**FLIGHT DELAY PREDICTION FOR AVIATION INDUSTRY USING**

**MACHINE LEARNING**

**TEAM ID: NM2023TMID25346**

**VARADHARAJAN S (49783B99A3376FE8E0E4303DCFC15894)**

**SUNDAR S (CC15DFB048BDBFA8E8844CFA0C023D6C)**

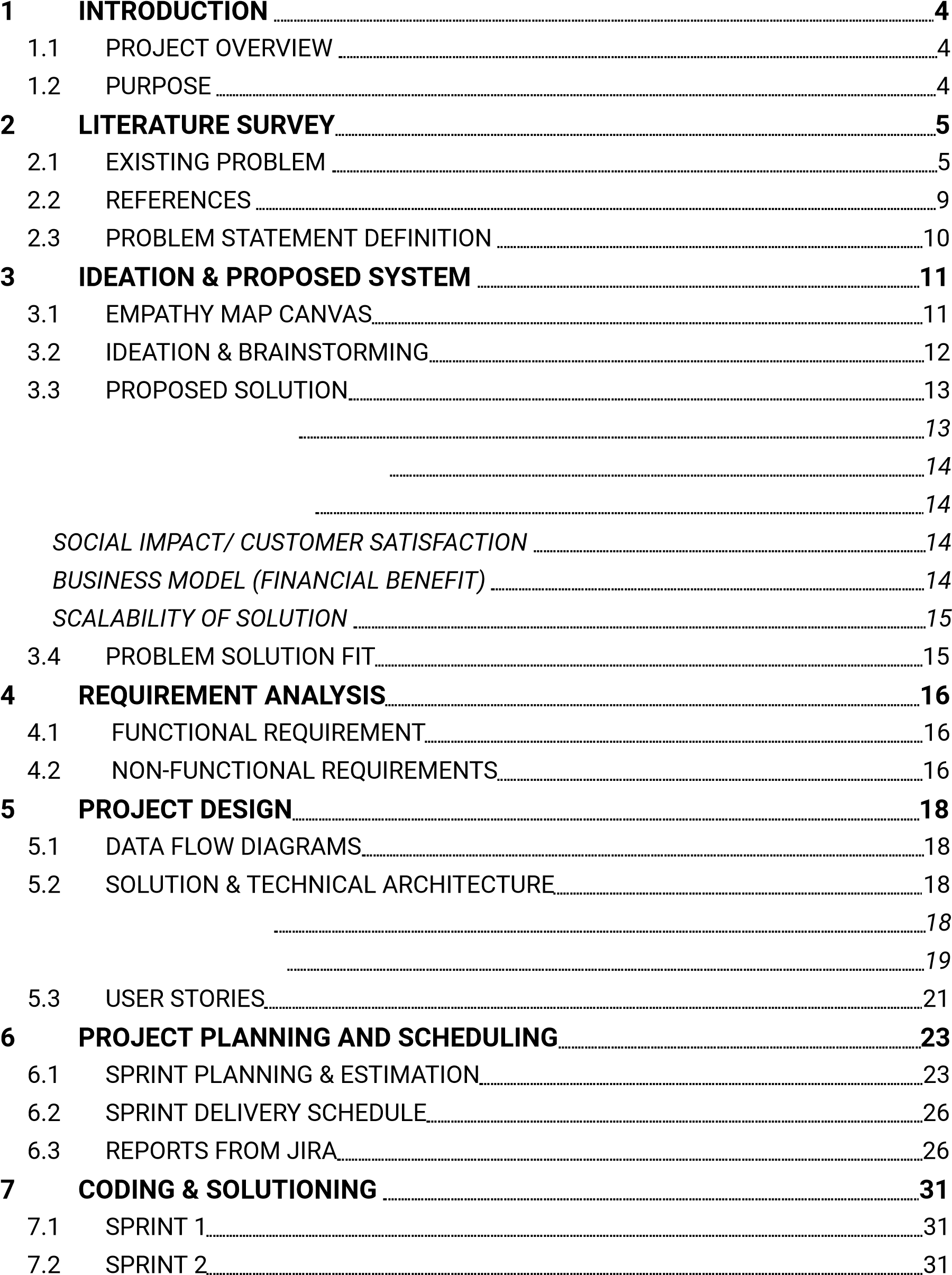
**SELVARAJ V (989DC89FBC2B31287E54516F27833858)**

**VISHNU K (EF8D5312C86C6ABBCB14CD83A529FE21)**

**SUBMITTED FOR THE PROJECT UNDER THE NAAN MUDHALVAN – SMARTINTERNZ**

**PROGRAM**

# TABLE OF CONTENTS

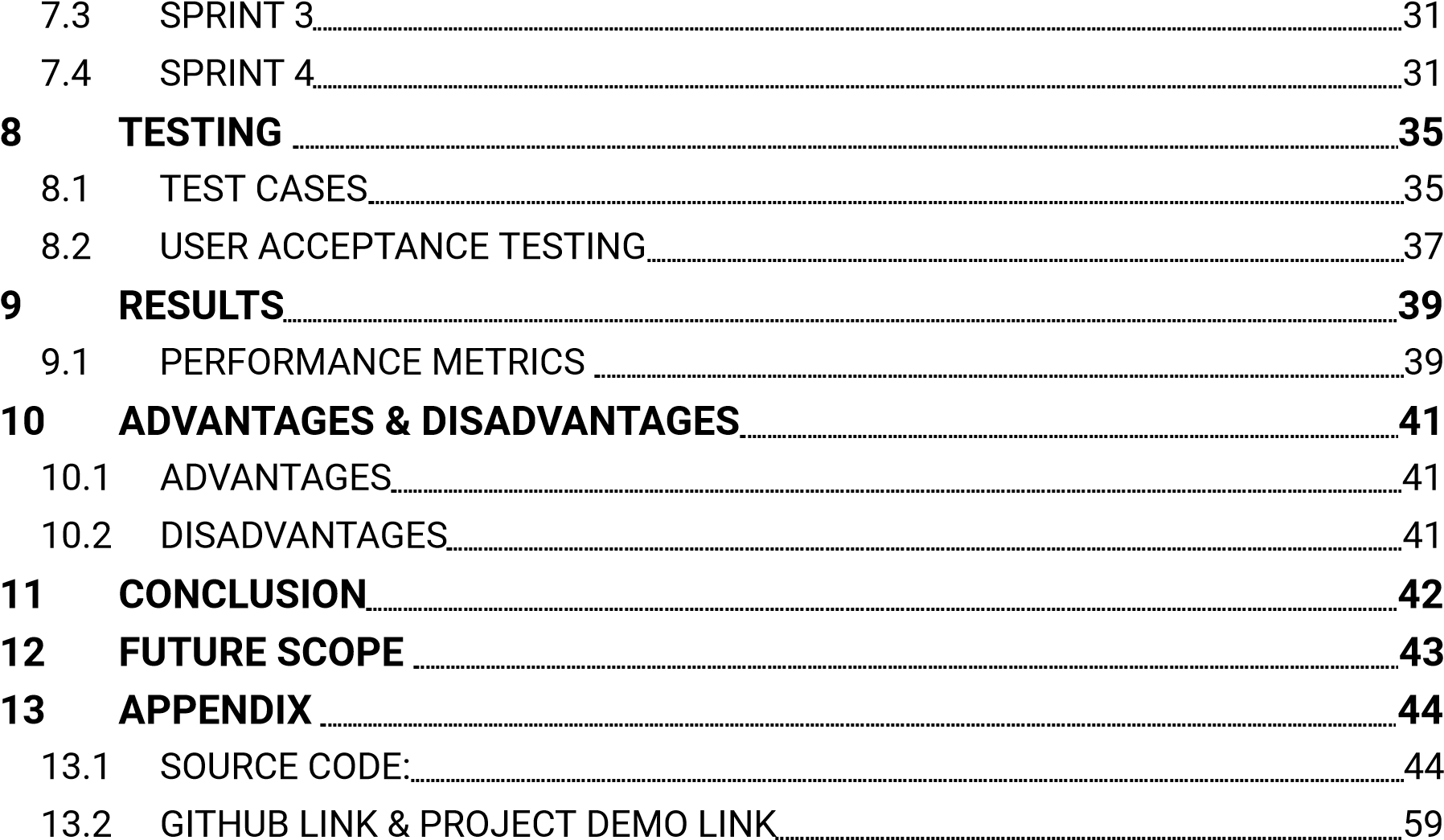
*PROBLEM STATEMENT*

*IDEA/ SOLUTION DESCRIPTION*

*NOVELTY/ UNIQUENESS*

*Solution Architecture*

*Technical Architecture*



**CHAPTER 1**

## 1INTRODUCTION

### 1.1 Project Overview

Due to its quickness and occasional comfort, air travel has become more and more popular among tourists during the past 20 years. The result has been a spectacular increase in land traffic and air traffic. Massive levels of aircraft delays on the ground and in the air have also been brought on by an increase in air traffic. There have been significant monetary and environmental losses as a result of these delays. To optimize flight operations and reduce delays, the model's primary goal is to estimate flight delays accurately.

Flight arrival delays can be predicted using a machine learning algorithm. Rows of feature vectors, such as departure date, delay, travel time between the two airports, and scheduled arrival time, provide the input to our algorithm. The Support Vector Machine is then used to determine whether or not the flight arrival will be delayed. When there is more than a 15-minute gap between the scheduled and actual arrival timings, a flight is deemed to be delayed.

### 1.2 Purpose

Flight delay forecasting can enhance airline operations and passenger happiness, which will boost the economy. Comparing the effectiveness of machine learning classification systems for predicting flight delays is the major objective. Flight delays are unavoidable, and they significantly affect the carriers' profits and losses. The traveler's ability to plan ahead and avoid wasting vital time can be greatly aided by this delay prediction. For airlines, estimating flight delays correctly is essential since the data may be used to boost client happiness and revenue for airline agencies.

**CHAPTER 2**

**LITERATURE SURVEY**

### 2.1 Existing problem

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S.No** | **Author** | **Title of the paper** | **Year of publicati on** | **Algorithm/ Method** | **Results** |
| 1 | Guan Gui,  Fan Liu,  Jinlong  Sun, Jie  Yang, Ziqi  Zhou,  Dongxu Zhao | Flight  Delay  Prediction  Based on  Aviation Big Data and Machine  Learning | 18  November 2019 | Long- Short  Term  Memory (LSTM) based method, Random  Forest based model | Experiment  al results show that long short term memory (LSTM) is capable of handling the obtained aviation sequence data, but an overfitting problem occurs in our limited dataset. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  | Compared with the previous  schemes, the proposed random  forest based model can obtain higher prediction accuracy (90.2% for the binary classificatio n) and can  overcome the overfitting problem |
| 2 | Kaiquan Cai,  Yue Li,  YiPing Fang,  Yanbo Zhu, | A Deep  Learning Approach  for Flight Delay  Prediction  Through TimeEvolvi ng Graphs | 12 August  2021 | Graph Convolution al Neural  Network  (GCN) | Through extensive experiment  s, it has been shown that  the proposed approach |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  | outperfor ms benchmark methods with a satisfying accuracy improveme  nt at the cost of acceptable execution time. The obtained results reveal that deep  learning approaches based on graph structured inputs have great potential in  the flight delay prediction problem. |
| 3 | Zhen Guo,  Bin Yu,  Mengyan | A novel hybrid method for | September 2021 | Hybrid method of  Random | The proposed RFR-MIC |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Hao, Wensi  Wang, Yu  Jiang, Fang Zong | flight  departure delay prediction using Random  Forest Regression and Maximal  Information  Coefficient |  | Forest Regression and Maximal  Information  Coefficient  (RFR-MIC) | model  exhibits good performan ce  compared with linear regression (LR), knearest neighbors  (kNN),  artificial neural network (ANN), and standard  Random  Forest  Regression (RFR). The results also  show that  flight  information on multiple  air routes can certainly improve the  accuracy of  flight  departure delay |
|  |  |  |  |  | prediction. |
| 4 | Fan Liu,  Jinlong Sun,  Miao Liu, Jie  Yang, Guan Gui | Generaliz ed Flight Delay  Prediction  Method  Using  Gradient  Boosting  Decision Tree | 30 June  2020 | Gradient boosting decision tree  (GBDT) based model | Experiment  al results show that the proposed  GBDTbased model can obtain higher prediction accuracy (87.72% for the binary classificatio n) when handling a limited dataset. |

