

# WeatherApp [Weather Prediction Webapp]

*Final Project Report*

**Team Members :**

*Kulvir Singh (19BCE2074)*

*Gurtavrein Singh (19BCE2101)*

*C S Sahil (19BCE2094)*

**Course Code : CSE3001**

**Course Title : Software Engineering**

Under the guidance of

**Prof. Manjula R**

**VIT, Vellore**



# VIT<sup>®</sup>

## Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

**SCHOOL OF COMPUTER SCIENCE AND ENGINEERING**

**JUNE, 2021**

# Chapter 1 :

## **Abstract :**

## **Aim :**

To prepare a cost-efficient, compact and an accurate model, which will help the users get a fair idea, about the upcoming weather forecast, using the weather prediction system.

## **Objective :**

1. The primary objective of this project, is to build a cost efficient, and a reliable system, which will help users, get a fair idea about how the future weather is going to be.
2. The system is made to be user friendly, so that users from all sections of society will be able to easily access information about the future's weather condition.

## **Project Scope :**

“Weather” is an unreliable natural entity. Controlling it, is something only mankind could dream of. But predicting it, could help humans plan their next move in their day to day lives. Seeing the importance of weather prediction, and how people all over the world, all classes of society will be using this tool, our team is introducing the “Weather Prediction System”, a cost-efficient, accurate and a user-friendly model, which will help citizens all around the world, to get a fair prediction of the upcoming weather, after they interact with their interface. The primary objective of this model, is to make weather prediction user-friendly and accurate.

# Chapter 2 :

## Software Requirement Specification

### **1 Introduction**

#### **1.1 Purpose**

The purpose of this document is to describe the software specification requirements for Weather Predicting System. The document will describe how the product will collect and display local weather data and analyze the weather forecast. The document contains the functional behavior and non-functional requirements of the system project. The document also contains the guidelines for system engineers and programmers to start working and accomplish the project on a given time frame. The product will be beneficial to the people as they will know what weather lies ahead of them.

#### **1.2 Document Conventions**

The format of this Software Requirement Specifications for Weather Predicting System is simple. Bold face and indentation is used on general topics and on specific points of interest. The rest of the document will be written using the standard font, Calibri

#### **1.3 Intended Audience and Reading Suggestions**

The intended readers of this document are the developers of the station, testers, station owners, managers and coordinators.

In case of any suggested changes on the requirements listed on this document should be included in the last version of it so it can be a reference to developing and validating teams.

#### **1.4 Product Scope**

The software product is of Weather Predicting System . It will be able to display local weather data and analyze weather forecast based on the local weather data fed to the software on autonomous instruments. The product will be beneficial to the common people different organizations who will need weather information from weather station for various reason. The main goal is to have an autonomous record instruments and a database for the safe storage of all the records from weather station.

This product does not include a live stream of the weather. It does not store the data on its on. The data is fed by the personnel handling the software and the prediction of the weather ahead is carried out by the project.

The weather details used are not from India, but are from Switzerland. This is because in order to get the Indian Weather Data one needs to pay certain amount of money. Whereas the data from Switzerland is free.

## **2.Overall Description**

### **2.1 Product Perspective**

The weather prediction system, basically belongs to the class of devices and applications, which determine the weather of a particular location. The only difference is that, our product has a far more superior user interface platform, and also can be easily accessed by people, belonging to all sections of the society. Simple components have been used to build this project.

### **2.2 Product Functions**

- a) The product is built to provide an accurate report of the current day, the previous day, or the next day.
- b) The product has been designed, in such a way that the user interface, (i.e) the website which acts as the GUI, is easily accessible by users from all sections of the society.
- c) The GUI for the project is a website, which has multiple functions like login, select day, etc. which will be further explained in the next module of this document.

### **2.3 User Classes and Characteristics**

The primary classes of users, who use this product, are:-

- a) Naïve users
- b) Application users (with little technical experience)
- c) Product designers

Naïve users:- These users are the ones, who are not involved with the technicality of the product, but they just use the features of the product, like, viewing the weather report of the current day, the previous day or the next day of their respective city/village.

Application users:- These users are the ones, who access the product to fix minor technicalities/bugs etc. which can possibly arise in this product.

Product designers:- Product designers are the ones, who decide to change the existing model into a

better version with respect to their vision over the product, or make required changes/modifications over the product and provide an update over the GUI, for better usability and increased quality of the product.

## **2.4 Operating Environment**

This project has both hardware and software components, integrated to make the best out of the final product. The hardware components are monitored by the product designers, which are capable of working under any conditions, since they are required to perfectly detect the conditions of the weather.

The software is basically a website, which acts as a GUI for the project, as the website is the place where the user will interact and obtain the details that the product is supposed to be served for. The Software GUI component comprises of a front-end and a back-end, which deals with the user interface and data handling, producing results, etc. respectively.

## **2.5 Design and Implementation Constraints**

### Design constraints:-

- a) With respect to the basic model/prototype of the product that is being presented to you, the hardware components should be present in each and every city across a particular region, where the product's operation is valid, for the detection of weather.
- b) The GUI being built has to be made as simple as possible, since the main principle of this product is that the product should be user friendly, and easily accessible by users from all sections of the society.
- c) Language requirement is another constraint, because not everyone who wishes to have a fair idea about the weather of the near future, will know English

### Implementation constraints:-

Users are required to have basic internet facilities until new updates/modifications arise, which won't need this facility.

The hardware components, which detect the weather, must be present in an accurate location of the city, which can give a fair idea about the entire city's weather condition.

## **2.6 User Documentation**

- a) Terms and regulations norms will be given to the user, before they can access the website.
- b) A user manual will be provided as a soft copy in the website.

## **2.7 Assumptions and Dependencies**

Multiple assumptions have been made, which can further enhance the functioning of the product. The use of satellites, which can easily collect statistics and data, which can be useful to display the weather report more accurately, and avoid the hassle with the hardware components.

The product has a primary dependency, (i.e) it depends on the predictability of weather too, so that the hardware components can be fed with the required instructions, which can help in predicting a particular type of weather.

# **3. External Interface Requirements**

## **3.1 User Interface Requirements**

User Interface: Graphical (GUI) or Command-Line (CLI) This system will provide a graphical user interface. The layout of the system will be a webpage that has a header on the top displaying the name of the website, under the header there will be a navigation bar to go to related pages, under the navigation bar is the body or main focus and at the bottom of the page will be a footer that will have contact information

## **3.2 Hardware Interfaces**

No hardware interfaces required

## **3.3 Communications Interfaces**

No communication interface is required

### **3.4 Software Interfaces**

The software interfaces that will be used are Visual Studio/sublime, MyphpAdmin . Visual Studio/sublime will be used for coding the website. MyphpAdmin will be used to maintain the database to store the forecast. We will be using the following programming languages: HTML,Python,CSS. The following libraries are used:

- Pyflask: . Pyflask will be used for implementing request,response and other utility functions which enables the web framework to be built on top of it.
- Scikit: Contains basic mathematical tools and used for building machine learning model.
- Pandas: Shall be used for data analysis
- Numpy: Provides multidimensional arrays and tools to compute these arrays

## **4. System Features**

\*\*The ratings are given from 0 to 9, where 0 being no priority and 9 being a very high priority

### **4.1 LOGIN:-**

#### **4.1.1 Description and Priority**

In this feature the user can type in his credentials to log-in and get his proper customized

settingsRating:-8

#### **4.1.2 Stimulus/Response Sequences**

The user fills in his

- Username
- Password

- If doesn't have an account can create an account
- Has to accept to access the device

location. The system

- recognizes the given username and password are correct or not.
- The system also displays an error message if the username and the password are incorrect.
- The system opens the sign-up page if required.

