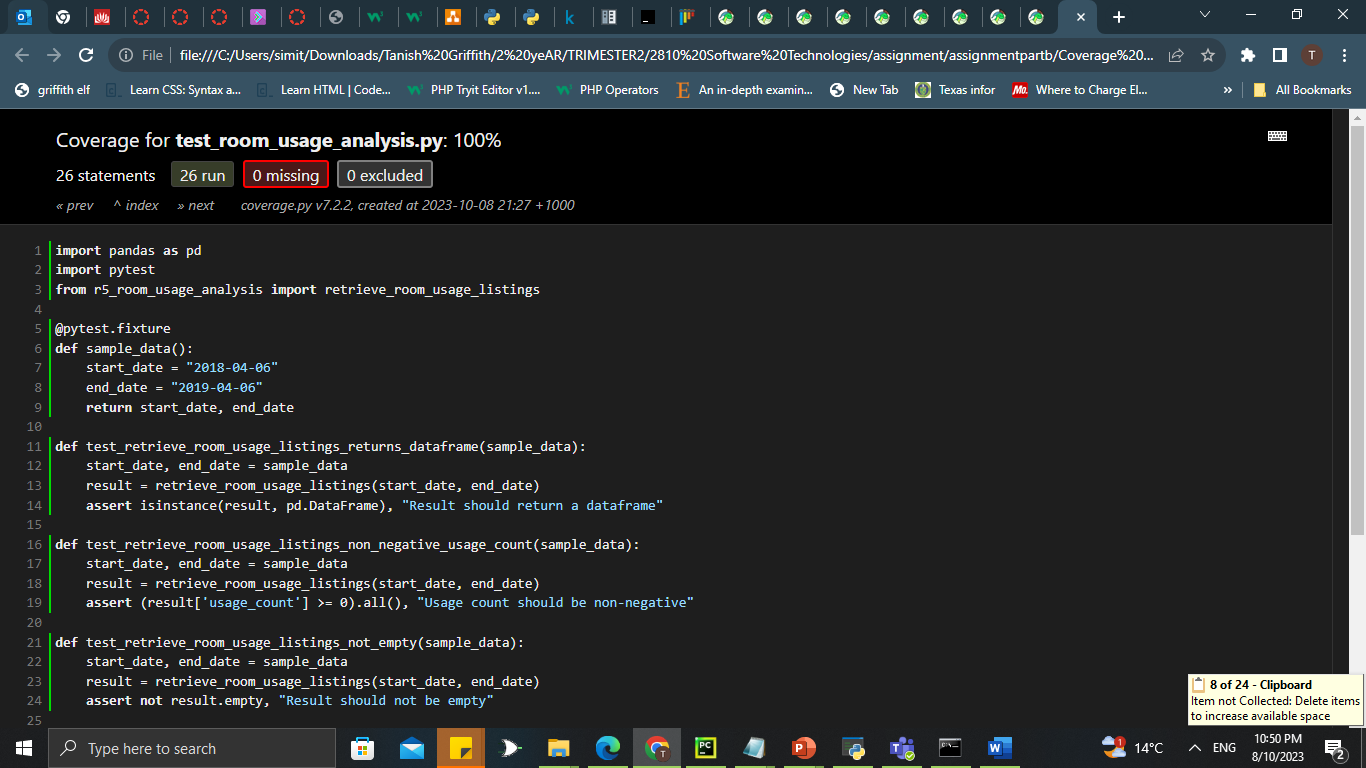


Figure 5.3

A screenshot of a computer

Description automatically generated

Figure 5.4

# Requirements Acceptance Testing

(You will need to fill out the column on the left with the requirements listed in software design documents and the columns on the right with the results of your own testing)

| **Software  Requirement No** | **Test** | **Implemented (Full /Partial/ None)** | **Test Results (Pass/ Fail)** | **Comments (for partial implementation or failed test results)** |
| --- | --- | --- | --- | --- |
| 1 | The program shall accept user input through a list of suburbs | Full | Pass |  |
| 2 | The program shall show what rooms are available and show brief description about them | Partial | Pass | In our program, using listings function user can see rooms with their lisitng id. We were unable to show brief description about them. |
| 3 | It shall have a date option to see for a particular date | Full | Pass |  |
| 4 | It shall present a price for each room including details of payment | None |  | We were unable to show this feature in our final Graphical user interface. |
| 5 | It shall display number of customers who gave feedback chart of a room on basis of cleanliness and similar keywords like environment, tidy, etc. These will have values which indicate what is the review of the room. These are selected as tourists will look for these features in a room as it makes their visit a pleasant experience. | Full | Pass |  |
| 6 | The program shall present a price distribution chart when user selects a suburb and click on view price distribution chart | Partial | Pass | It is implemented partially as we decided to create price chart based on date range and did not provide option of choosing suburb to user. |
| 7 | The program shall present how many times a property has been used for a user selected date | Full | Pass |  |
|  |  |  |  |  |
|  |  |  |  |  |