### 2.2 References

https://ieeexplore.ieee.org/document/8903554 https://ieeexplore.ieee.org/document/9512525

https://www.sciencedirect.com/science/article/abs/pii/S1270963821003321 https://ieeexplore.ieee.org/document/9129110

### 2.3 Problem Statement Definition

**Problem Statement:** Passenger flights getting delayed

**Defining the Problem:**

|  |  |
| --- | --- |
| **Question** | **Description** |
| Who does the problem affect? | Passengers who are in emergency situation to reach destination |
| What is the issue? | An airline flight takes off and lands later than its scheduled time. A cancellation occurs when the airline does not operate the flight at all for a certain reason.  However, depending on why the flight was canceled, finding seats on a new flight will also be difficult and may change travel plans |
| When does the issue occur? | The issue occurs when there are adverse weather conditions, if the flight is waiting for cargo or crew or if the airport staff strike. |
| Where is the issue occurring? | Airport |
| Why is it important that we fix the problem? | Flight delays cause inconvenience for both airline companies and passengers. They cause a decrease in efficiency, an increase in capital costs, reallocation of flight crew and aircraft, and additional crew expenses and require the consumption of extra labor, capital, and other inputs necessary in the process. Another impact of flight delay can be a risk which represents dissatisfaction of passengers and their loss in time. |

**CHAPTER 3**

## 3 IDEATION & PROPOSED SYSTEM

### 3.1 Empathy Map Canvas

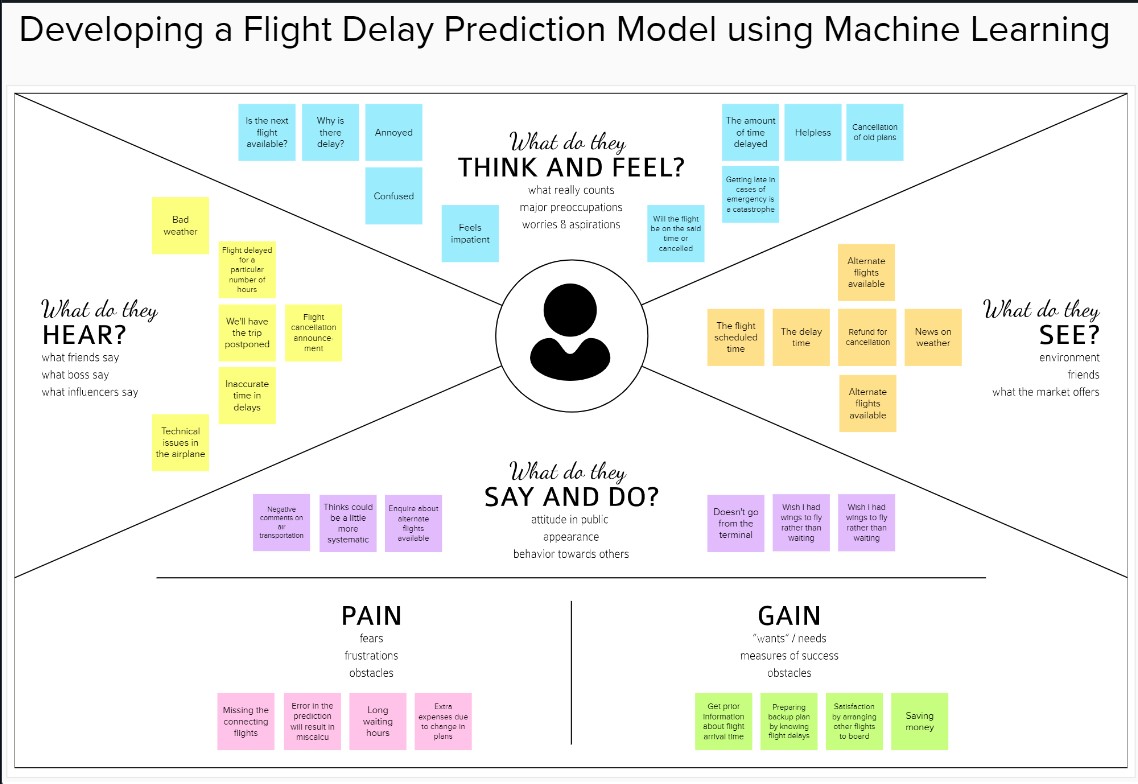


Figure 3.1 Empathy Map Canvas

### 3.2 Ideation & Brainstorming

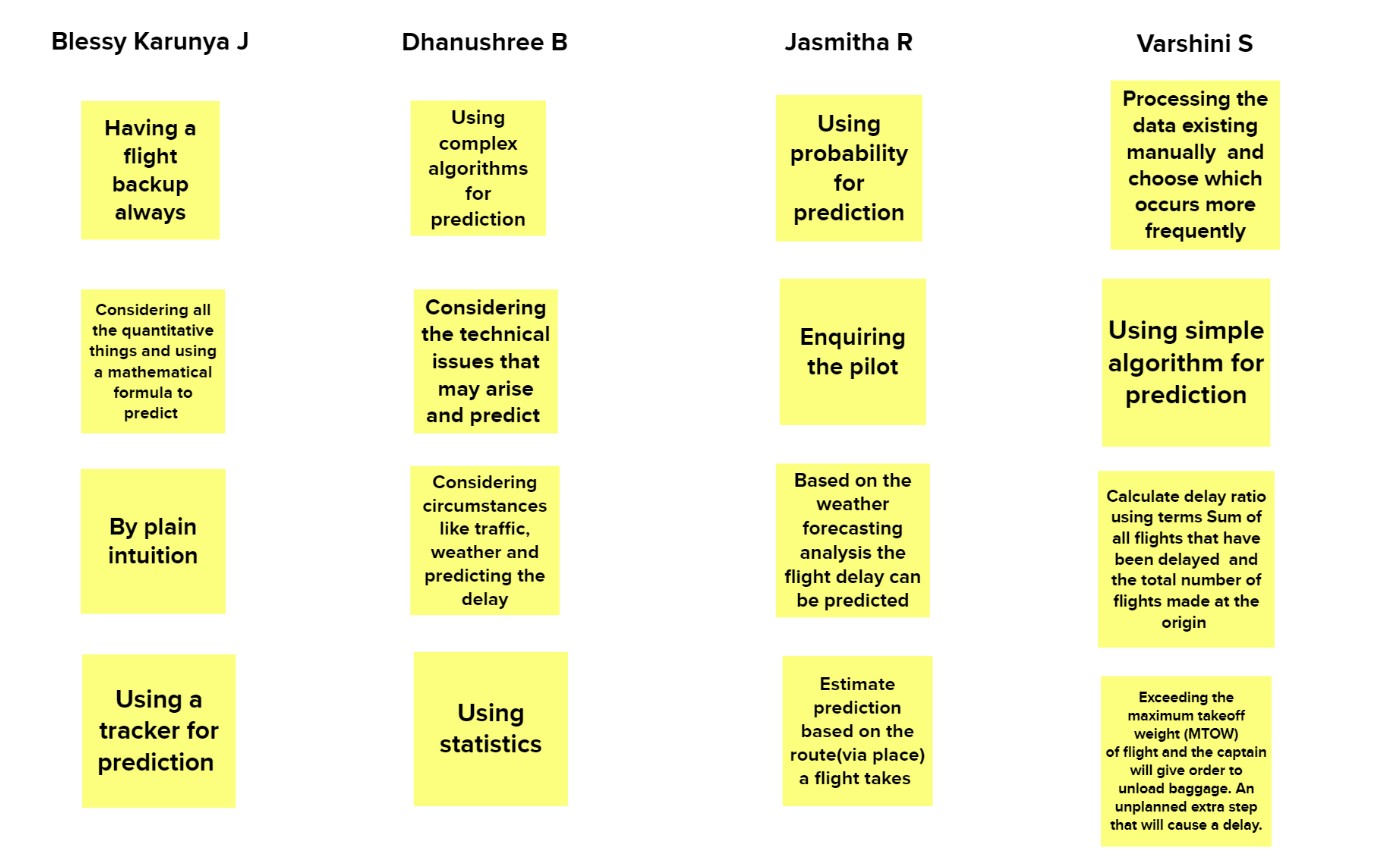


Figure 3.2 Brainstorming

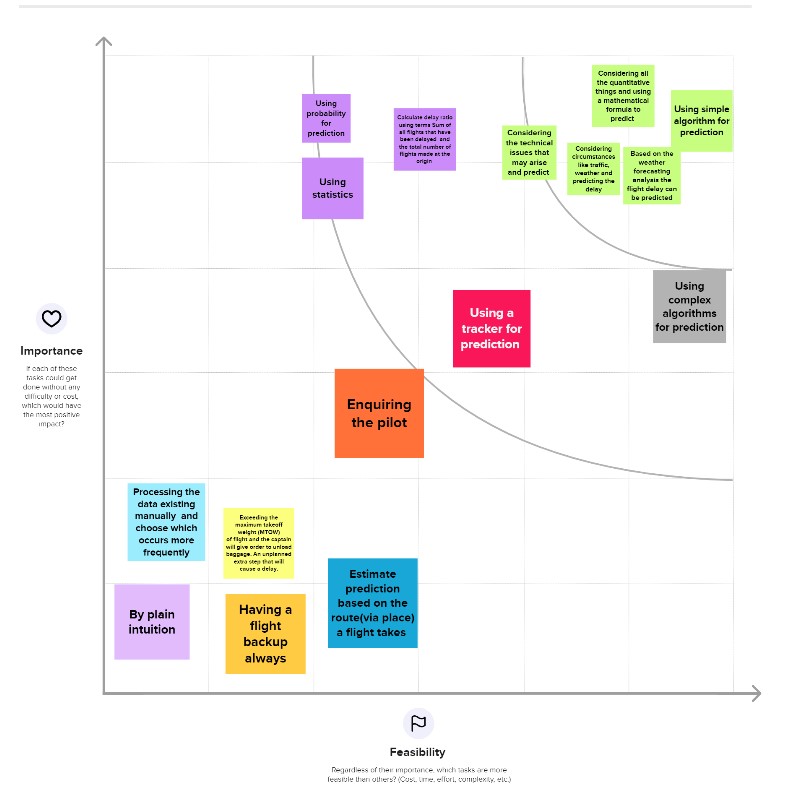


Figure 3.4 Prioritise

### 3.3 Proposed Solution

#### PROBLEM STATEMENT

For several reasons, flying is substantially more exciting for an ordinary traveller than utilizing other modes of transportation. The first is that it is the fastest way of transportation, followed by the comfort it gives in that it is a somewhat quieter and well air-conditioned mode of transit. In addition, it is a more organized style of traveling since staff members support and assist visitors in every way possible. Land traffic and aviation travel have dramatically increased as a result. A rise in air traffic has also resulted in extremely high levels of aircraft delays both on the ground and in the air.

#### IDEA/ SOLUTION DESCRIPTION

The main objective of the model is to accurately estimate airplane delays. The entire system is a web application. We employ HTML, CSS, and JavaScript to make the application user interactive. The flight delay is predicted using a ML model. This model accepts inputs as departure date, departure delay, travel time between the two airports, and other data. By turning the ML model to a pickle file, it is linked to a web application. The ML model here uses random forest as its prediction method. Because of its rapid prediction, impressiveness, excellence with high dimensionality, etc., random forest is selected.