### **4.1.3 Function Requirements**

The main function requirements required for the working of Log-In page is

- Must have a strong backend database.
- Ability to interact with the device's GPS to access the device location.
- The webpage must be very interactive for the user to have a easy and effortless experience.

## **4.2 SIGN-UP:-**

### **4.2.1 Description and Priority**

In this feature the user can type in his credentials like email-id, username and password to sign-in and create an account

Rating:-8

### **4.2.2 Stimulus/Response Sequences**

The user fills in his

- Email-id
- Username
- Password
- Confirmed

password. The system then,

- Creates the account .

### **4.2.3 Function Requirements**

The main function requirements required for the working of Sign-up page is

- Must should have a strong backend database.
- The webpage must be very interactive for the user to have a easy and effortless experience.



## **4.3 MAIN PAGE:-**

### **4.3.1 Description and Priority**

In this feature the user is greeted with a welcome message, Location, and also the relevant weather information of the present day, the next day and the later day

The weather information includes

- Temperature
- Precipitation
- Humidity
- Win

dRating:-8

### **4.3.2 Stimulus/Response Sequences**

The user clicks

- Today
- Tomorrow
- The next day

The system then displays ,

- Temperature
- Precipitation
- Humidity
- Wind

### **4.3.3 Function Requirements**

The main function requirements required for the working of login page is

- Must should have a strong backend database.
- The webpage must be very interactive for the user to have a easy and effortless experience.
- Ability to interact with the device's GPS to access the device location.
- A proper Machine Learning model.
- Proper connection between the front end and the machine learning model.

## **4.4 SETTINGS:-**

### **4.4.1 Description and Priority**

In this feature the user can choose his preferred settings and can thus customize it using his way. Rating:-4

### **4.4.2 Stimulus/Response Sequences**

The user can change

- Units of speed(km/hr or miles/hr)
- Dark-Mode or Light-Mode
- Units of temperature(F/C)
- Logou

tThe system  
can

- Change according to the inputs give to user.
- Logout from the users' account.

### **4.4.3 Function Requirements**

The main function requirements required for the working of Log-In page is

- Must have a strong backend database.
- Ability to interact with the device's GPS to access the device location.
- The webpage must be very interactive for the user to have a easy and effortless experience.

## **5. Other Non-functional Requirements**

### **5.1 Safety Requirements**

If there is extensive damage to a wide portion of the database due to catastrophic failure, such as a disk crash, the recovery method restores a past copy of the database that was backed up to archival storage (typically tape) and reconstructs a more current state by reapplying or redoing the operations of committed transactions from the backed up log, up to the time of failure.

### **5.2 Security Requirements**

Security systems need database storage just like many other applications. However, the special requirements of the security market mean that vendors must choose their database partner carefully. The data stored in the databases can be encrypted to prevent from unauthorized access.

### **5.3 SOFTWARE QUALITY ATTRIBUTES**

AVAILABILITY: The location must have all the weather related information.

CORRECTNESS: The weather data should be accurate to a fair extent.

MAINTAINABILITY: The administrators must have the option to maintain the web site and provide updates

USABILITY: The website must allow multiple users to access the website at the same time with no difficulty.

## **5.4 Performance Requirements**

### NORMALIZATION

The basic objective of normalization is to reduce redundancy which means that information is to be stored only once. Storing information several times leads to wastage of storage space and increase in the total size of the data stored.

### ACCURATE MACHINE LEARNING MODEL

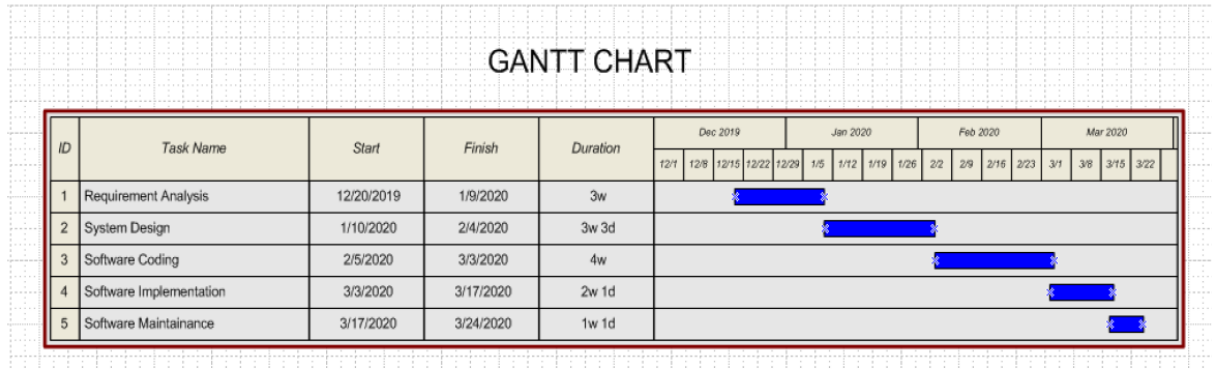
The chosen machine learning model must be accurate in providing the predicted data to a fair extent as none of the model are 100% accurate. The model must be best fitting model.

## **6. Other requirements**

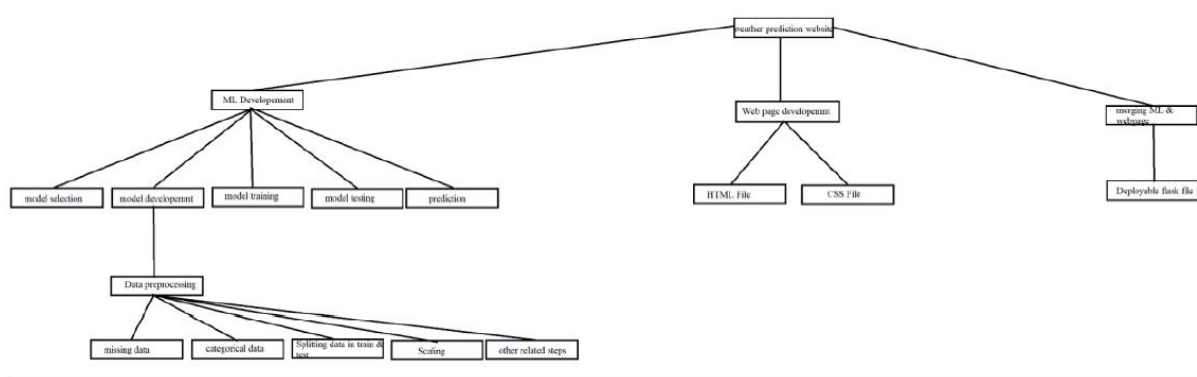
- a) Legal access from the Govt. Of India to setup the hardware in various cities, for predicting the weather.
- b) Database of huge size is mandatory, which must store the login details of the users, who use the product
- c) Reliable host, which can support the functioning of the website, (i.e) the GUI of the project.

