A
MINI PROJECT REPORT
ON

Weather Forecasting System

Submitted By

Ms. Shelar Prajakta Ishwar 117
Mr. Shelke Abhishek Ramnath 118
Mr. Shinde Bhavesh Vitthalrao 119

Guided By

Prof. A.S.Bodhe

(Savitribai Phule Pune University, Pune)

In the academic year 2021-22

Department of Computer Engineering

Sanjivani College of Engineering

Kopargaon - 423 603.

Sanjivani College of Engineering, Kopargaon

CERTIFICATE

This is to certify that

Ms. Shelar Prajakta Ishwar 117
Mr. Shelke Abhishek Ramnath 118
Mr. Shinde Bhavesh Vitthalrao 119

(B.E. Computer)

Has successfully completed their project report on

Weather Forecasting System

Towards the partial fulfilment of

Bachelor's Degree In Computer Engineering

During the academic year 2021-22

Prof. A.S.Bodhe
[Guide]

Dr. D.B.KSHIRSAGAR
[H.O.D. Comp Engg]

Dr. A.G.THAKUR
[Director]

Acknowledgement

The entire session of project completion phase so far was a great experience providing us with great insight and innovation into learning various classification concepts and achievement of it. As is rightly said, for the successful completion of any work, people are the most important asset my seminar would not be materialized without the cooperation of many of the people involved.

First and foremost, I am very much thankful to my respected project guide **Prof. A.S.Bodhe** and for their leading guidance and sincere efforts in finalizing this topic. They took deep interest in correcting the minor mistakes and guided us through my journey so far. Also they has been persistent source of inspiration for me.

I am also very thankful of **Dr. D.B.Kshirsagar**, Head of Dept. of Computer Engineering for the symmetric guidance and providing necessary facilities and I Express deep gratitude to all the staff members and our department's technical Staff for providing me needed help.

Ms. Shelar Prajakta Ishwar Janardan

Mr. Shelke Abhishek Ramnath

Mr. Shinde Bhavesh Vitthalrao

B. E. Computer

Contents

List of Figures

Chapter 1 Abstract

Weather forecasting is the application of science and technology to predict the state of the atmosphere for a given location. Ancient weather forecasting methods usually relied on observed patterns of events, also termed pattern recognition. For example, it might be observed that if the sunset was particularly red, the following day often brought fair weather. However, not all of these predictions prove reliable. Here this system will predict weather based on parameters such as temperature, humidity and wind. This system is a web application with effective graphical user interface. User will login to the system using his user ID and password. User will enter current temperature; humidity and wind, System will take this parameter and will predict weather from previous data in database. The role of the admin is to add previous weather data in database, so that system will calculate weather based on these data. Weather forecasting system takes parameters such as temperature, humidity, and wind and will forecast weather based on previous record therefore this prediction will prove reliable. This system can be used in Air Traffic, Marine, Agriculture, Forestry, Military, and Navy etc.

Chapter 2 Introduction

2.1 Introduction.

Weather forecasting is the application of science and technology to predict the state of the atmosphere for a given location.Ancient weather forecasting methods usually relied on observed patterns of events, also termed pattern recognition. For example, it might be observed that if the sunset was particularly red, the following day often brought fair weather.However, not all of these predictions prove reliable. Here this system will predict weather based on parameters such as temperature, humidity and wind. This system is a web application with effective graphical user interface. User will login to the system using his user ID and password. User will enter current temperature; humidity and wind, System will take this parameter and will predict weather from previous data in database. The role of the admin is to add previous weather data in database, so that system will calculate weather based on these data. Weather forecasting system takes parameters such as temperature, humidity, and wind and will forecast weather based on previous record therefore this prediction will prove reliable. This system can be used in Air Traffic, Marine, Agriculture, Forestry, Military, and Navy etc. Forecasting the temperature and rain on a particular day and date is the main aim of this paper. In the paper we forecast rain and temperature for Europe; year up to 2051 and also we forecast temperature of world; year up to 2100.Our paper is aimed to provide real time weather forecast service at finest granularity level with recommendations. We grab user's location (longitude, latitude) using GPS data service whenever user requests for our services. Our system will process the users query and will mine the data from our repository to draw appropriate results. Users will be provided with recommendations also and that is the key facility of our service. Personalized forecast is generated for each individual user based on their location.

Chapter 3 Manual Testing Importance

In general, testing is finding out how well something works. In terms of human beings, testing tells what level of knowledge or skill has been acquired. In computer hardware and software development, testing is used at key checkpoints in the overall process to determine whether objectives are being met. Software testing, depending on the testing method employed, can be implemented at any time in the development process. Software testing can be stated as the process of validating and verifying that a software program/application/product.

3.1 Manual Testing

Manual testing is a software testing process in which test cases are executed manually without using any automated tool. All test cases executed by the tester manually according to the end user's perspective. It ensures whether the application is working, as mentioned in the requirement document or not. Test cases are planned and implemented to complete almost 100 percent of the software application. Test case reports are also generated manually. Manual Testing is one of the most fundamental testing processes as it can find both visible and hidden defects of the software. The difference between expected output and output, given by the software, is defined as a defect. The developer fixed the defects and handed it to the tester for retesting. Manual testing is mandatory for every newly developed software before automated testing. This testing requires great efforts and time, but it gives the surety of bug-free software. Manual Testing requires knowledge of manual testing techniques but not of any automated testing tool. Manual testing is essential because one of the software testing fundamentals is "100% automation is not possible." There are various methods used for manual testing. Each technique is used according to its testing criteria. Types of manual testing are given below:

• **White Box Testing :-** The white box testing is done by Developer, where they check every line of a code before giving it to the Test Engineer. Since the

code is visible for the Developer during the testing, that's why it is also known as White box testing.

• **Black Box Testing :-**The black box testing is done by the Test Engineer, where they can check the functionality of an application or the software according to the customer /client's needs. In this, the code is not visible while performing the testing; that's why it is known as black-box testing.

• **Gray Box Testing : -** Gray box testing is a combination of white box and Black box testing. It can be performed by a person who knew both coding and testing. And if the single person performs white box, as well as black-box testing for the application, is known as Gray box testing.

3.2 How to perform manual testing

A complete manual testing process consists of the following steps:

• **Step 1:** First, gather the requirements using the requirement analysis step. Once we gather and understand the requirements, we know what the expected behavior is and what we need to test, and when we say we have found the defect.

• **Step 2:** Secondly, once we understand the requirements, we identify and draft the test cases that will cover all the requirements contained in the project documentation. Additionally, the test cases help us follow a sequence to test functionality and various test scenarios such that we cover the entire application and check expected results.

• **Step 3:** Once test cases are ready, the tester has to review the test cases with the team leader and with the client if need be. By examining the test cases, we will find glitches, if any, and correct them before executing the test cases.

• **Step 4:** Once test cases are ready, and the test environment sets, we execute the test cases one by one.

Each test case will have one of the following states:

Passed: If the scenario under test works as expected.

Failed: If the working is not as expected.

Skipped: If the test case cannot complete. It may be because of some limitations or unforeseen circumstances.