#### NOVELTY/ UNIQUENESS

A hybrid ensemble flight delay prediction model is built by combining the Logistic Regression Model, Decision Tree, Support Vector Machine, K-Nearest Neighbour Model, and the Naive Bayes Model to provide better accuracy and the F1 score of the hybrid model can be compared with that of each individual model. This ensures that the proposed model is best in predicting results.

#### SOCIAL IMPACT/ CUSTOMER SATISFACTION

An accurate estimation of flight delay can help to increase customer satisfaction and incomes of airline agencies. The delay information can reduce anxiety and tension among passengers thus, saving their time and improving their travel experience, which is significant to enhance passengers’ loyalty to an airline.

#### BUSINESS MODEL (FINANCIAL BENEFIT)

Predicting flight delays prevents reallocation of flight crew and aircraft and additional crew expenses. This avoids consumption of extra labour, capital and other inputs necessary in the process thereby saving the capital costs. The proposed solution is also a low-cost model and the customers are not charged for the service they receive.

#### SCALABILITY OF SOLUTION

The model’s ability to increase its performance is possible by building it more accurately with the use of several classification models and selecting the best accurate model among them. Also, through integration of these models, an optimized hybrid model can be obtained in order to result in more scalability. Deploying the ML model into cloud also makes it easy for enterprises to experiment with the model capabilities and scale up. Placing a finished flight prediction model into a live environment can be used for its intended purpose and it is integrated with Flask, so that they can be accessed by end users.

### 3.4 Problem Solution fit



Figure 3.5 Problem Solution Fit

### CHAPTER 4

**4 REQUIREMENT ANALYSIS**

#### 4.1 Functional requirement

|  |  |  |
| --- | --- | --- |
| **FR No.** | **Functional Requirement**  **(Epic)** | **Sub Requirement (Story /**  **Sub-Task)** |
| FR-1 | User Registration | Registration through form  Registration through email |
| FR-2 | User Confirmation | OTP verification for confirmation |
| FR-3 | Dashboard and Search | Search for flights by entering the flight details Get to know the details of the flight and other flights |
| FR-4 | View flight details | View the details of the flight  View if there were any delays previously |
| FR-5 | Display prediction results | Based on the input given by the user predict the delay Display the prediction results to the user |

#### 4.2 Non-Functional requirements

|  |  |  |
| --- | --- | --- |
| **NFR No.** | **Non-Functional**  **Requirement** | **Description** |
| NFR-1 | Usability | Compatible with all browsers |
| NFR-2 | Security | The application will be designed in such a way that the user data won’t be let out and is secure |
| NFR-3 | Reliability | The application will be fast enough and be run on cloud servers to ensure there is no DoS |
| NFR-4 | Performance | High accuracy predictions |
| NFR-5 | Availability | Available 24/7 as it is deployed in cloud |
| NFR-6 | Scalability | Scalable to a very large extent. The only limit is the amount invested to buy  cloud services. |

**CHAPTER 5**

## PROJECT DESIGN

### 5.1 Data Flow Diagrams

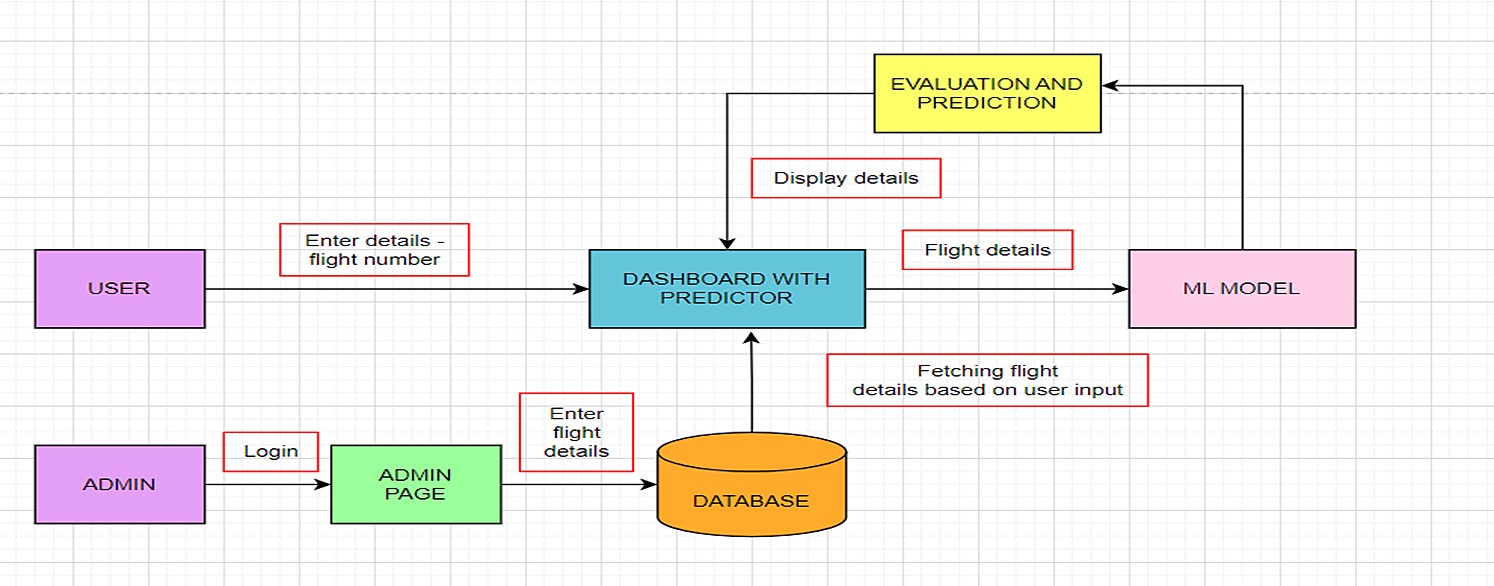


Figure 5.1 Data Flow Diagrams

### 5.2 Solution & Technical Architecture

#### Solution Architecture

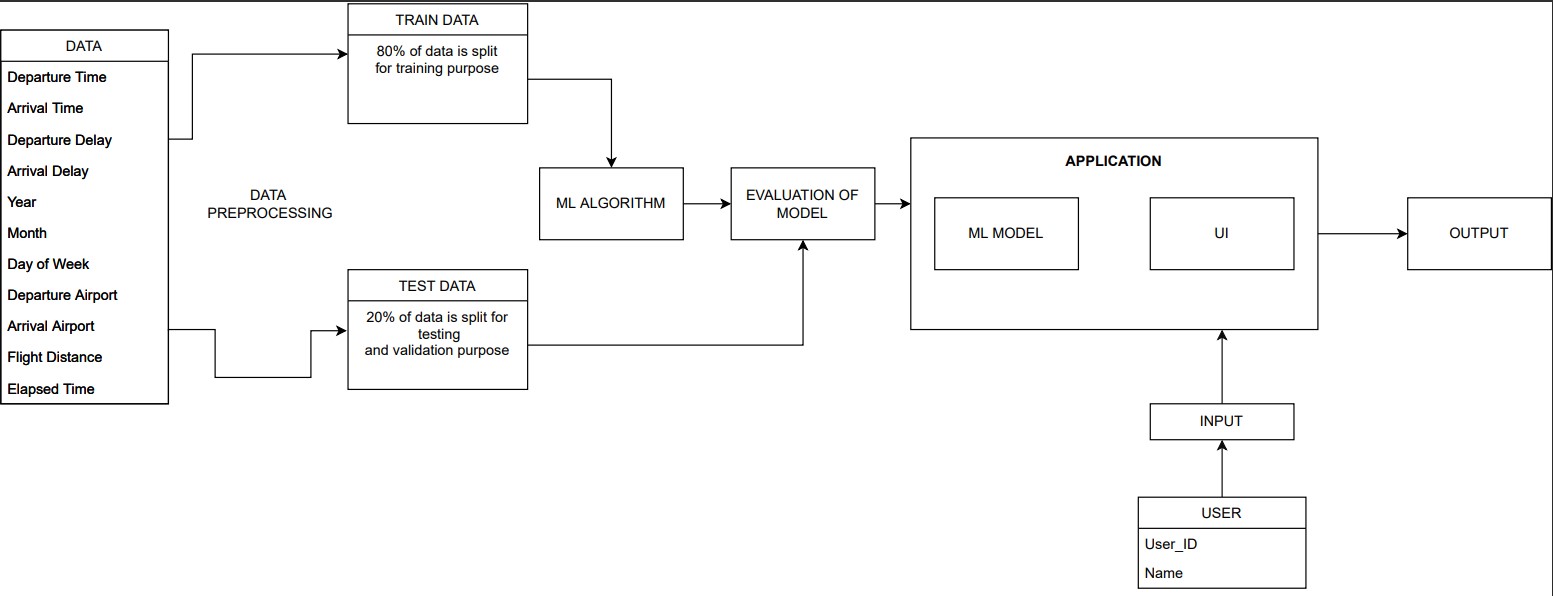


Figure 5.2 Solution Architecture

#### Technical Architecture

The Deliverable shall include the architectural diagram as below and the information as per the table1 & table 2.

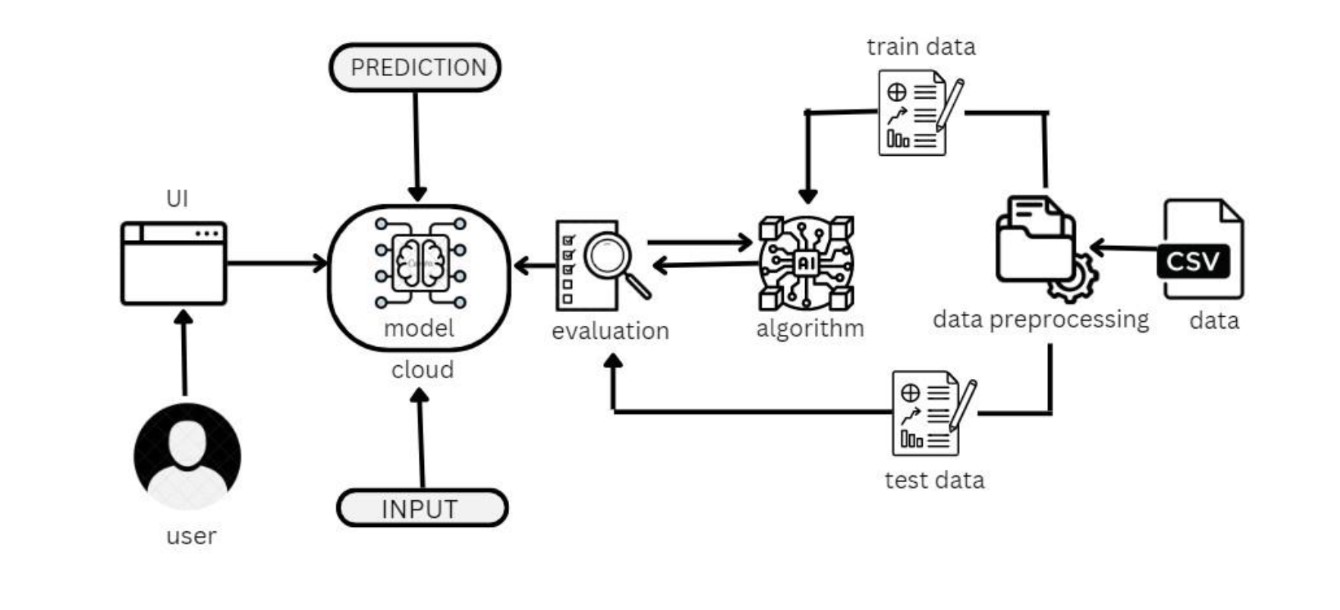


Figure 5.3 Technical Architecture

Table-1: Components & Technologies:

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No.** | **Component** | **Description** | **Technology** |
| 1. | User Interface | How the user interacts with application e.g.  Web UI, Mobile App,  Chatbot etc. | HTML, CSS,  JavaScript /  Angular Js / React  Js etc. |
| 2. | Application Logic-1 | Using Python's regularization approach with Regression Analysis to create predictions about future delays | Python |
| 3. | Application Logic-2 | Build, run and manage AI models | IBM Watson  Machine Learning |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  | service |
| 4. | Database | Data Type,  Configurations etc. | MySQL, NoSQL, etc. |
| 5. | Cloud Database | Database Service on Cloud | IBM DB2, IBM  Cloudant etc. |
| 6. | File Storage | File storage requirements | IBM Block Storage or Other Storage  Service or Local  Filesystem |
| 7. | External API-1 | Defines communication |  |
| between customer and the administration | Flask (Python), etc. |  |  |
| 8. | Machine Learning  Model | To predict flight delay model | Object Recognition  Model, etc. |
| 9 | Infrastructure  (Server / Cloud) | Application  Deployment on  Local System /  Cloud Local Server Configuration: local host server on which flask runs Cloud Server  Configuration: Cloud object storage | Local, Cloud  Foundry,  Kubernetes, etc. |
| Table-2: Application  Characteristics: |  |  |  |
| S.No | Characteristics | Description | Technology |
| 1. | Open-Source  Frameworks | List the opensource frameworks used | Flask(python) |
| 2. | Security  Implementations | List all the security / access controls implemented, use  of firewalls etc. | e.g. SHA-256, Encryptions, IAM  Controls, OWASP  etc. |
| 3. | Scalable  Architecture | Justify the scalability of architecture (3 – tier, Micro-services) | Flask or ML |
| 4. | Availability | Justify the availability of application (e.g. use of load balancers, distributed servers etc.) | Flask or ML |
| 5. | Performance | Design consideration for the performance of the application (number of requests per sec, use of |  |
| Cache, use of  CDN’s) etc. | Flask or ML |  |  |

### 5.3 User Stories

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **User Type** | **Functional Requirem**  **ent (Epic)** | **User Story**  **Number** | **User Story**  **/ Task** | **Acceptan**  **ce criteria** | **Priority** | **Release** |
| Customer (Web user) | Dashboa rd and Search | USN-1 | As a user, I can search for flights with the flight  number | I can receive informati on on various flights | High | Sprint-1 |
|  | View | USN-2 | As a user, I can view the details of the flights | I get the flight details such as  flight  number, departure and arrival time, etc. | Medium | Sprint-1 |
|  | Display | USN-3 | As a user, I can see the predicted flight delay | I get the informati on about how long  the flight has been delayed | High | Sprint-2 |
| Administr ator | GPS | USN-4 | As an admin, I need to know the location of flights | I can keep track of where the flights are | High | Sprint 3 |
|  | Analyze | USN-5 | As an admin, I can analyze the dataset and come up with reports | I can analyze the dataset | High | Sprint 2 |

**CHAPTER 6**

## 6 PROJECT PLANNING AND SCHEDULING

### 6.1 Sprint Planning & Estimation

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Sprint** | **Functional Requirem**  **ent (Epic)** | **User Story**  **Number** | **User Story**  **/ Task** | **Story Poin**  **ts** | **Priority** | **Team**  **Members** |
| Sprint-1 | Registrati on | USN-1 | As a user, I can register for the applicati on by entering my email, password, and confirmi ng my password. | 2 | High | Blessy  Karunya J |
| Sprint-2 | Confirmati on email | USN-2 | As a user,  I will  receive confirmati on email once I have registered for the applicatio n. | 2 | Medium | Dhanushree B |
| Sprint-1 | User Login | USN-3 | As a user, I can log into the | 2 | High | Jasmitha R |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  | applicati on by entering email & password. |  |  |  |
| Sprint-2 | Admin panel | USN-4 | As an admin, I can authentica te the registrati on and login credentia ls of the  passenger  s. | 2 | High | Varshini S |
| Sprint-3 | Arrival and Departure time of flights | USN-5 | As a user, I can find all the details of a specific flight with its number or name. | 2 | High | Blessy  Karunya J |
| Sprint-3 |  | USN-6 | As a user, I can find exactly how long  the flight will be delayed. | 2 | High | Dhanushree B |
| Sprint-4 | Helpdesk | USN-7 | As a customer care executive,  I can | 1 | Medium | Jasmitha R |
|  |  |  | provide airline details for contact. |  |  |  |
| Sprint-4 |  | USN-8 | As a passenge  r, I am able to find available alternative flights to the destinatio n. | 1 | High | Blessy  Karunya J  Dhanushree B  Jasmitha R Varshini S |
| Sprint-4 | Feedback | USN-9 | As a user, I can provide my suggestio ns and feedback for the  improvem ent of the applicatio n. | 2 | Medium | Varshini S |

### 6.2 Sprint Delivery Schedule

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Sprint | Total  Story  Points | Duration | Sprint  Start Date | Sprint End  Date  (Planned) | Story  Points Complet  ed (as on Planned  End Date) | Sprint  Release  Date  (Actual) |
| Sprint-1 | 4 | 6 Days | 27 Oct  2022 | 01 Nov  2022 | 4 | 01 Nov  2022 |
| Sprint-2 | 4 | 6 Days | 02 Nov  2022 | 07 Nov  2022 | 4 | 07 Nov  2022 |
| Sprint-3 | 4 | 6 Days | 08 Nov  2022 | 03 Nov  2022 | 4 | 03 Nov  2022 |
| Sprint-4 | 4 | 6 Days | 14 Nov  2022 | 19 Nov  2022 | 4 | 19 Nov  2022 |

### 6.3 Reports from JIRA

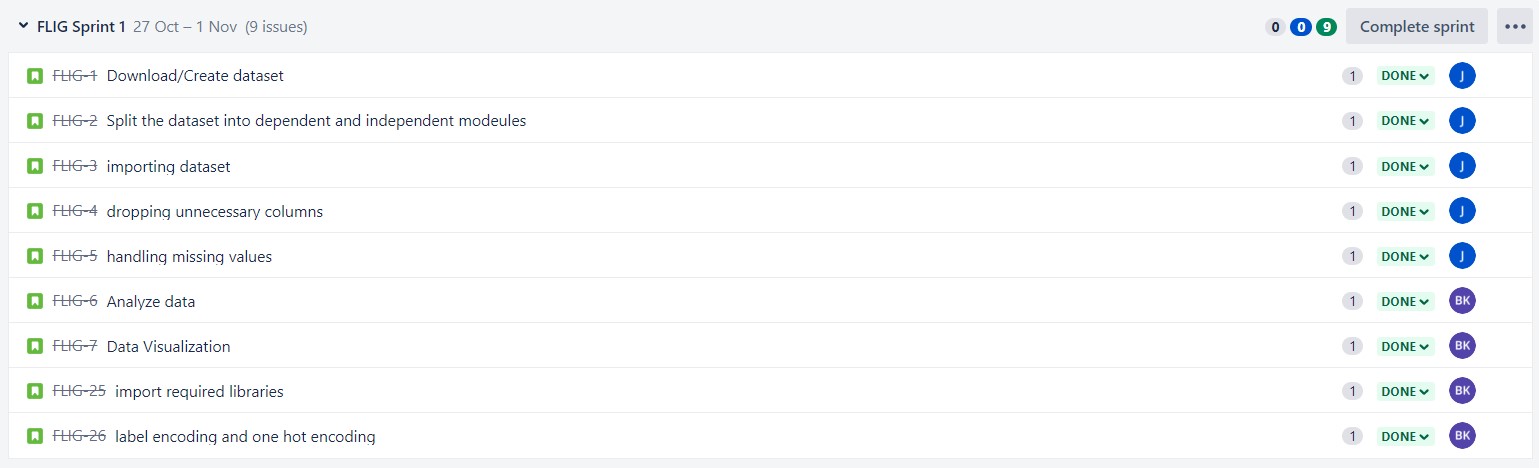


Figure 6.1 Tasks to be performed in Sprint 1

The above figure displays the various tasks to be performed in sprint 1. The goal of the sprint 1 is to prepare data for model training. There are nine issues that need to be addressed in the sprint 1. The story points of each issue is mentioned in the above diagram.

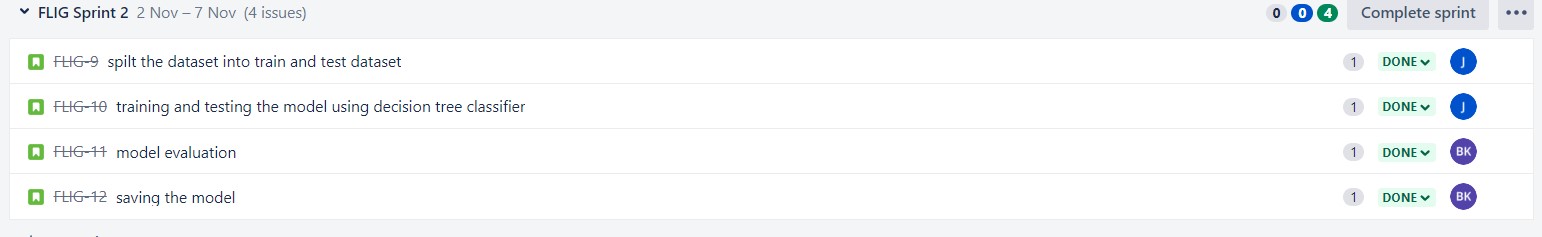


Figure 6.2 Tasks to be performed in Sprint 2

The above figure displays the various tasks to be performed in sprint 2. The goal of the sprint 2 is to train and save the model. There are four issues that need to be addressed in the sprint 2. The story points of each issue is mentioned in the above diagram.

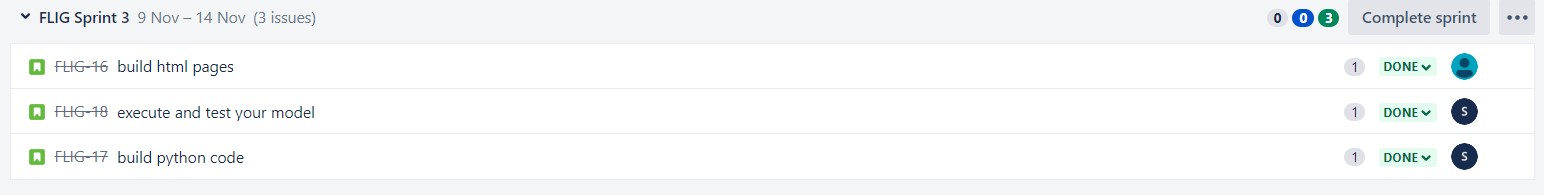


Figure 6.3 Tasks to be performed in Sprint 3

The above figure displays the various tasks to be performed in sprint 3. The goal of the sprint 3 is to build the application and execute the model. There are three issues that need to be addressed in the sprint 3. The story points of each issue are mentioned in the above diagram.