# Project Planning – Scheduling Charts

## Gantt Chart



## Work Breakdown Structure



Project: Building a weather predicting website

Activity 1: ML Development

- Task 1.1: model selection
- Activity 1.2: model development
- Task 1.3: model training
- Task 1.4: model testing
- Task 1.5: prediction

Activity 1.2: model development

Activity 1.2.1: Data Preprocessing

Activity 1.2.1.1: Data Preprocessing

- Task 1.2.1.1.1: Missing data
- Task 1.2.1.1.2: Categorical data
- Task 1.2.1.1.3: Splitting data into train and test
- Task 1.2.1.1.4: Scaling
- Task 1.2.1.1.5: Other related steps

Activity 2: Web Page Development

- Task 2.1: HTML Page
- Task 2.2: CSS File

Activity 3: Merging ML and Web page

- Task 3.1: Deployable flask file

## **Process Model - Incremental Model**

For our project of weather prediction system, we have decided to go with the “Incremental model”

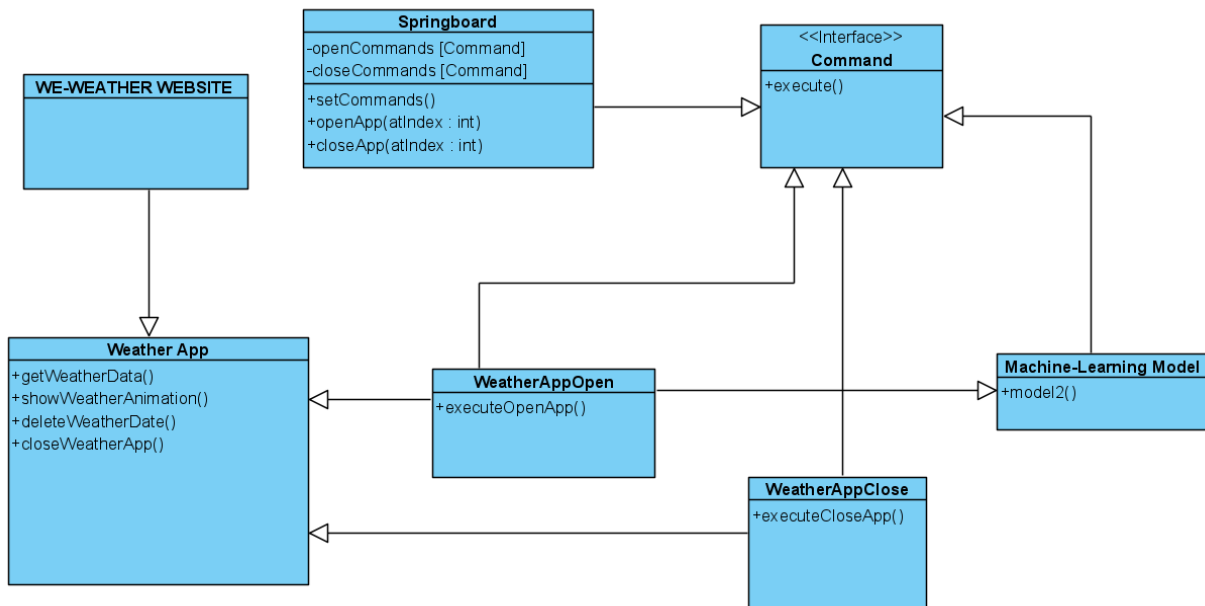
The reasons for choosing “Incremental Model” to other models are as follows;

- Generates working software quickly and early during the software life cycle.
- This model is more flexible – less complicated to change scope and requirements.
- It is easier to test and debug during a smaller iteration.
- In this model client (in this case; faculty) can respond to each built.
- Easier to manage risk because risky pieces are identified and handled during its iteration.
- With only three members in the team the resources and the skillset is less thus making model an apt model.

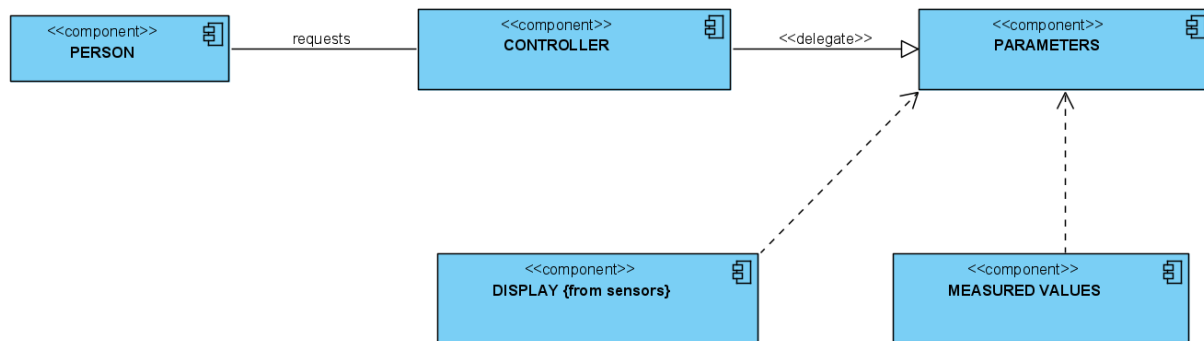
# Chapter 3 :