• **Step 5:** As the test cases execute, we have to report the identified bugs and defects to the concerned developer and submit a bug report.

• **Step 6:** Finally, we create a detailed test report that will include detailed information on how many defects or bugs we found, how many test cases need to be rerun, how many test cases failed, and how many we skipped. Once we

fix the bugs and defects, execute the test cases that could not verify the fixed bugs.

3.3 Importance Of Manual Testing

1. **Manually Testing Tests From a Human Perspective:-** Human testers can quickly identify when something looks "off." Automated test scripts don't pick up these visual issues. When a tester interacts with software as a user would, they're able to discover usability issues and user interface glitches. Automated test scripts can't test for these things.

2. **Exploratory Testing Can Only Be Done Manually:-** Automated tests only perform the actions that you tell them to. They require planning and preparation to write, which restricts the test to certain boundaries. These boundaries mean there isn't any room to stray from the written test to truly "explore" the application. Exploratory testing (or ad hoc testing) gives us the opportunity to answer questions like, "what happens if I do this?" It enables us to carve our own path throughout the test with little to no boundaries.

3. **Automated Tests Can Contain Errors and Holes:-** Just like code can have bugs, automated test scripts can also have bugs. This means that automated testing has the potential to report false positives and false negatives. By including a human touch throughout the testing process, these errors are avoided.

4. **Some Scenarios Are Not Technically Feasible to Automate or Cost Too Much :-** Let'stake, for example, an iPad app that relies heavily on tap gestures. Automating the "tap" can not only be costly but also may not be the most accurate test compared to a human's finger touching the user interface. It often makes more sense to manually test certain features. By the time you find a resource to write an automated test, a manual test could be completed with bugs already fixed. Sometimes, it makes more sense to just go the "manual" route.

5. **Manual Testing Helps Us Understand the Whole Problem :-** Over time, automated testing can save time. It is great for getting quick results on a broad level, but manual testing allows us to understand the problem on a conceptual and emotional level. It connects us with the end-user and introduces us to a level of empathy automated testing doesn't provide.

3.4 Advantages

• Manual testing of an application identifies most of the issues, including the look and feel issues of the application.
• Visual components like text, layout, other components can easily be accessed by the tester, and UI and UX issues can be detected.
• It usually has a low cost of operation as we do not use any tools or high-level skills.
• It is well-suited in case we make some unplanned changed to the application as it is adaptable.
• Humans can observe, judge, and also provide intuition in case of manual tests, and this is useful when it comes to user-friendliness or rich customer experience.

3.5 Disadvantages

• Manual testing is time-consuming.
• It isn't easy to find size difference and color combination of GUI objects using a manual test.
• Load testing and performance testing is impractical in the manual tests.
• When there is a large number of tests, then running tests manually is a very time consuming job.

Chapter 4 SRS Plan

4.1 Introduction

4.1.1 Purpose

The purpose of this document is to present a detailed description of the Weather Forecasting System. It will explain the purpose and the purpose and features of the system, the interfaces of the system, what the system will do, the constraints under which it must operate and how the system will react to external stimuli. This document is intended for both the stakeholders and developers of the system and will be liable for the approval or disapproval of the project by the community of organization.

4.1.2 Scope

The project mainly focuses on weather forecasting .We will use an API key of open-weather.com to get weather updates . Application will retrieve weather update of location entered by user in search bar .

4.1.3 Objective

Forecasting the temperature and rain on a particular day and date is the main aim of this paper. In the paper we forecast rain and temperature for Europe; year up to 2051 and also we forecast temperature of world; year up to 2100.Our paper is aimed to provide real time weather forecast service at finest granularity level with recommendations. We grab user's location (longitude, latitude) using GPS data service whenever user requests for our services. Our system will process the users query and will mine the data from our repository to draw appropriate results. Users will be provided with recommendations also and that is the key facility of our service. Personalized forecast is generated for each individual user based on their location.

4.1.4 Definitions and Abbreviations

Term	Definition
API	Application Programming Interface
CSS	Cascading Style Sheet
HTML	Hyper Text Markup Language
IDE	Integrated Development Environment
dev	development
User	A person who is maintaining the inventory of organization.
SRS	A document that completely describes all of the functions of a proposed system and constraints under which it must operate.
Stakeholder	Any person with an interest in the project who is not a developer

Table 4.1: Term Index

4.2 General Description

4.2.1 Product Perspective

Weather forecasting is the application of science and technology to predict the state of the atmosphere for a given location.Ancient weather forecasting methods usually relied on observed patterns of events, also termed pattern recognition. For example, it might be observed that if the sunset was particularly red, the following day often brought fair weather.However, not all of these predictions prove reliable. Here this system will predict weather based on parameters such as temperature, humidity and wind. This system is a web application with effective graphical user interface. User will login to the system using his user ID and password. User will enter current temperature; humidity and wind, System will take this parameter and will predict weather from previous data in database. The role of the admin is to add previous weather data in database, so that system will calculate weather based on these data. Weather forecasting system takes parameters such as temperature, humidity, and wind and will forecast weather based on previous record therefore this prediction will prove reliable. This system can be used in Air Traffic, Marine, Agriculture, Forestry, Military, and Navy etc. Forecasting the temperature and rain on a particular day and date is the main aim of this paper. In the paper we forecast rain and temperature for Europe; year up to 2051 and also we forecast temperature of world; year up to 2100.Our paper is aimed to provide real time weather forecast service at finest granularity level with recommendations. We grab user's location (longitude, latitude) using GPS data service whenever user requests for our services. Our system will process the users

query and will mine the data from our repository to draw appropriate results. Users will be provided with recommendations also and that is the key facility of our service. Personalized forecast is generated for each individual user based on their location.

4.2.2 Product Function

Weather Forecasting System must be designed to meet the dictates of the users. System should provide accurate weather information to user .

4.2.3 User Classes and Characteristics

The product is a general-purpose application software where any user can access the software to gather information about the weather based on a current or a previous data analysis. In this product the software makes a request to the server to access the current data-set through which the data is analyzed using various machine learning algorithms.

4.2.4 Assumption and Dependencies

According to our view that system will Contain module like :

- The software product is dependent on a data-set that is been retrieved from the server using various commands.
- The product will work based on the algorithm that has been discussed above.

4.2.5 Design and Implementation Constraints

When the project team is started to design the system, at that time the team always tries to meet all the requirements of their client properly. But some time client's make the job difficult for the team to design the system by raising some last moment requirements. This is considered as main constraints of designing a system.

4.3 Specific Requirements

4.3.1 Functional Requirements

Accessing Database

- The system should allow administrator to add historical weather data.

- The system should be able to recognize patterns in temperature, humidity, and wind with use of historical data.

Prediction

- System Should periodically apply prediction algorithm or models on obtained data and store results to central database .
- System shall obtain and display confidence value for each prediction given to user .

4.4 Non Functional Requirements.

4.4.1 Performance Requirements

The proposed software that we are going to develop will be used as the general-purpose application software .Therefore , it is expected that the database would perform functionally all the requirements that are specified by the user .