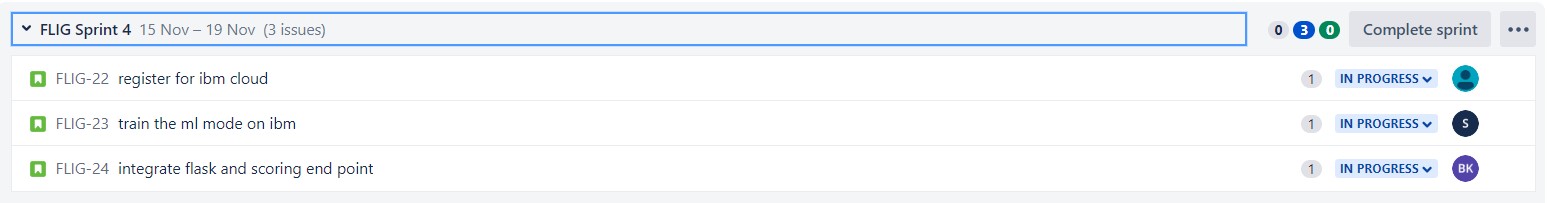


Figure 6.4 Tasks to be performed in Sprint 4

The above figure displays the various tasks to be performed in sprint 4. The goal of the sprint 4 is to integrate the model with the application. There are three issues that need to be addressed in the sprint 3. The story points of each issue are mentioned in the above diagram.