## System Design – UML Diagrams

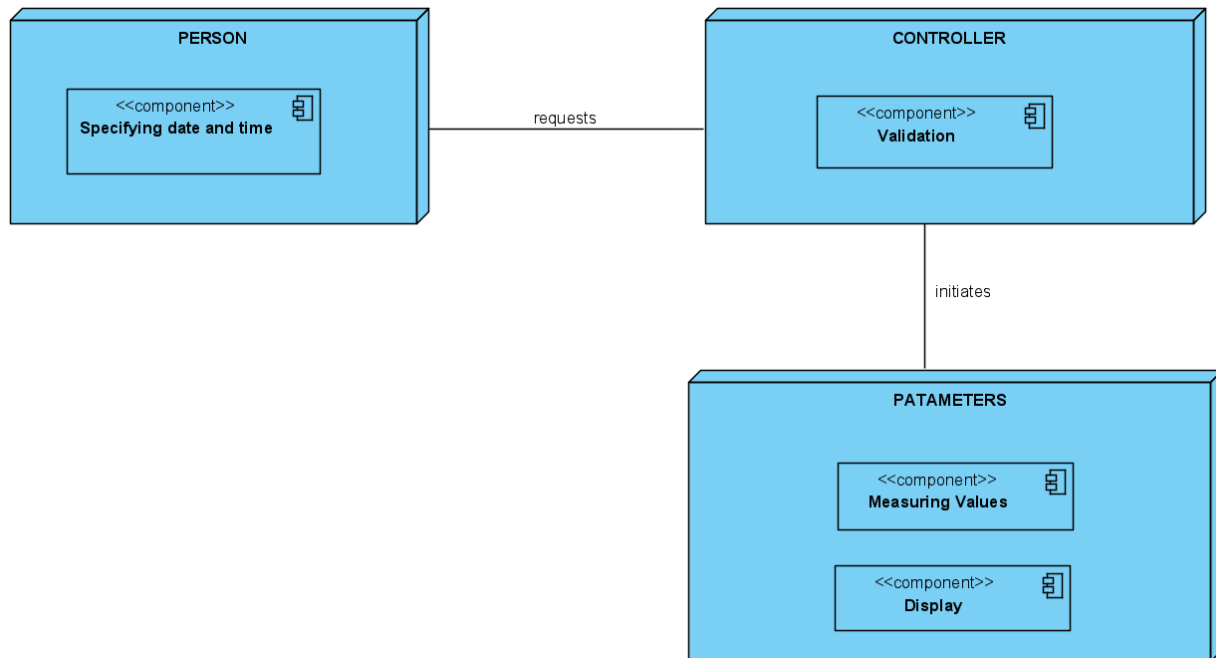
### Class diagram



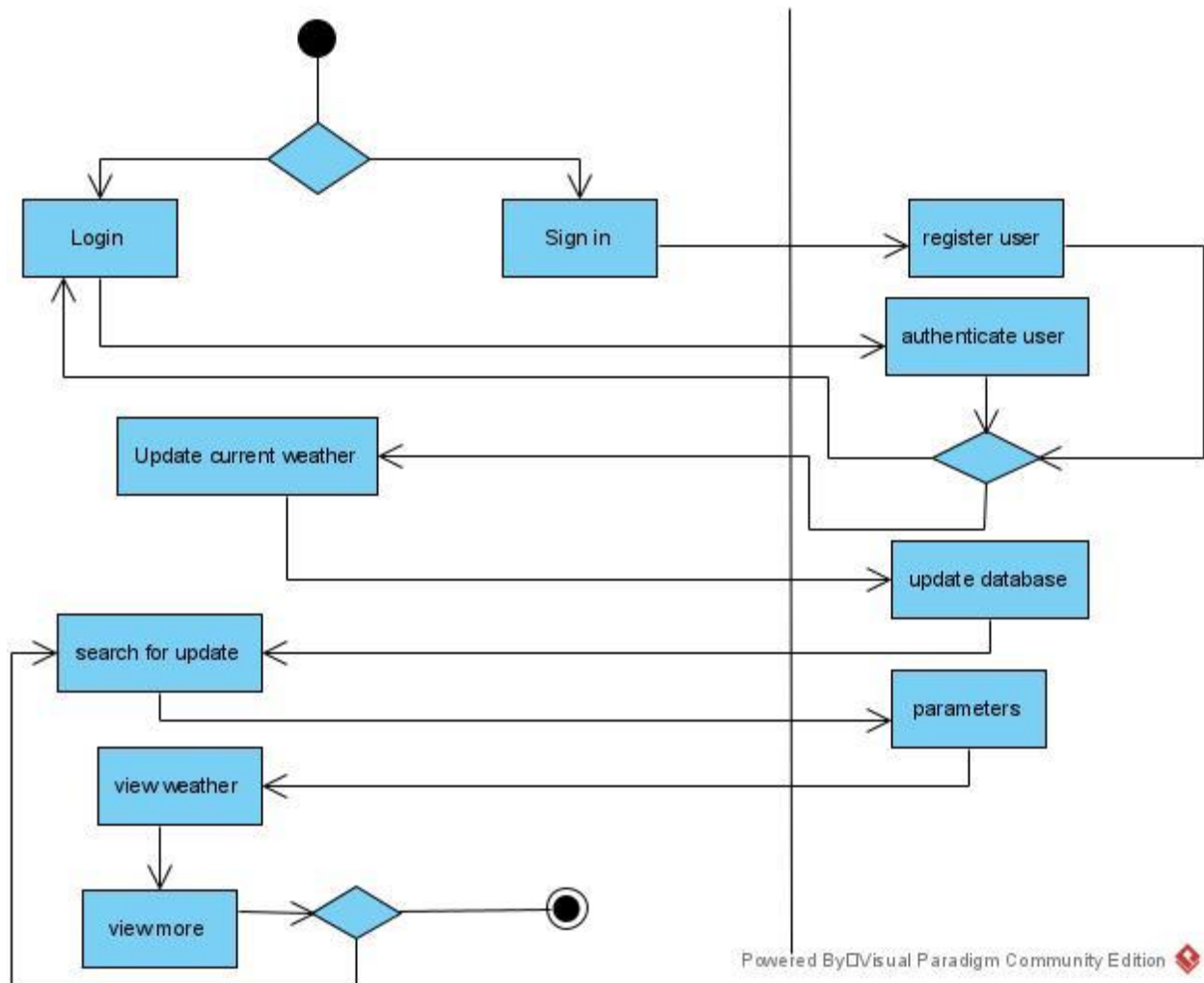
### Component



## Deployment Diagram

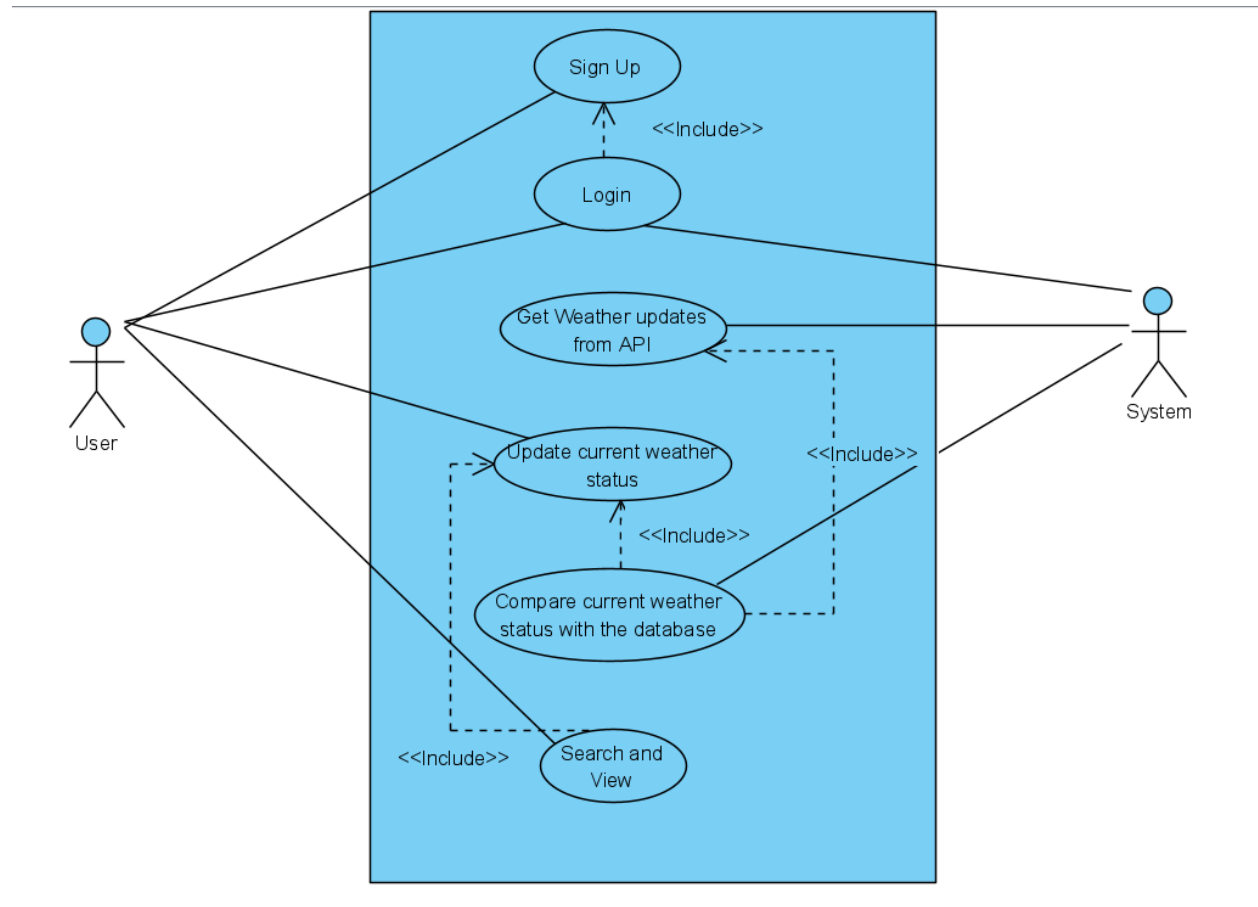


## Activity Diagram

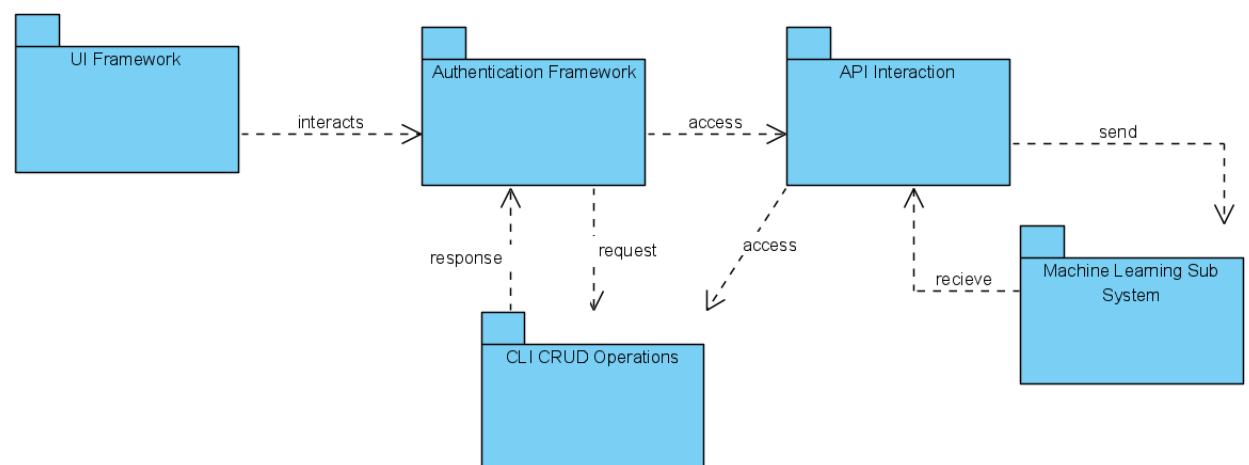




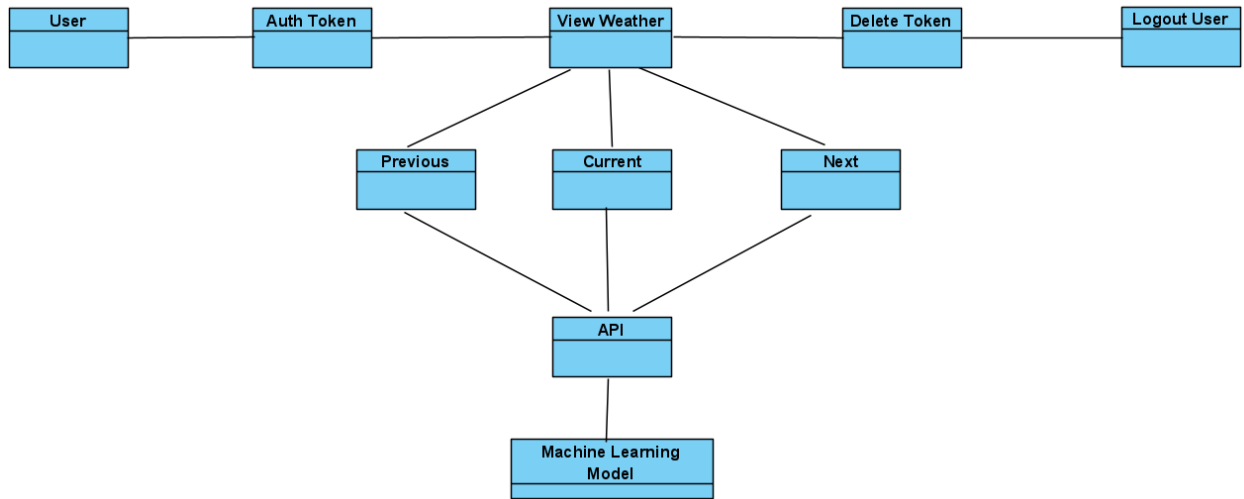