4.4.2 Safety Requirements

The database may get crashed at any certain time due to virus or operating system failure . Therefore ,it is required to take the database .

4.4.3 Security Requirements

We are going to develop a secured database for the user .Software Quality Attributes .The Quality of the database is maintained in such way that it can be very user friendly to all the users .

4.4.4 Hardware Requirements

The system requires a database in order to store persistent data .

4.4.5 Software Constraints

The development of the system will be constrained by the availability of required software such as servers , dataset and development tools .

4.4.6 System Non-Functional Requirements

- System should be able to run with core functionality from computer system .
- System should be able to show interactive animations to users regarding current and future climatic condition.

4.4.7 User Non-Functional Requirements

- System shall allow for users to get prediction for weather within almost two mouse clicks .
- System should ensure that features that do not require a user to be logged in .

Chapter 5 UML Diagrams

5.1 Context Level Diagram

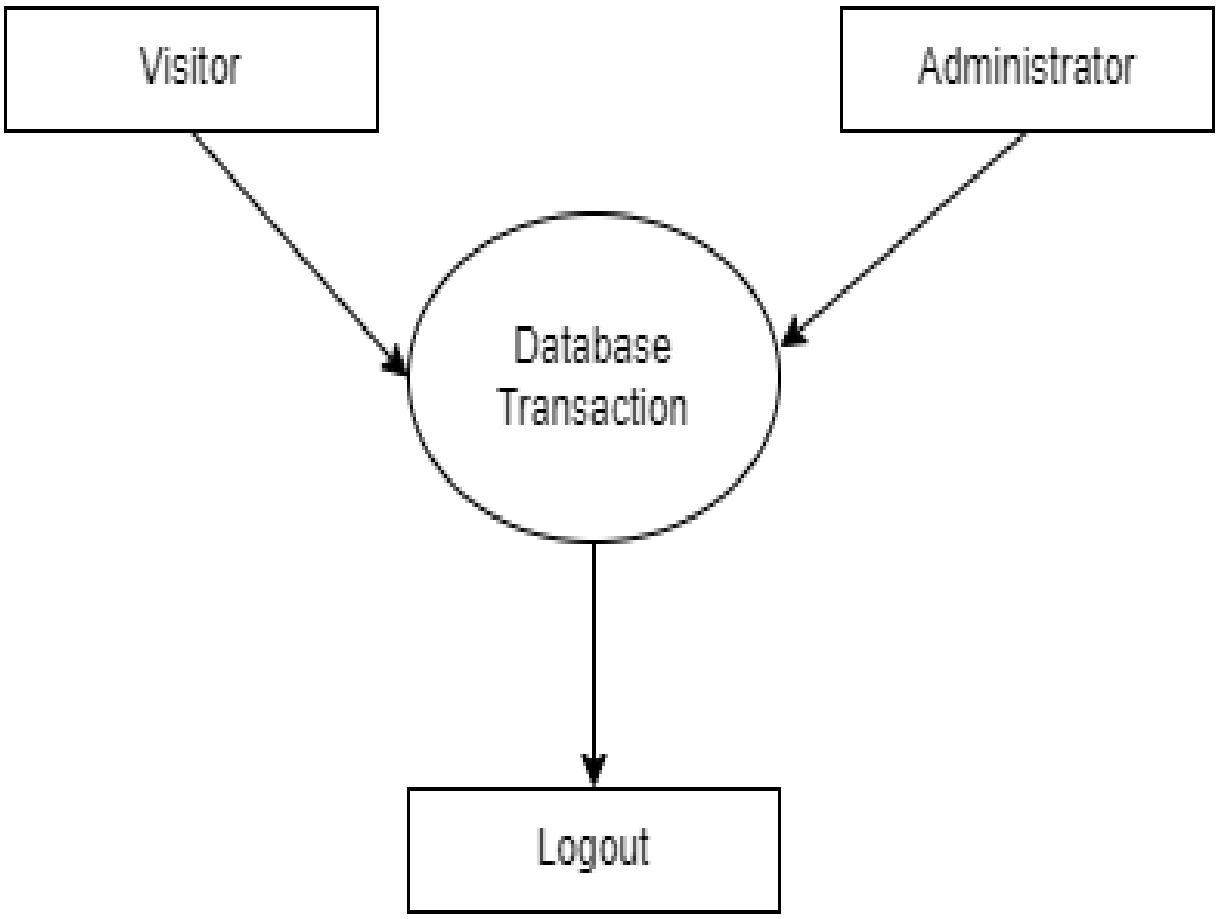

Figure 5.1: Context Level Diagram

5.2 DFD1

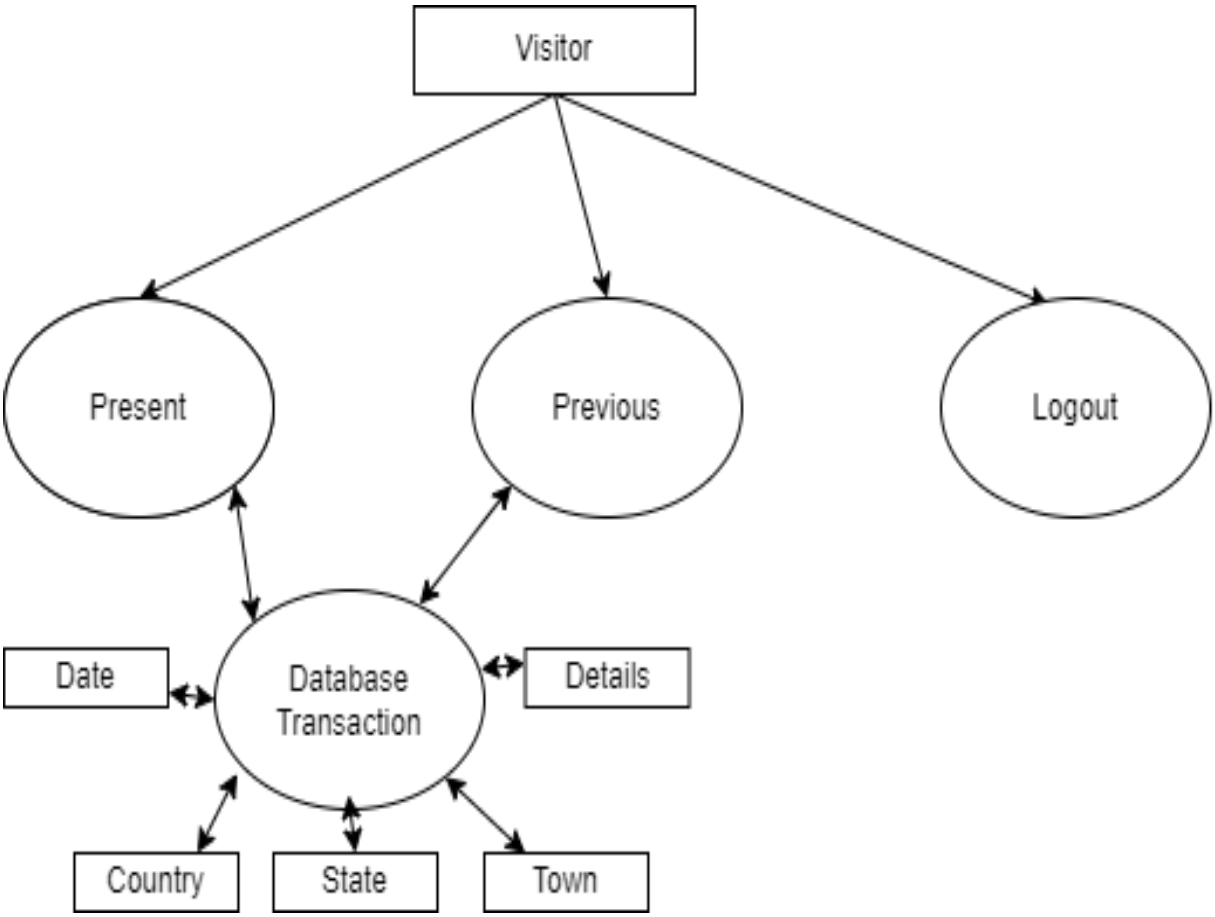

Figure 5.2: Data Flow Diagram Level 1

5.3 DFD2

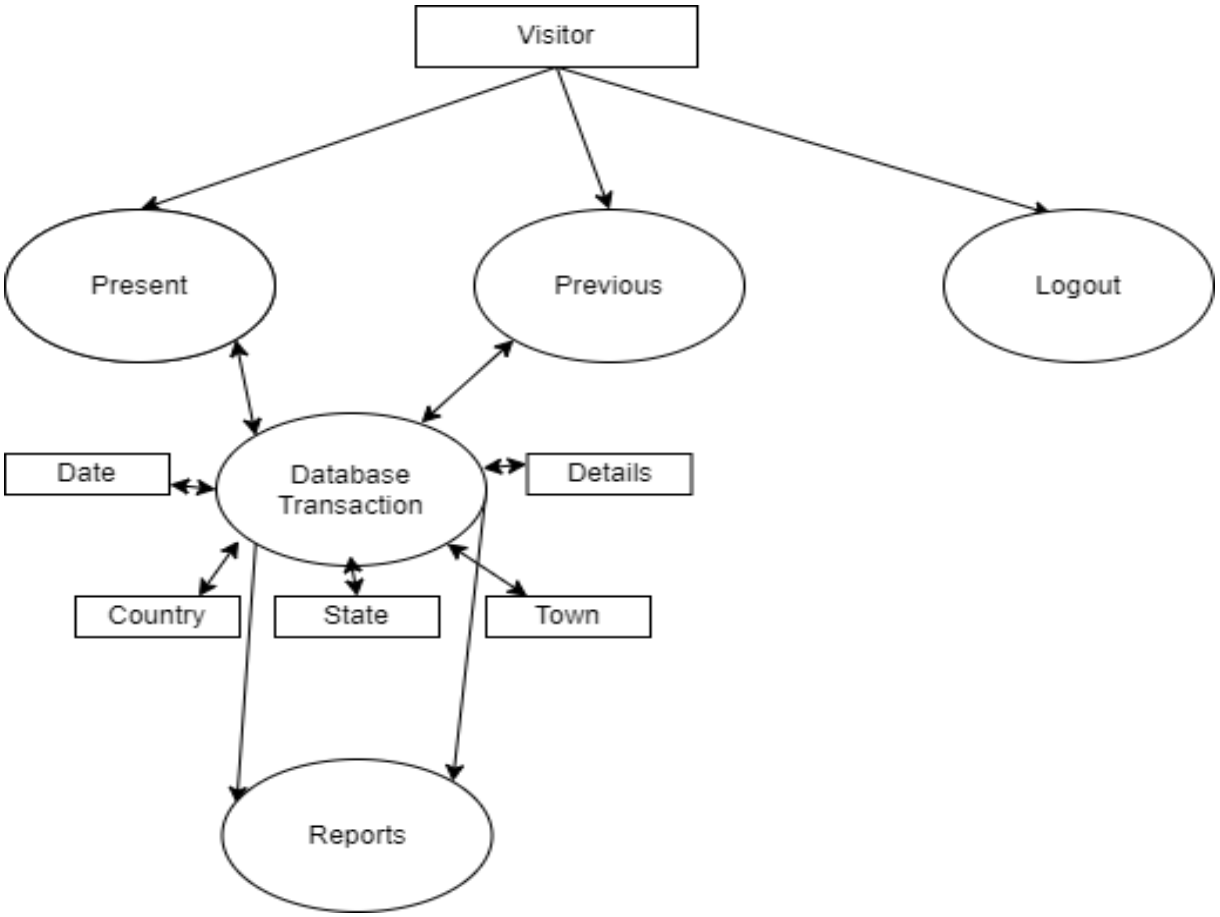

Figure 5.3: Data Flow Diagram 2

5.4 Class Diagram

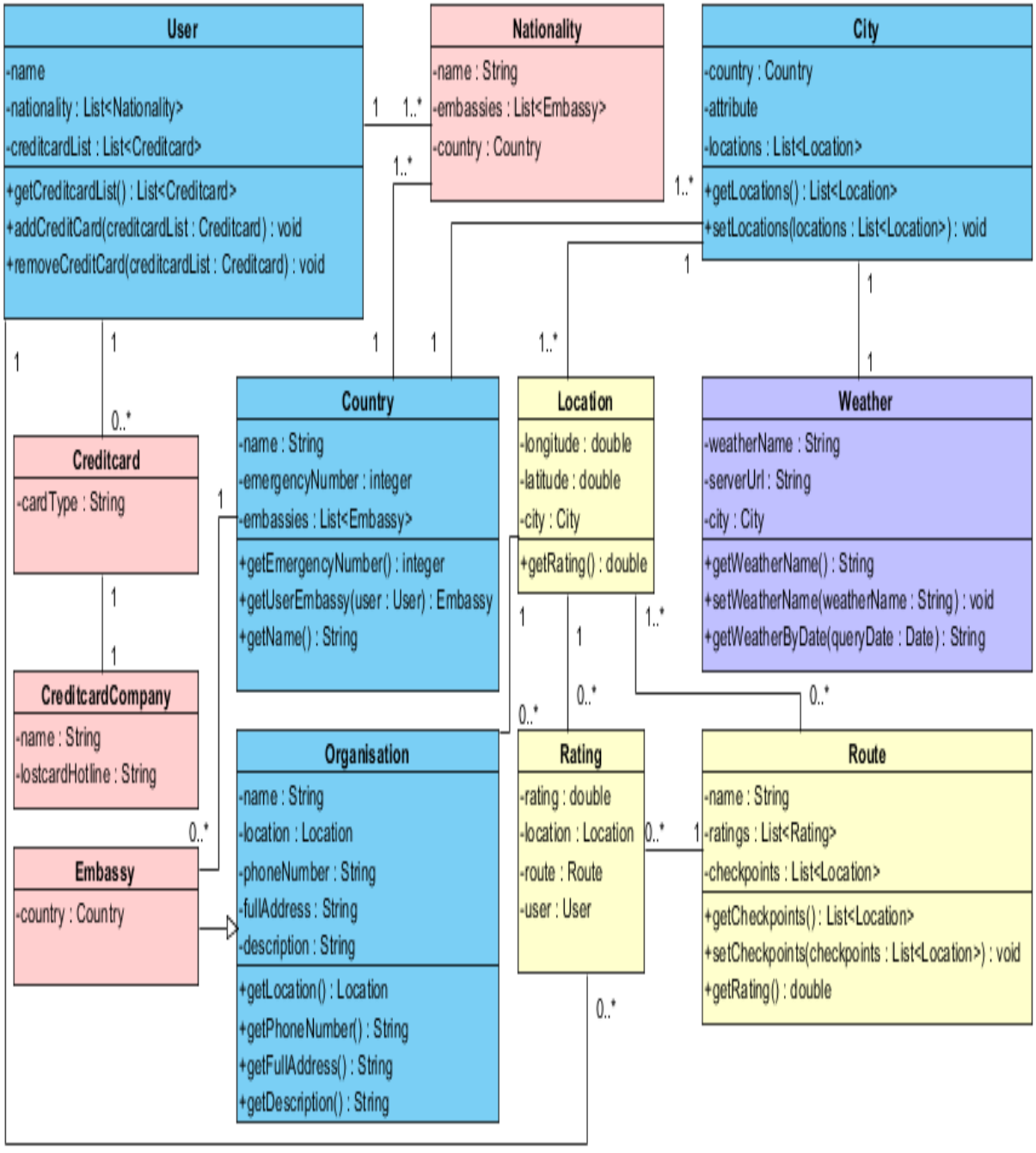

Figure 5.4: Class Diagram

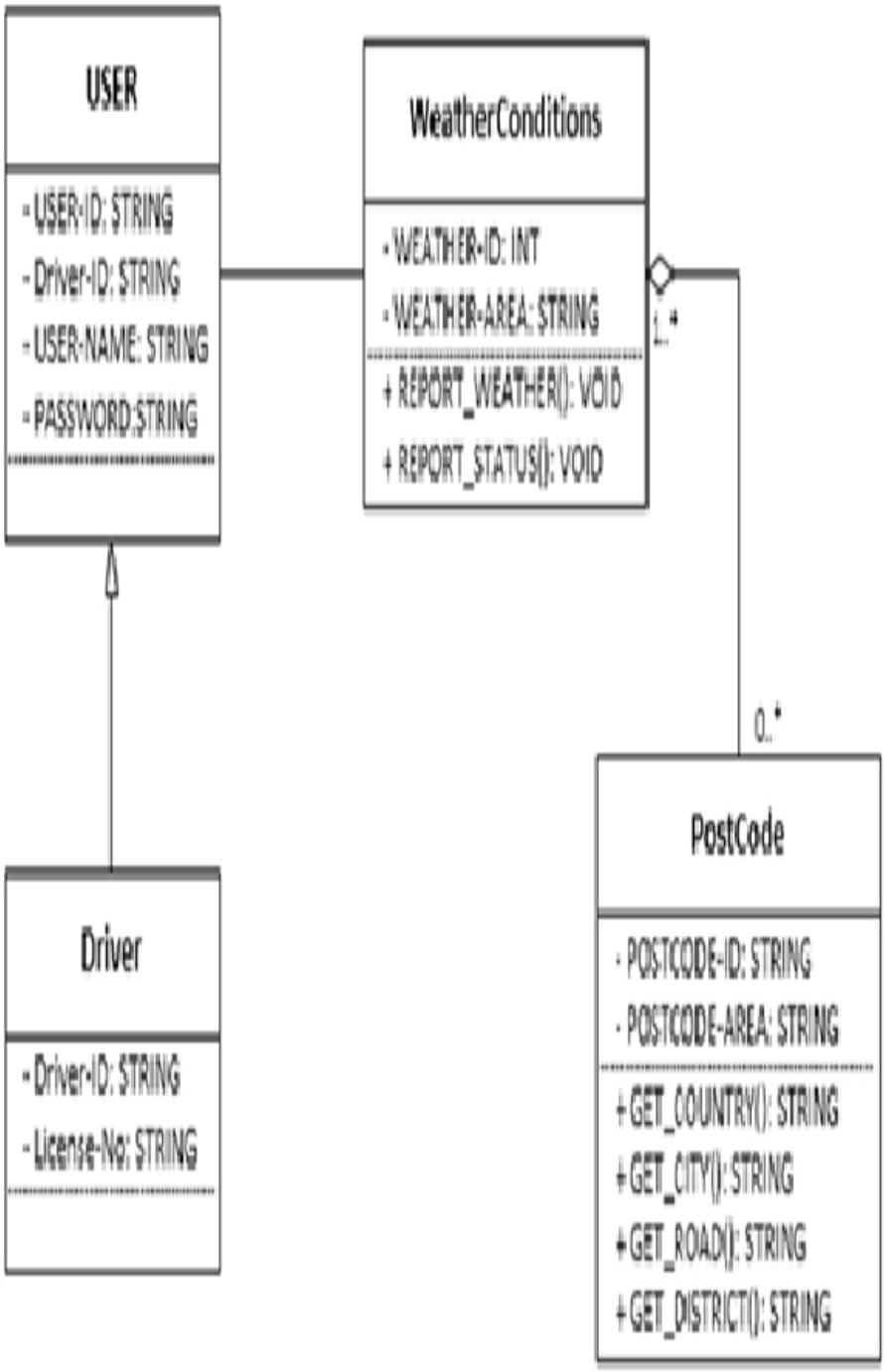

Figure 5.5: Class Diagram

Chapter 6 Requirement Tracebility Matrix

REQUIREMENTS TRACEBILITY MATRIX					
Project Name: Weather Forecasting System					
Business Requirement document BRD		Function Requirements Document FSD			Test case Document
Business Requirement id#	Business Requirement/Business use case	Function Requirement id#	Function Requirement/use case	priority	Test case ide
BR_1	Weather Report	Fr_1	Enter Correct City	High	TC#001 TC#002
		Fr_2	Error on wrong input	High	TC#003 TC#004
		Fr_3	Display Weather report of given city	medium	TC#005 TC#006

Figure 6.1: Requirement Tracebility Matrix

Chapter 7 Manual Test Cases

TEST CASE ID	TEST SCENARIO	TSET CASE	PRE-CONDITION	TEST STEPS	TEST DATA	EXPECTED RESULT	POST CONDITION	ACTUAL RESULT	STATUS
TC_WEATHER_001	Verify insertion of city name	Enter city name e.g., Pune	Need a valid name of city	1.Enter valid city name 2.Submit	<Valid city name> <submit>	Weather report will be shown	Weather card added to screen	Weather report shown	Ok
TC_WEATHER_001	Verify insertion of city name	Enter wrong city name e.g., Punes	Need a valid name of city	1.Enter valid city name 2.Submit	<Invalid city name> <submit>	Invalid city name	Give proper details	Give proper details	Ok
TC_WEATHER_001	Verify insertion of city name	Leave blank field	Need a valid name of city	1.Enter valid city name 2.Submit	<Null> <submit>	Invalid city name	Give proper details	Give proper details	Ok