***Burn up report***

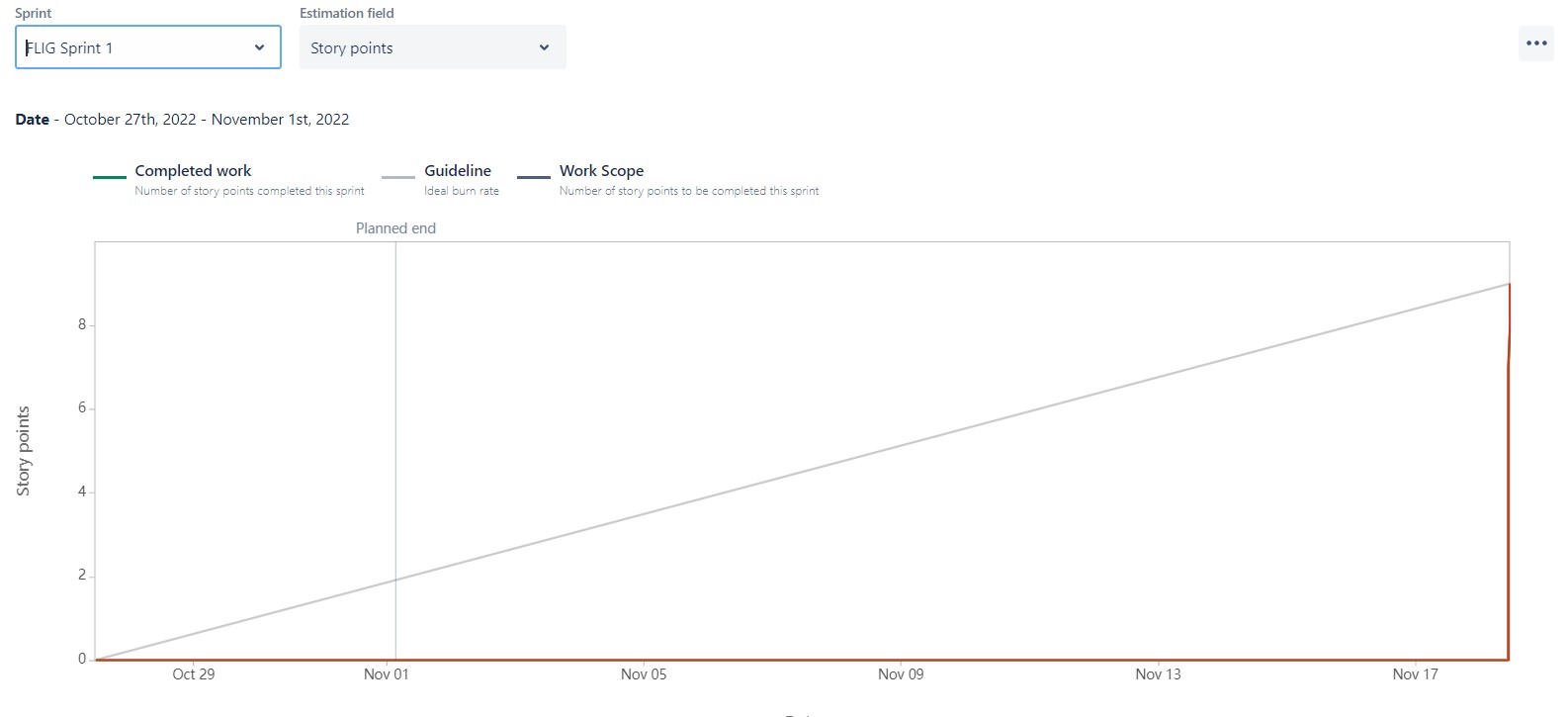


Figure 6.5 Burnup Report of Sprint 1

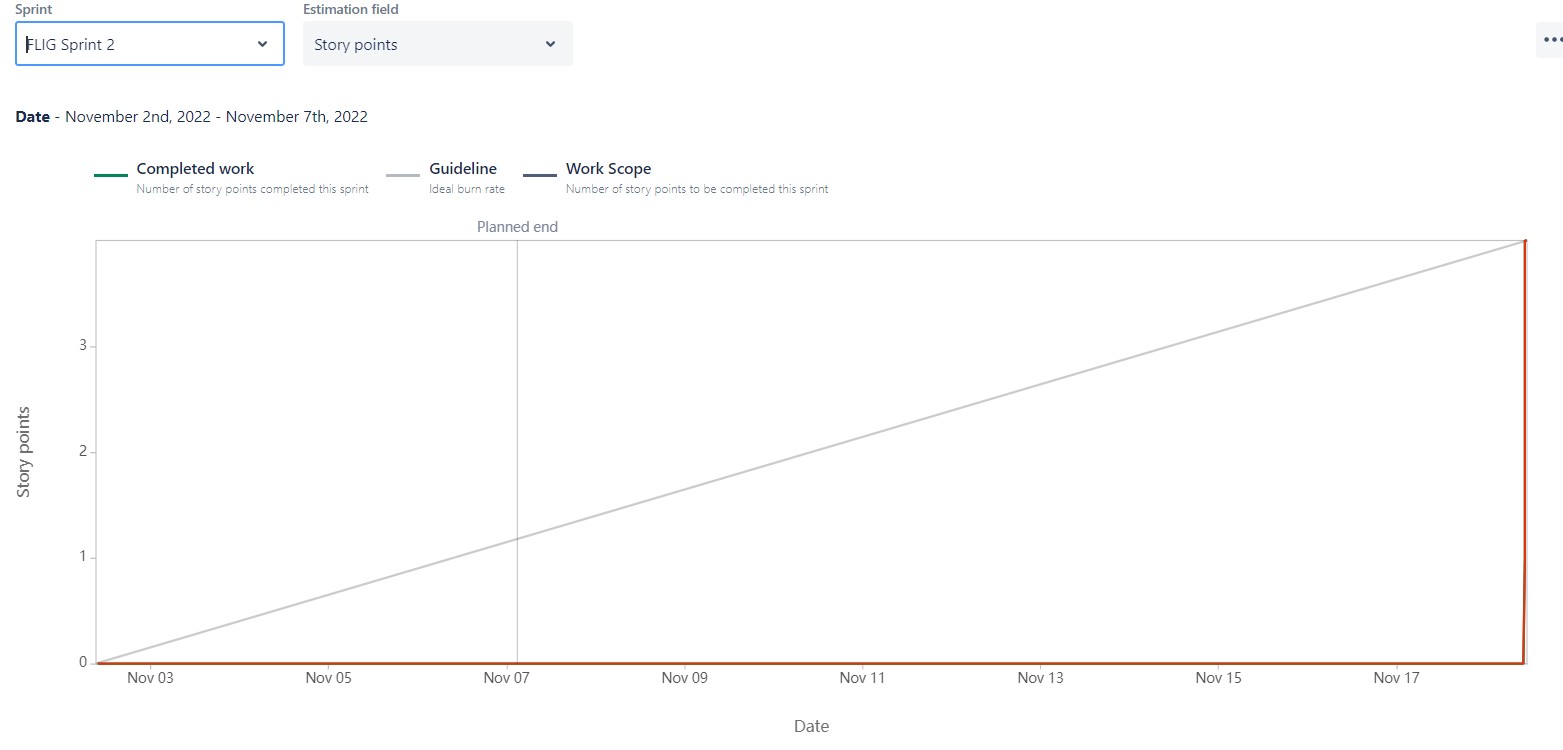


Figure 6.6 Burnup Report of Sprint 2

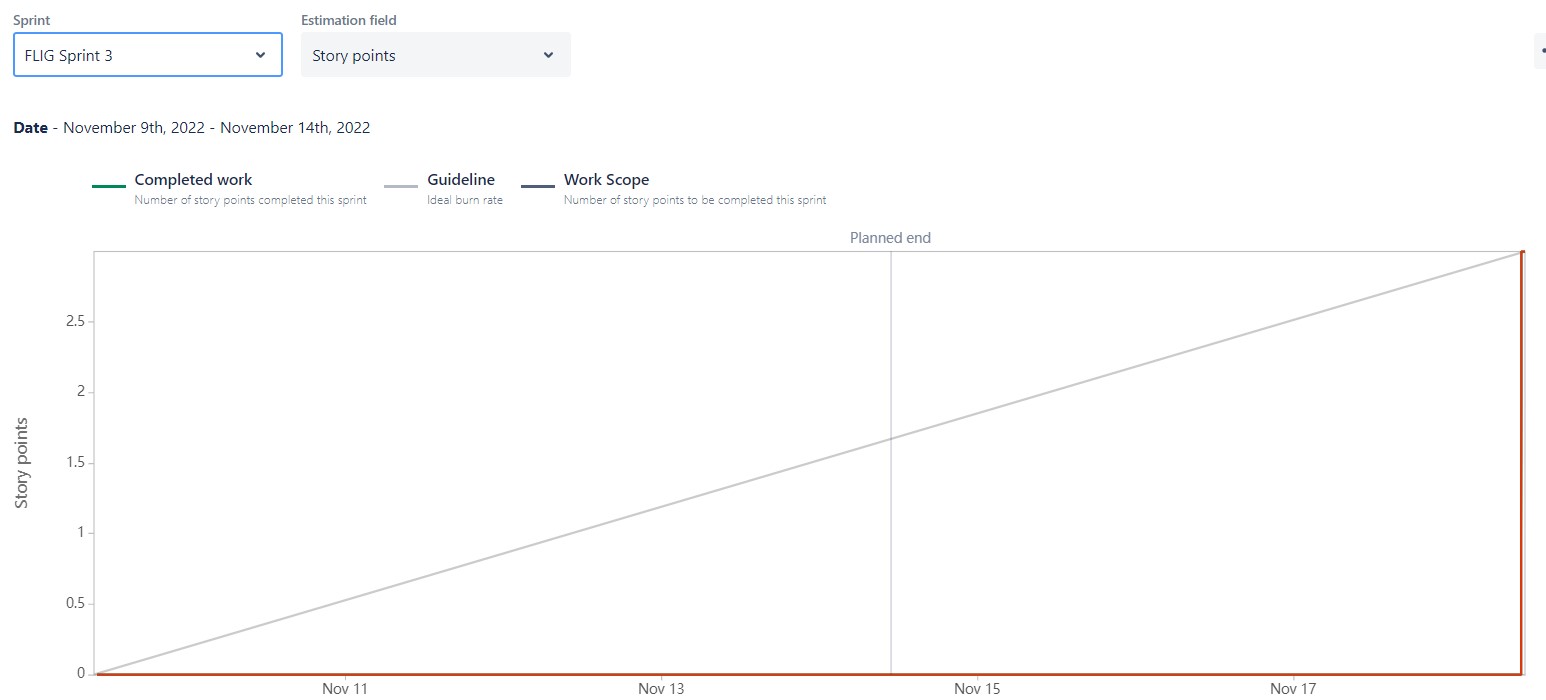


Figure 6.7 Burnup Report of Sprint 3

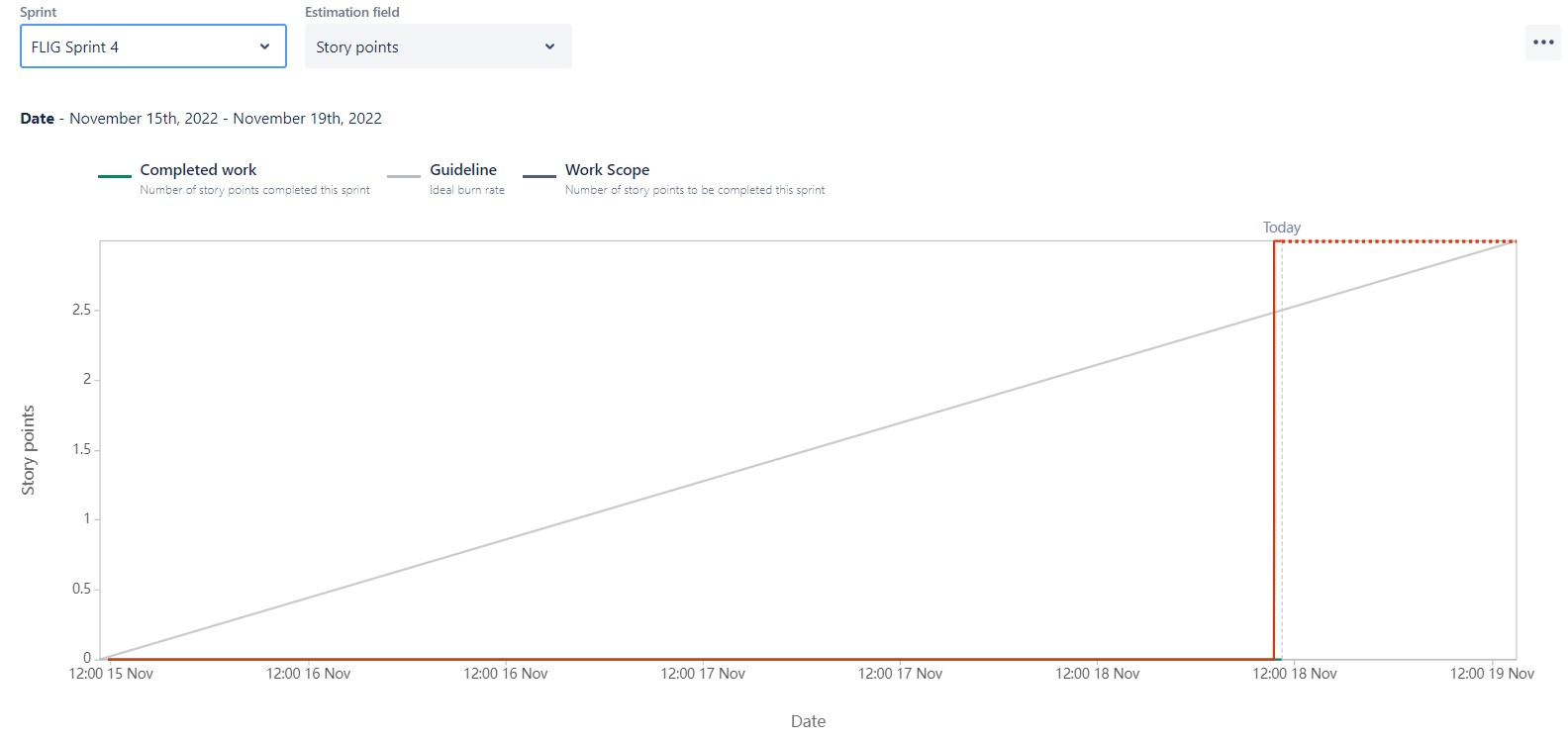


Figure 6.8 Burnup Report of Sprint 4

***Cumulative flow***

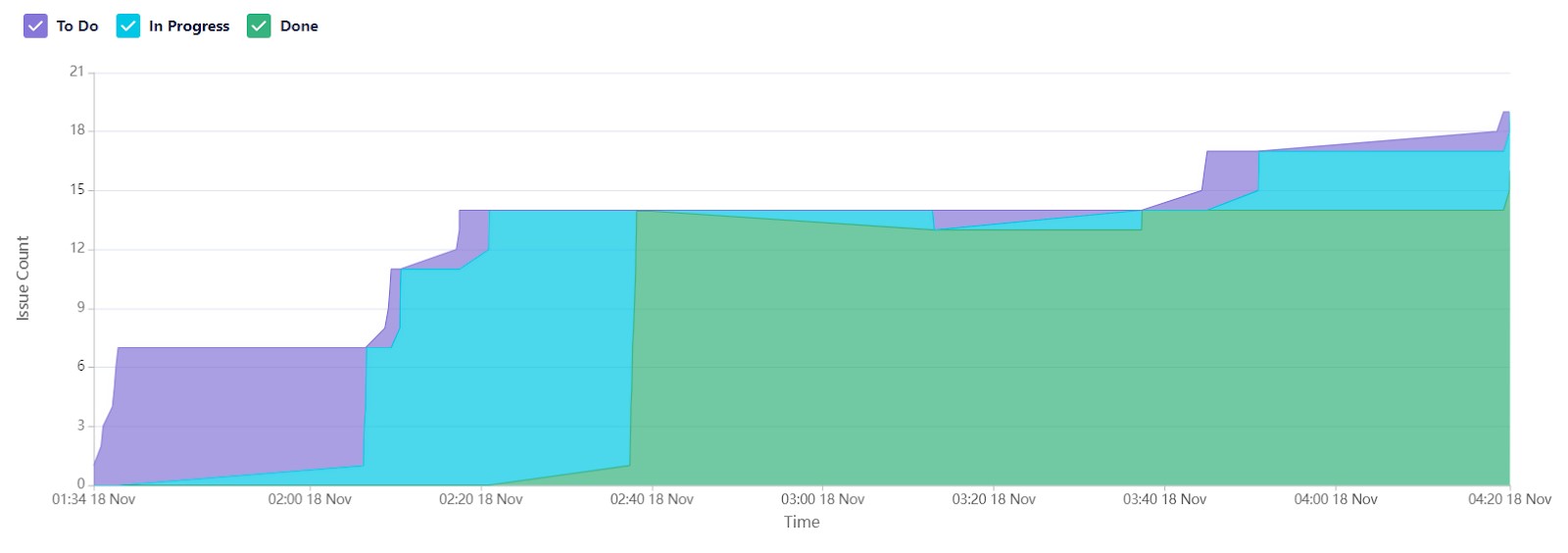


Figure 6.9 Cumulative flow diagram

**CHAPTER 7**

## 7 CODING & SOLUTIONING

We completed four sprints—Sprint 1, Sprint 2, Sprint 3 and Sprint 4—during the project development phase. A sprint is a predetermined amount of time in Agile product development during which particular tasks must be finished and prepared for review.

### 7.1 Sprint 1

The dataset has been downloaded. The features are analysed and visualized and data has been cleaned and pre-processed using techniques like encoding. The independent and dependent variables are then identified and the dataset is split into train and test sets. Several machine learning algorithms have been applied for classification like logistic regression, decision tree classifier, KNN classifier, random forest classifier and it is found that logistic regression gives the highest accuracy, so it is used for deployment. The model is then dumped into a pickle file.

### 7.2 Sprint 2

We had done building HTML files, writing Python code, and running the application during Sprint 2. The source code is attached in the appendix for reference.

### 7.3 Sprint 3

We then asked users to enter numerical and selection data and tested for many inputs and checked the correctness of the result during sprint 3.

### 7.4 Sprint 4

We trained the model on IBM where we will register for IBM cloud, train the ML model on

IBM and integrate flask with scoring end point. Registered on IBM cloud and activated Watson machine learning, cloud storage and Watson studio then trained the ML model on IBM using API KEY during sprint 4.