## Use case Diagram



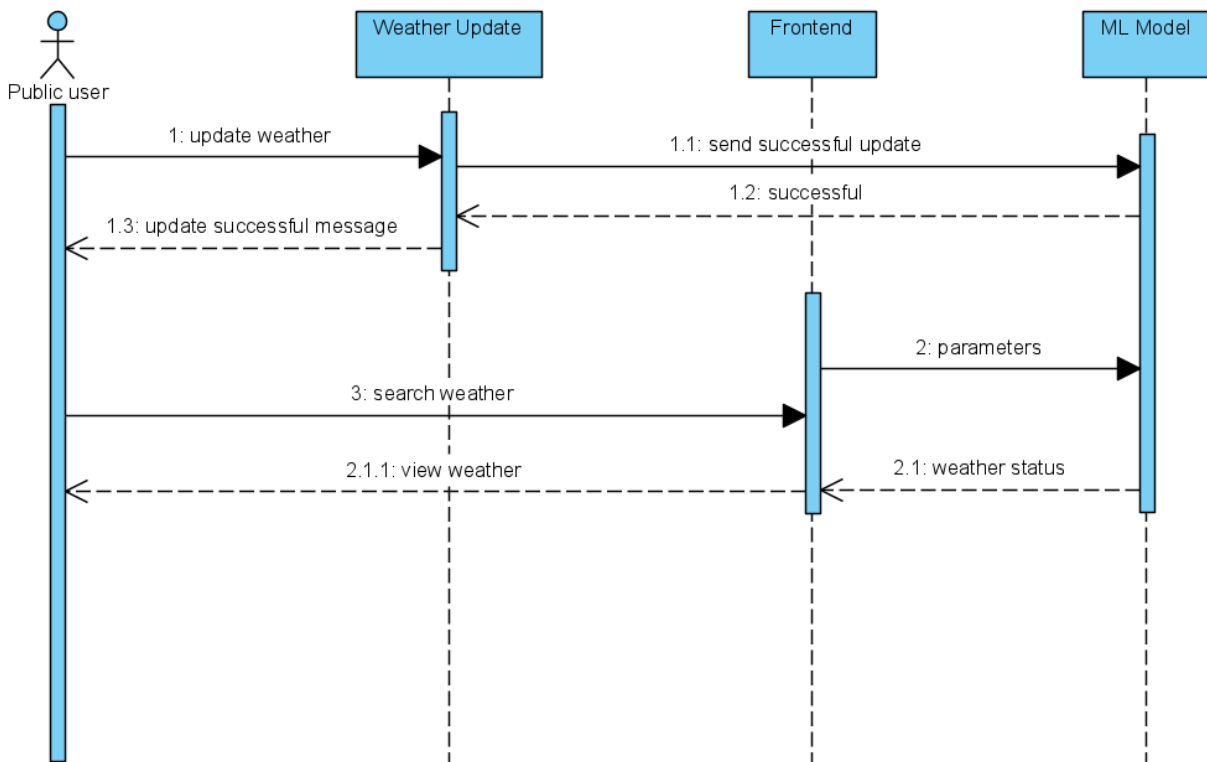
## Package Diagram



## Object Diagram



## Sequence Diagram



# Chapter 4 :

## Software Process Specification –

## Source Code

### Source Code:

The source code is given in the following drive links. Follow the link to view all the code files.