Figure 7.1: Test Case

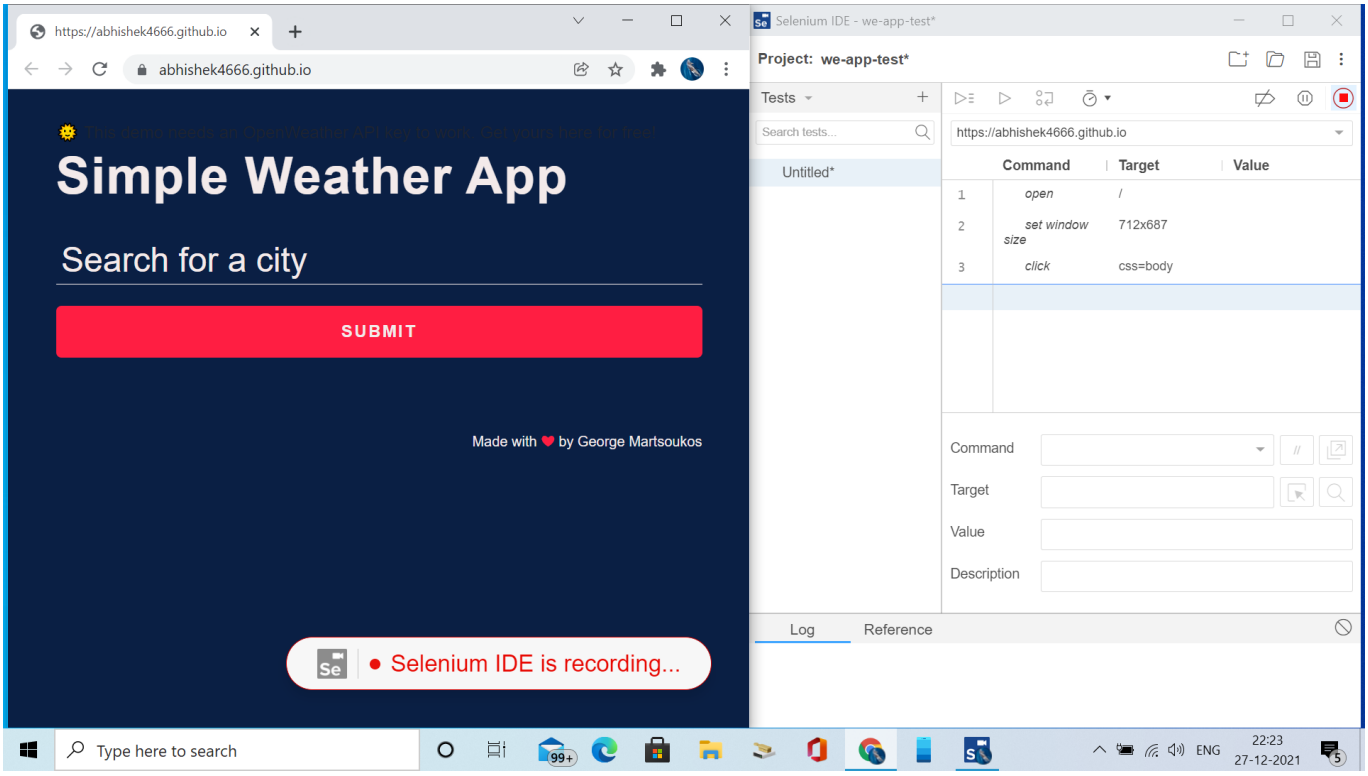

Figure 7.2: Test 1

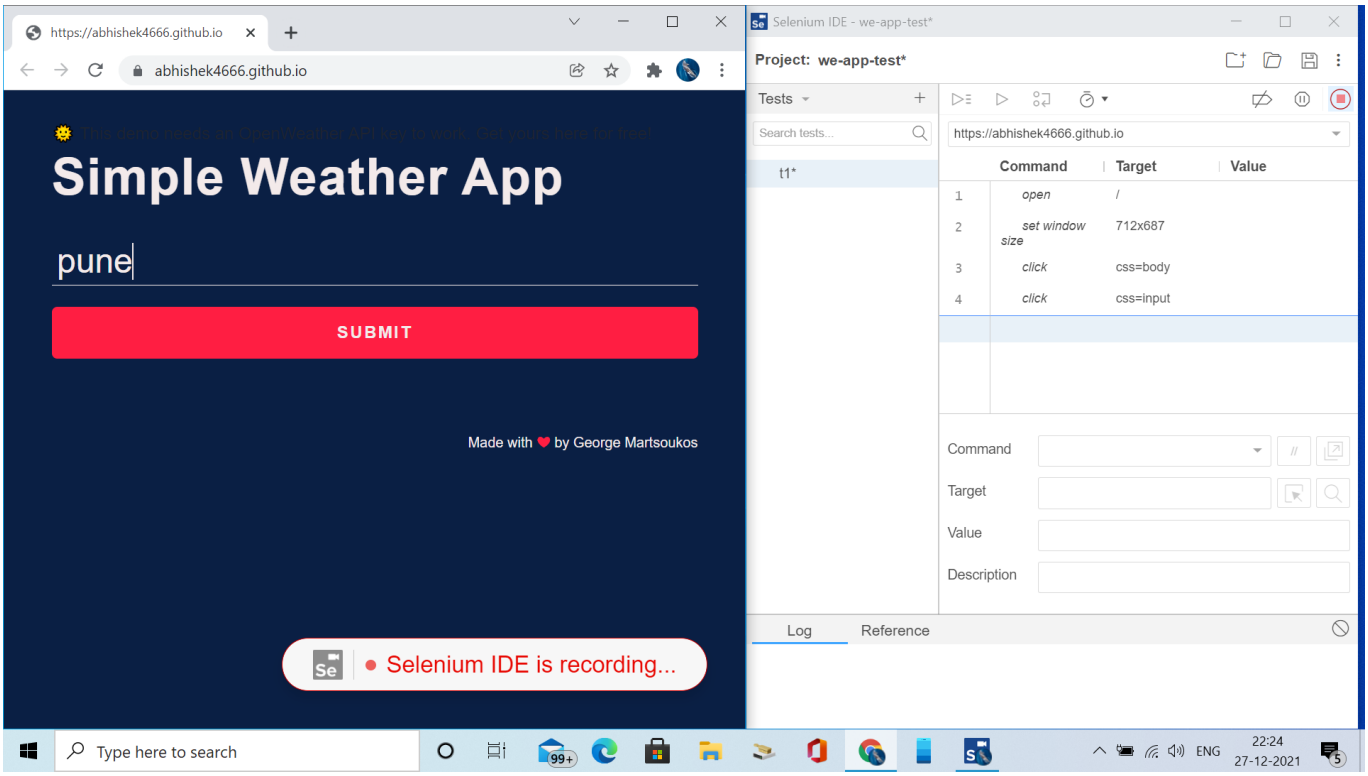

Figure 7.3: Test 2

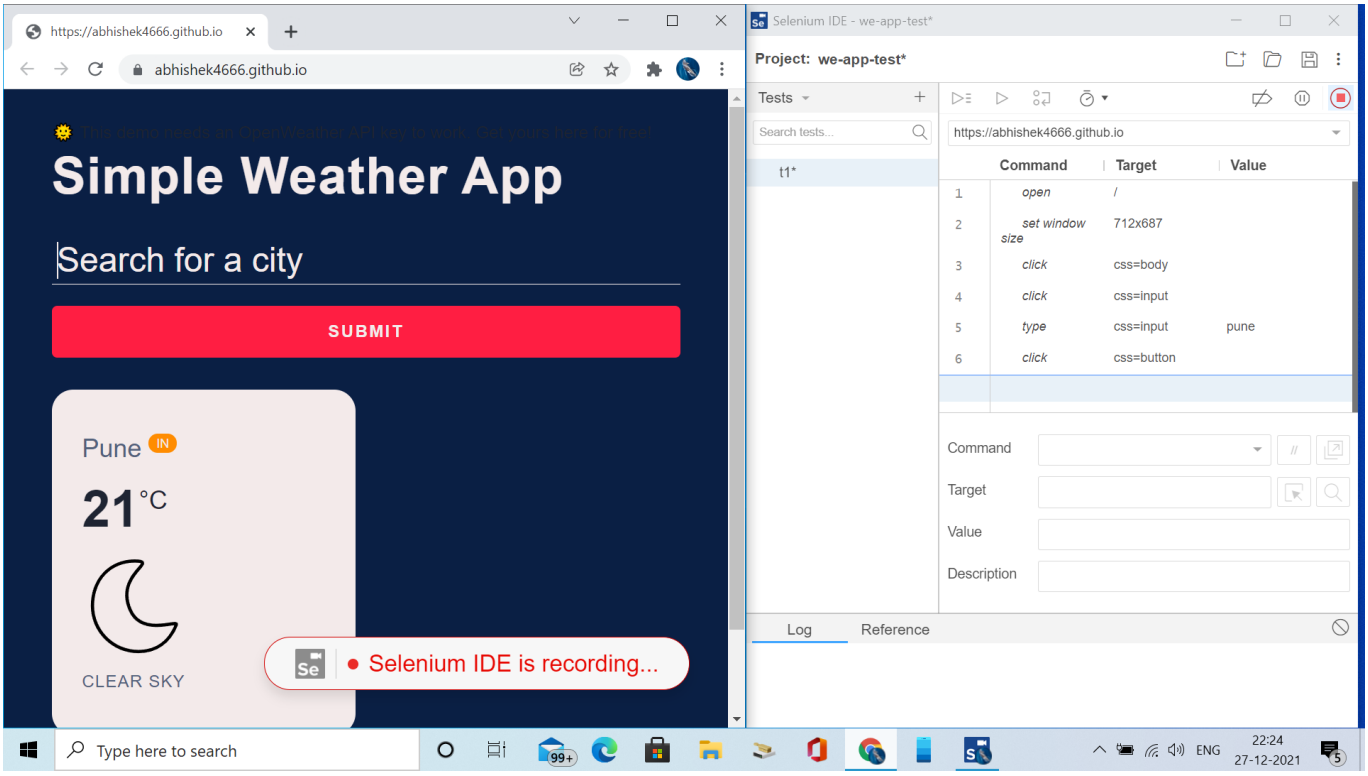

Figure 7.4: Test 3

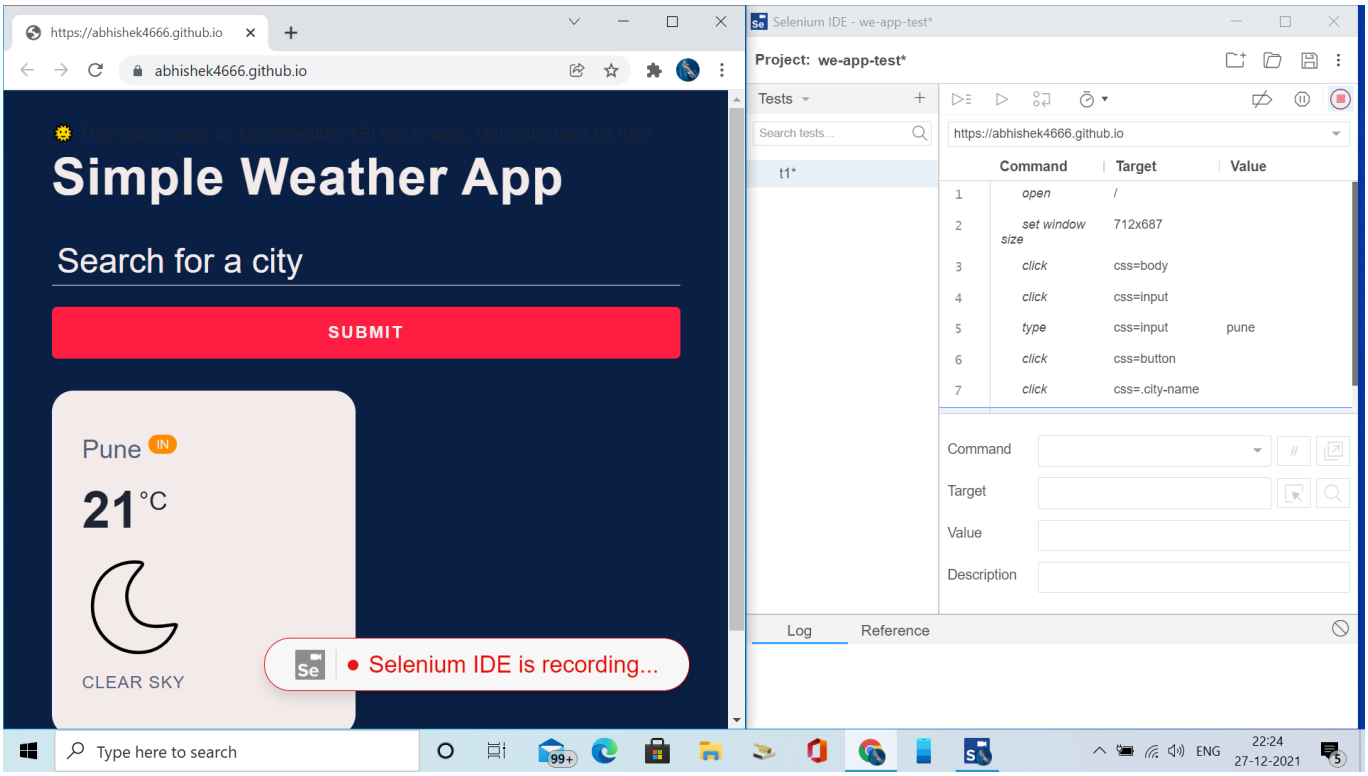