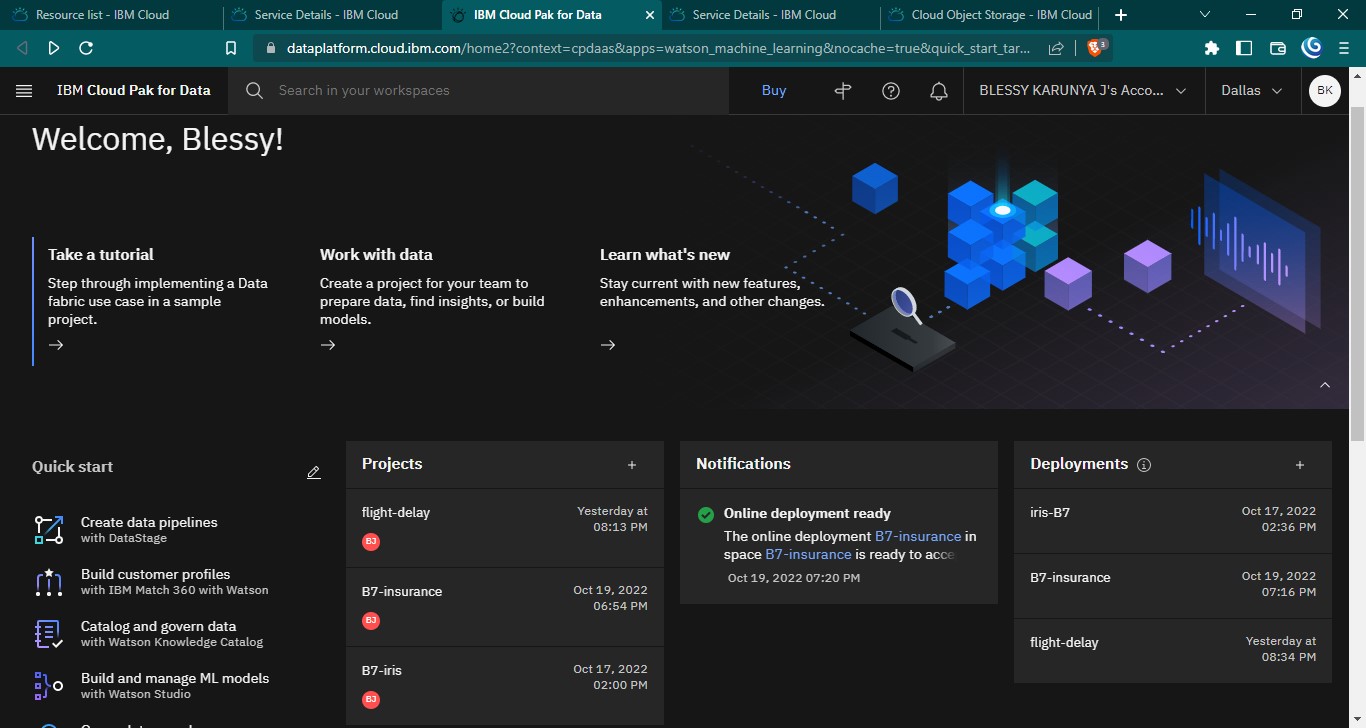


Figure 7.1 Watson Studio

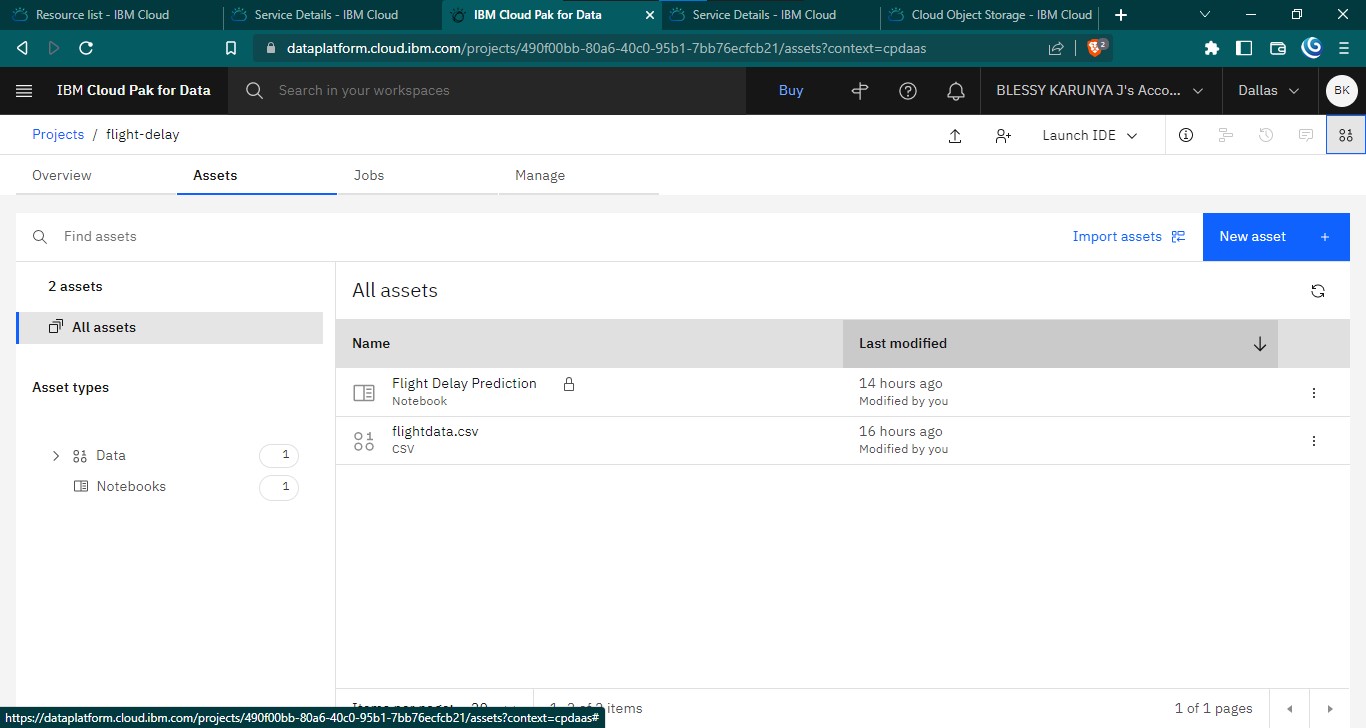


Figure 7.2 Assets of project

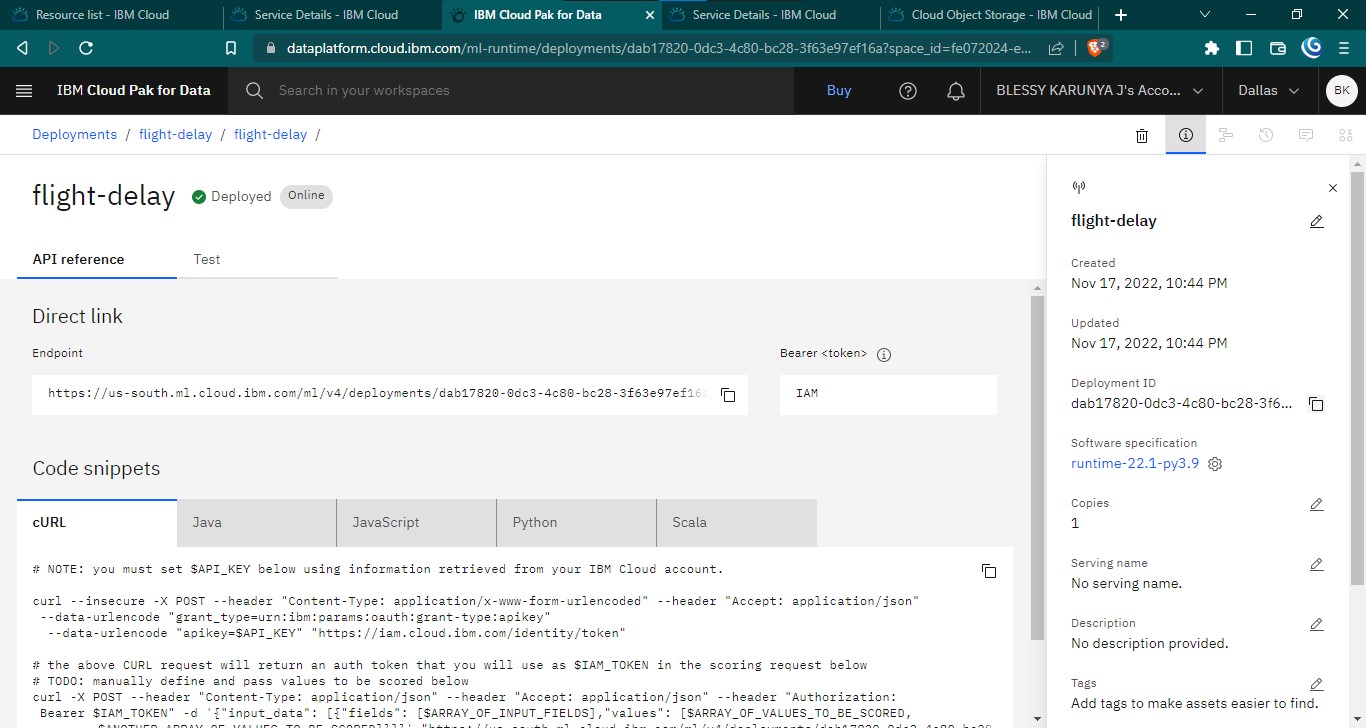


Figure 7.3 API reference of deployment

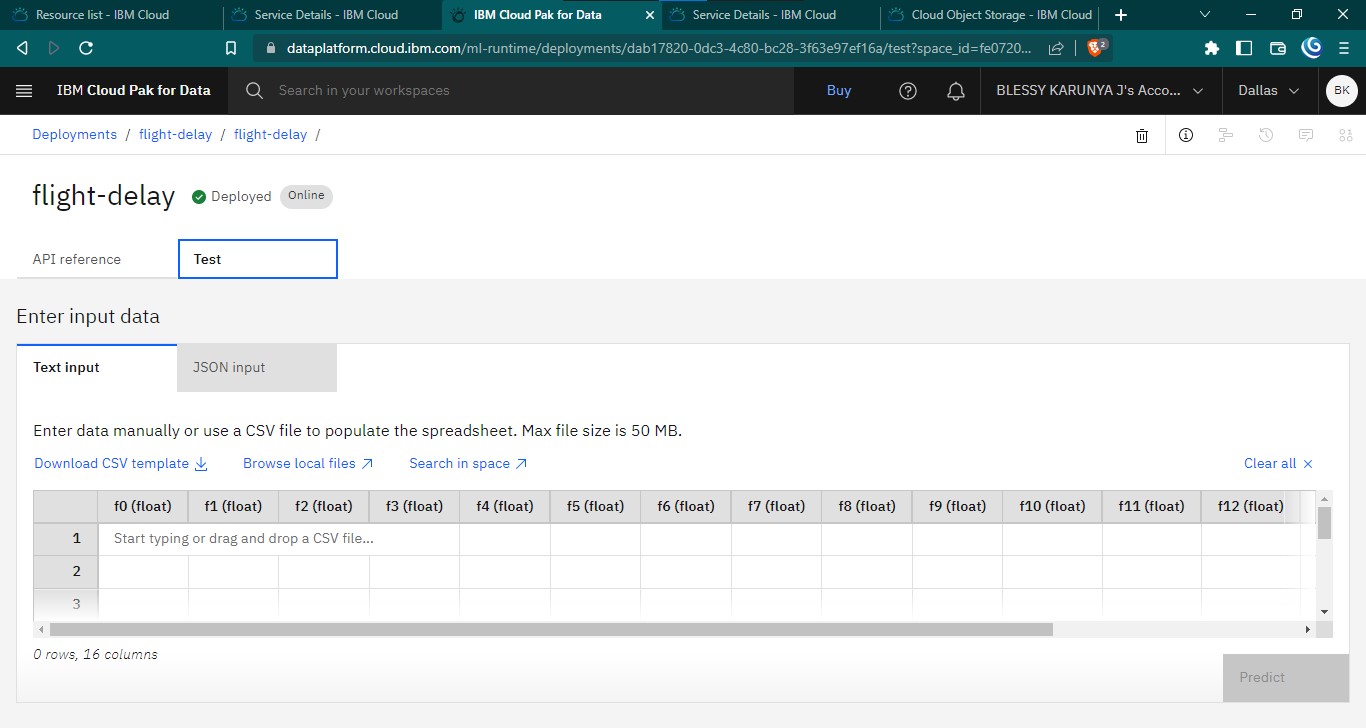