#### Full Project

[https://drive.google.com/drive/folders/1\\_cIZZKKISC3eP7IeIC3Wkkhiovn9wAK0?usp=sharing](https://drive.google.com/drive/folders/1_cIZZKKISC3eP7IeIC3Wkkhiovn9wAK0?usp=sharing)

#### Full Frontend code

public folder =

<https://drive.google.com/drive/folders/1NUftlafBhulsPplJoEj2pDPamHudryav?usp=sharing>

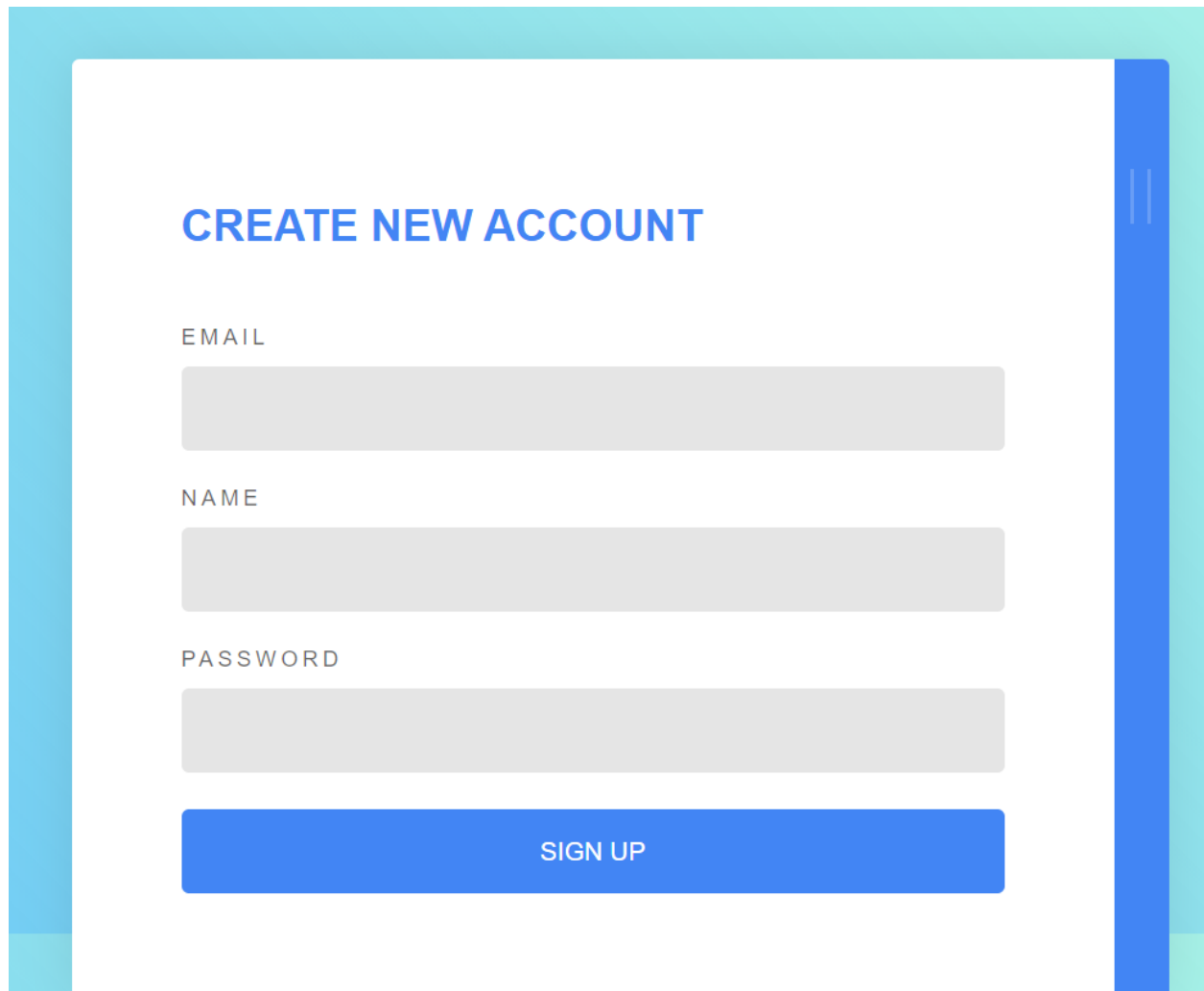
Ejs folder =

<https://drive.google.com/drive/folders/1PnUsk4n0QFRhQ9Rwq2pxXuKL-mP4gPHD?usp=sharing>

#### Full Backend code

- Models –  
[https://drive.google.com/drive/folders/1q9cZ9xaWBisVH3ICc-U0YM\\_8Yy6bVgTZ?usp=sharing](https://drive.google.com/drive/folders/1q9cZ9xaWBisVH3ICc-U0YM_8Yy6bVgTZ?usp=sharing)
- Routes -  
[https://drive.google.com/drive/folders/1q9cZ9xaWBisVH3ICc-U0YM\\_8Yy6bVgTZ?usp=sharing](https://drive.google.com/drive/folders/1q9cZ9xaWBisVH3ICc-U0YM_8Yy6bVgTZ?usp=sharing)
- Services -  
<https://drive.google.com/drive/folders/10U-E2Cphk0IWGXk5mOjyPpCeQr8OnJp5?usp=sharing>
- Utils –  
<https://drive.google.com/drive/folders/10U-E2Cphk0IWGXk5mOjyPpCeQr8OnJp5?usp=sharing>

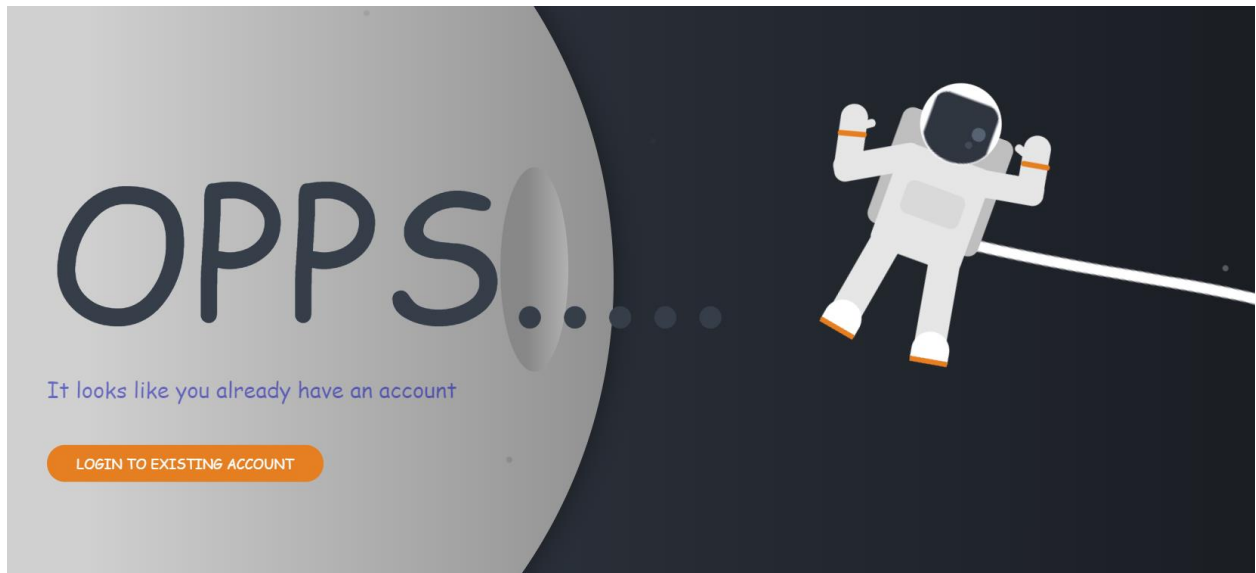
## UID Screenshots:



A screenshot of a user interface for creating a new account. The page is framed by a thick blue border. On the right side of the frame, there is a vertical blue bar with two white vertical lines near the top. The main content area is white and contains the following elements:

- CREATE NEW ACCOUNT**: A blue heading at the top.
- EMAIL**: A label above a light gray rectangular input field.
- NAME**: A label above a light gray rectangular input field.
- PASSWORD**: A label above a light gray rectangular input field.
- SIGN UP**: A blue rectangular button with white text, positioned below the input fields.

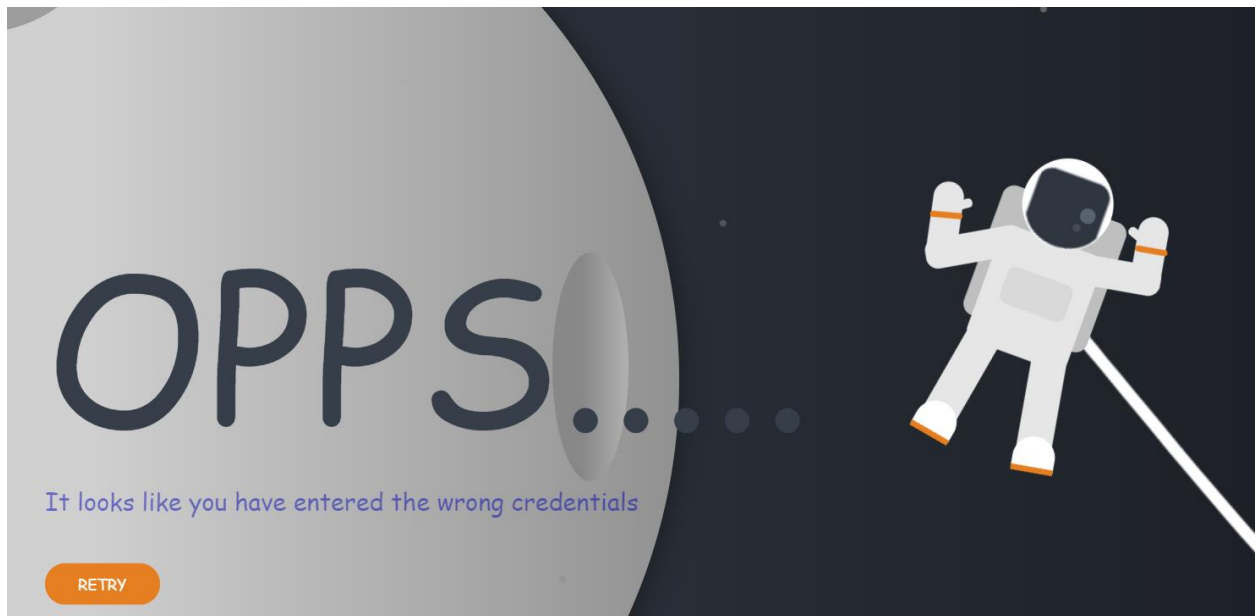
New account creation landing page



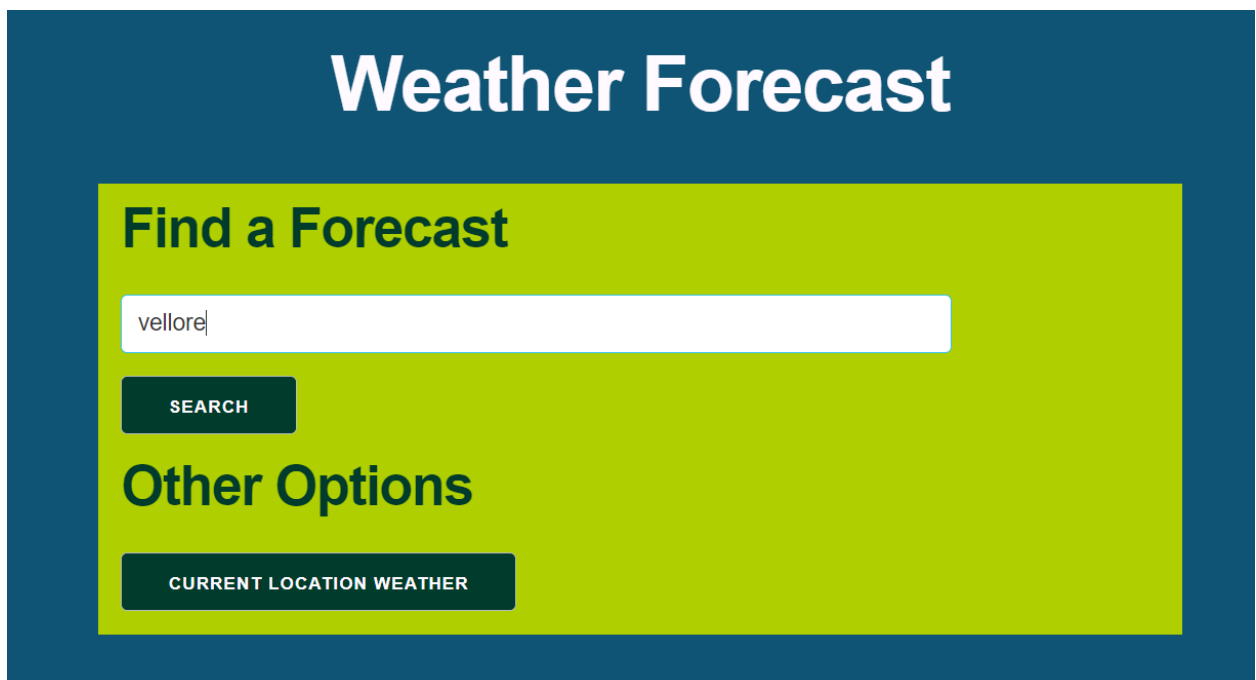
Error message page

An account login landing page with a light blue border. The page has a white background. At the top, the text "ACCOUNT LOGIN" is written in a bold, blue font. Below this, there are two input fields: one for "EMAIL" and one for "PASSWORD". Both fields are represented by light grey rectangular boxes. Below the password field is a blue button with the text "LOG IN" in white. On the right side of the page, there is a vertical blue bar with a white double-line icon near the top.