Figure 7.5: Test 4

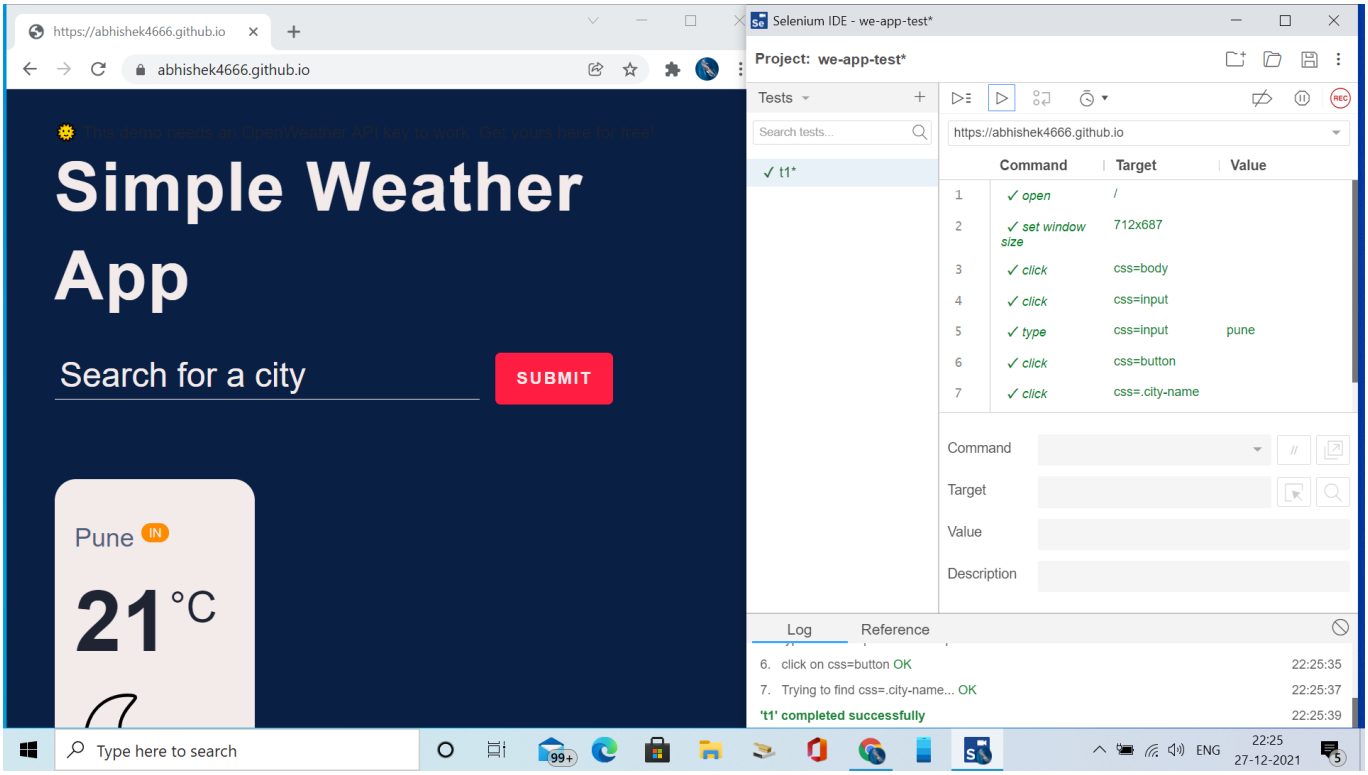

Figure 7.6: Test 5

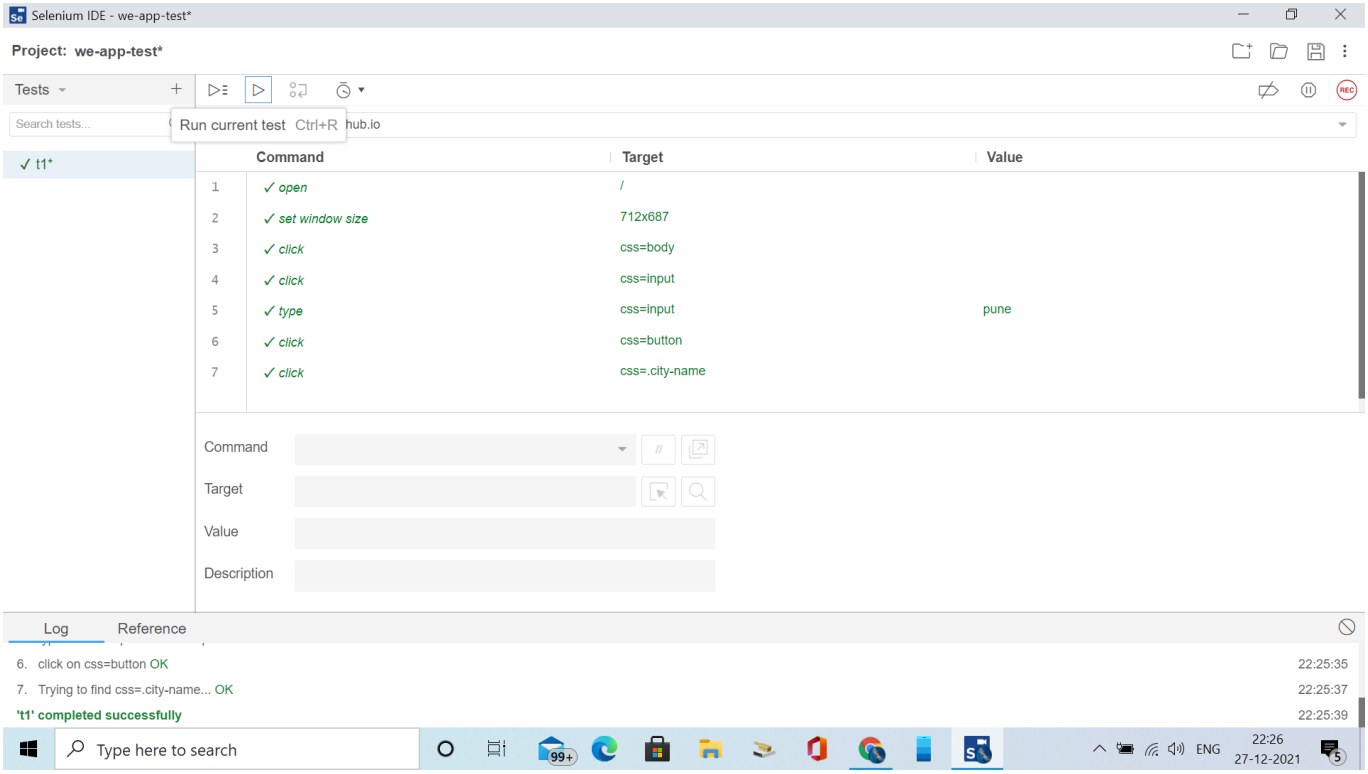

Figure 7.7: Test 6

Chapter 8 Conclusion and Reference

8.1 Conclusion

As part of mini project we built Weather Forecasting System using JavaScript . System contains weather reporting module . We used Selenium IDE to test each module of system.

8.2 Reference

1. www.academia.edu
2. scholar.google.com
3. Book:" Software Engineering- A Methodical Approach" by Elvis C. Foster.