Account login landing page



Error message page



User account home page

```
// 20210429153841
// http://localhost:3000/result

{
  "coord": {
    "lon": 79.1333,
    "lat": 12.9333
  },
  "weather": [
    {
      "id": 800,
      "main": "Clear",
      "description": "clear sky",
      "icon": "01d"
    }
  ],
  "base": "stations",
  "main": {
    "temp": 38.14,
    "feels_like": 38.12,
    "temp_min": 38.14,
    "temp_max": 38.14,
    "pressure": 1006,
    "humidity": 25,
```

JSON formatted result of predicted weather

```
// 20210429153915
// http://localhost:3000/result/ip_location_weather

{
  "coord": {
    "lon": 86.1833,
    "lat": 22.8
  },
  "weather": [
    {
      "id": 721,
      "main": "Haze",
      "description": "haze",
      "icon": "50d"
    }
  ],
  "base": "stations",
  "main": {
    "temp": 41,
    "feels_like": 40.83,
    "temp_min": 41,
    "temp_max": 41,
    "pressure": 1005,
    "humidity": 20
  }
}
```

Current location weather prediction via GPS



```
mysql> use weatherapp;
Database changed
mysql> select * from user;
```

name	email	password	createdAt	updatedAt
abc	abc	\$2b\$10\$.Jyadnyjq.Q8zQI83h8p7e8Qe5ILb.09FFAK/ctvtdglvDyFsygIG	2021-04-26 21:04:37	2021-04-26 21:04:37
asd	asd	\$2b\$10\$7VZdK/06zXTb5s0ETm6FF0iVFelRrLqKzMDLBi..uTCVijDcCcuKM2	2021-04-26 20:40:38	2021-04-26 20:40:38
zxc	zxc	\$2b\$10\$FLTTskWfNnwuipu8EHAP0ecmgXEAaNMVLS.4J9H3HinlLoG0KNpZy	2021-04-28 17:07:53	2021-04-28 17:07:53

```
3 rows in set (0.01 sec)

mysql>
```

User database with encryption of password for security requirements

# Chapter 5 :

## Software Test Plan – Test Cases

Test Case ID	Test Scenario	Test Steps	Test Data	Expected Result	Actual Result	Pass/Fail
T01	Create new user account	Go to homepage. Click on sign up button, enter details and click sign up button	Email = <a href="mailto:test@123.com">test@123.com</a> Name = Test Password = password	Main landing page opens and user can access features of the weather app	Landing page opened and user can access features of the weather app	Pass
T02	Login existing account	Go to homepage. Click on login button, enter details and click login button	Email = <a href="mailto:test@123.com">test@123.com</a> Password = password	Main landing page opens and user can access features of the weather app	Landing page opened and user can access features of the weather app	Pass
T03	Faulty data in login/create new user	Go to homepage. Click on login button, enter details and click login button	Email = <a href="mailto:test@123.com">test@123.com</a> Password = notPassword	Error page opens and displays error message	Error page opened and error message displayed	Pass
T04	Download data set	After logging into web app, click the download data set button.	nil	An excel file should start downloading after 2secs	Excel file is successfully downloaded	Pass
T05	Search for current location data	After logging into web app, click current location data button and then click view data button	nil	A page displaying the correct current location and its weather description and data.	The correct current location and its weather description and data is displayed	Pass
T06	Download data for current location	After logging into web app, click current location data button and then click download data button	nil	A text file starts downloading which has the correct current location and its weather description and data.	A text file is downloaded which has the correct current location and its weather description and data.	Pass
T07	Search for a given location and get results	After logging into the web app, enter a location and click search button and then view data	Location = Jamshedpur	A page displaying the correct search location and its weather description and data.	A page displays the correct search location and its weather description and data.	Pass

T08	Search for a given location and download results	After logging into the web app, enter a location and click search button and then download	Location = Jamshedpur	A text file starts downloading which has the correct search location and its weather description and data.	A text file is downloaded which has the correct search location and its weather description and data.	Pass
T09	Search for given location and predict weather data	After logging into the web app, enter a location and click predict data button	Location = Jamshedpur	A page displaying the search location and its predicted weather description and data.	A page displays the search location and its predicted weather description and data.	Pass

## Chapter 6 :

### Result Analysis and Discussion

Successfully developed a Weather -Forecasting web-app that can predict the weather of given specified city

A lot of applications, which predicts the weather, do exist, including web-apps. But as mentioned earlier, our primary aim is to help our users be able to use this app with much ease, making the output easily accessible to them at much less effort.

The primary novel feature of this project, is to that it is built in a cost efficient manner, and a reliable system, also helps users, get a fair idea about how the future weather is going to be.

The system is made to be user friendly, so that users from all sections of society will be able to easily access information about the future's weather condition.

Another novel feature about our project is that users can have a personal account in our web-app, which saves the dashboard containing the cities that the user had selected, to easily access the said city's predicted data.

The web-app developed by us also has the wonderful feature to store the history of search of cities. It remembers the user's search history, if ever the user wishes to predict the same city's weather at a diff. point in time.

## Chapter 7 :

### Conclusion

The development of a web-app, which could serve a greater purpose for users belonging to various sections of the society has indeed been a mind-boggling process for us to develop.

With the right guidance and application of the right tools in the right way, we have developed “WeatherApp, a Weather Predictor web-app” which could help people from different classes of society, to be able to predict data from trivial set of inputs, with negligible requirements expected otherwise.

With the novel feature, to be able to save the search history of the user, and help the users have a private access to the features of our web-application using the login system, our software has indeed realized its purpose and goals.

We, however would like to make great improvements on our web-application, to develop it into an application, which can be compatible into any software, which could be accessed from the most remote places and also provide a much better user-interface for the same.

## Chapter 8 :

### Future Scope

This project can be further worked on and developed to make a better application which can be used in real time. For starters, we can a mobile application implemented on the same concept which is used in the web application. The machine learning modules can be implemented using various other techniques such as deep learning. The User interface can be made better and more interactive use ReactJS. Apart from these features a good web hosting platform can be made. This can allow other websites and other developers to fetch data and display the same on their product. We can integrate this with the Operating system of any device be it a laptop or a mobile phone.

## *References :*

PROVIDING PASSWORD SECURITY BY SALTED PASSWORD HASHING USING BCrypt ALGORITHM

by P. Sriramya and R. A. Karthika

<https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.1072.20&rep=rep1&type=pdf>

An interactive predictive system for weather forecasting IEEE

<https://ieeexplore.ieee.org/document/6220375>

Weather Forecast Prediction: An Integrated Approach for Analyzing and Measuring Weather Data by Munmun Biswas, Tanni Dhoom, Sayantanu Barua

[https://www.researchgate.net/publication/329922758\\_Weather\\_Forecast\\_Prediction\\_An\\_Integrated\\_Approach\\_for\\_Analyzing\\_and\\_Measuring\\_Weather\\_Data](https://www.researchgate.net/publication/329922758_Weather_Forecast_Prediction_An_Integrated_Approach_for_Analyzing_and_Measuring_Weather_Data)

How transferable are features in deep neural networks?

by Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson

<https://proceedings.neurips.cc/paper/2014/file/375c71349b295fbe2dcdca9206f20a06-Paper.pdf>

A Review on Weather Forecasting Techniques

by Garima Jain , Bhawna Mallick

<https://ijarcce.com/upload/2016/december-16/IJARCCE%2037.pdf>

Other Links :

<https://docs.npmjs.com/>

<https://idratherebwriting.com/learnapidoc/>

<https://developer.mozilla.org/en-US/docs/Web/JavaScript>

<https://ejs.co/>

<https://www.npmjs.com/package/bcrypt>

<https://expressjs.com/>