Figure 7.4 Deployment testing phase

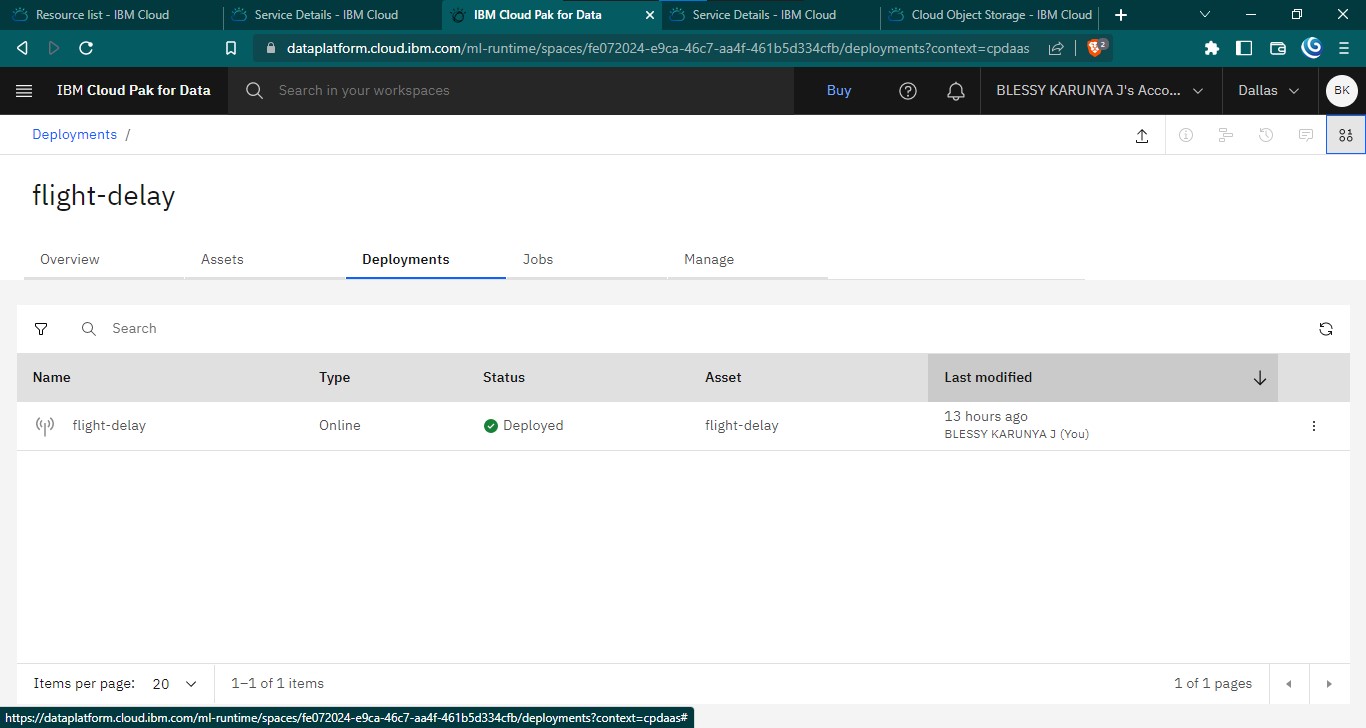


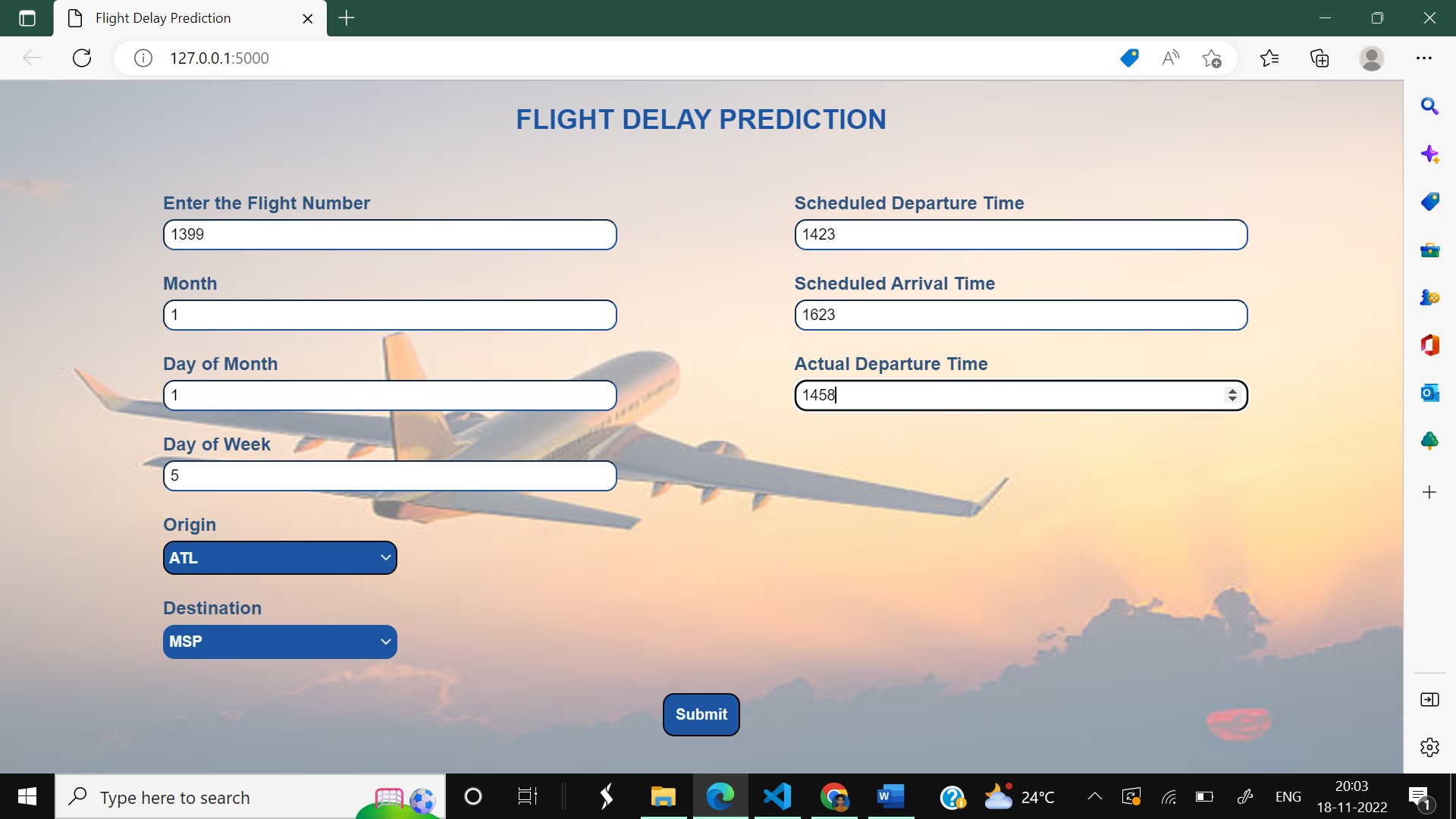
Figure 7.5 Deployment

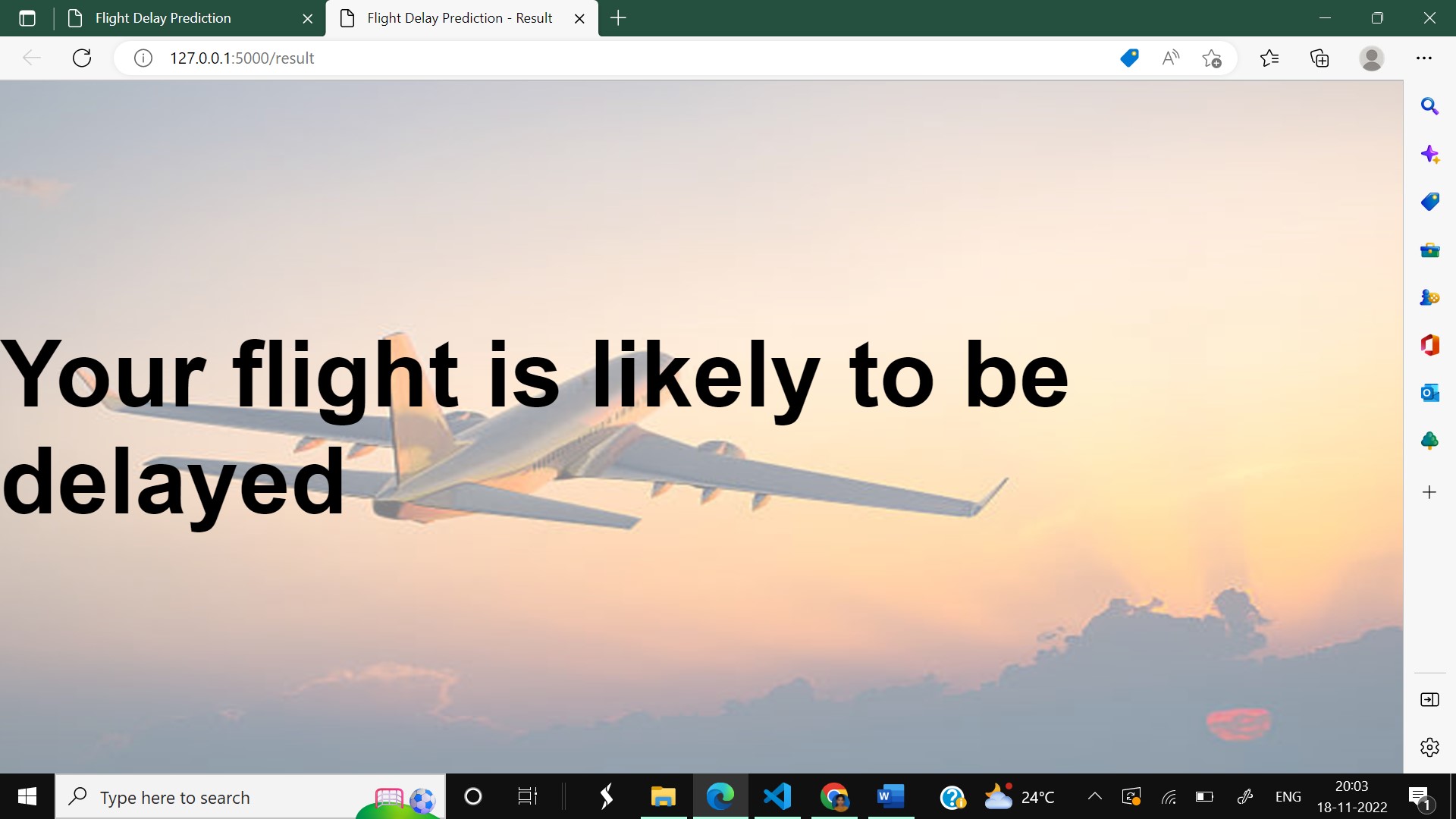
**CHAPTER 8**

## 8 TESTING

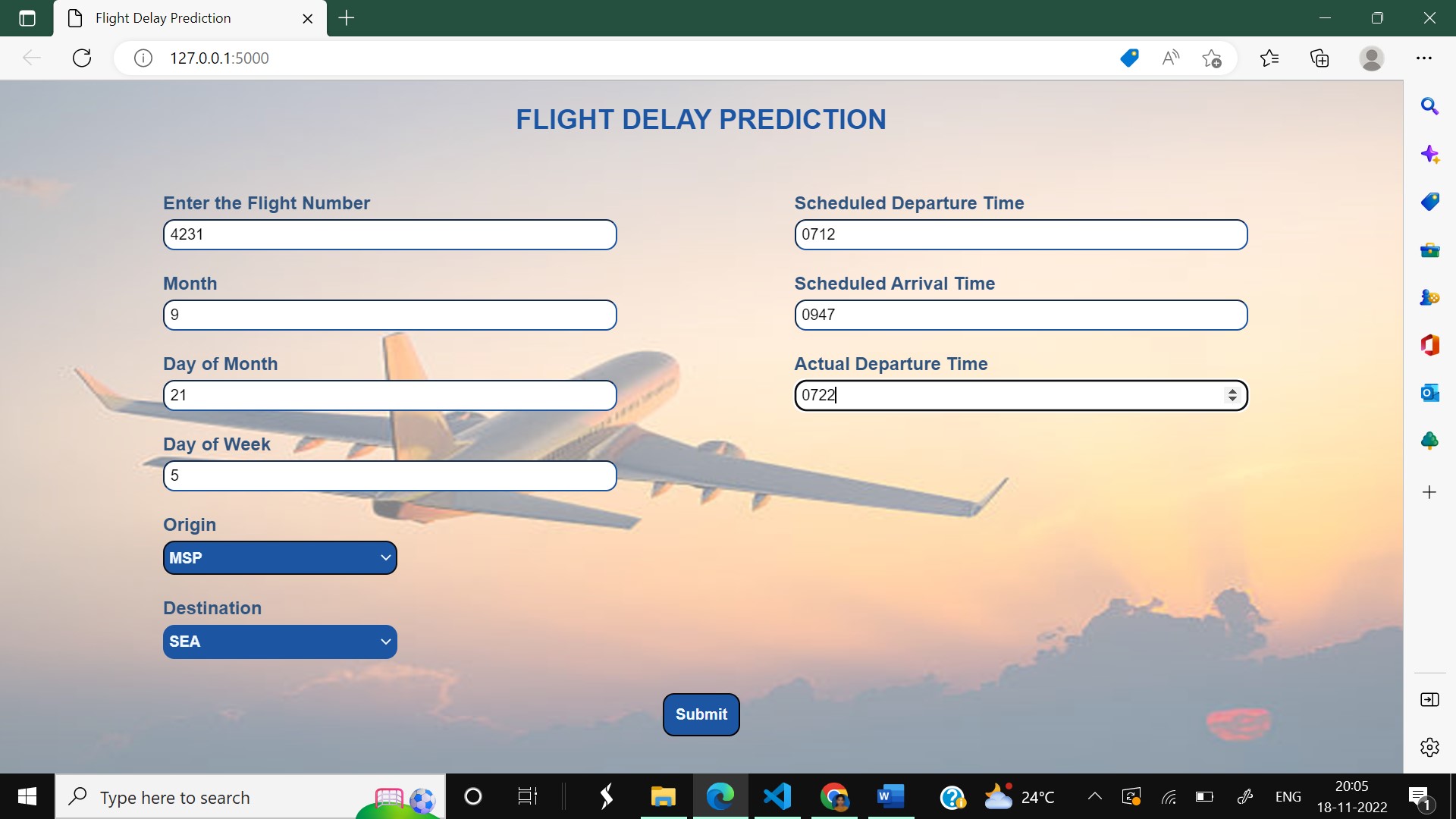
### 8.1 Test Cases

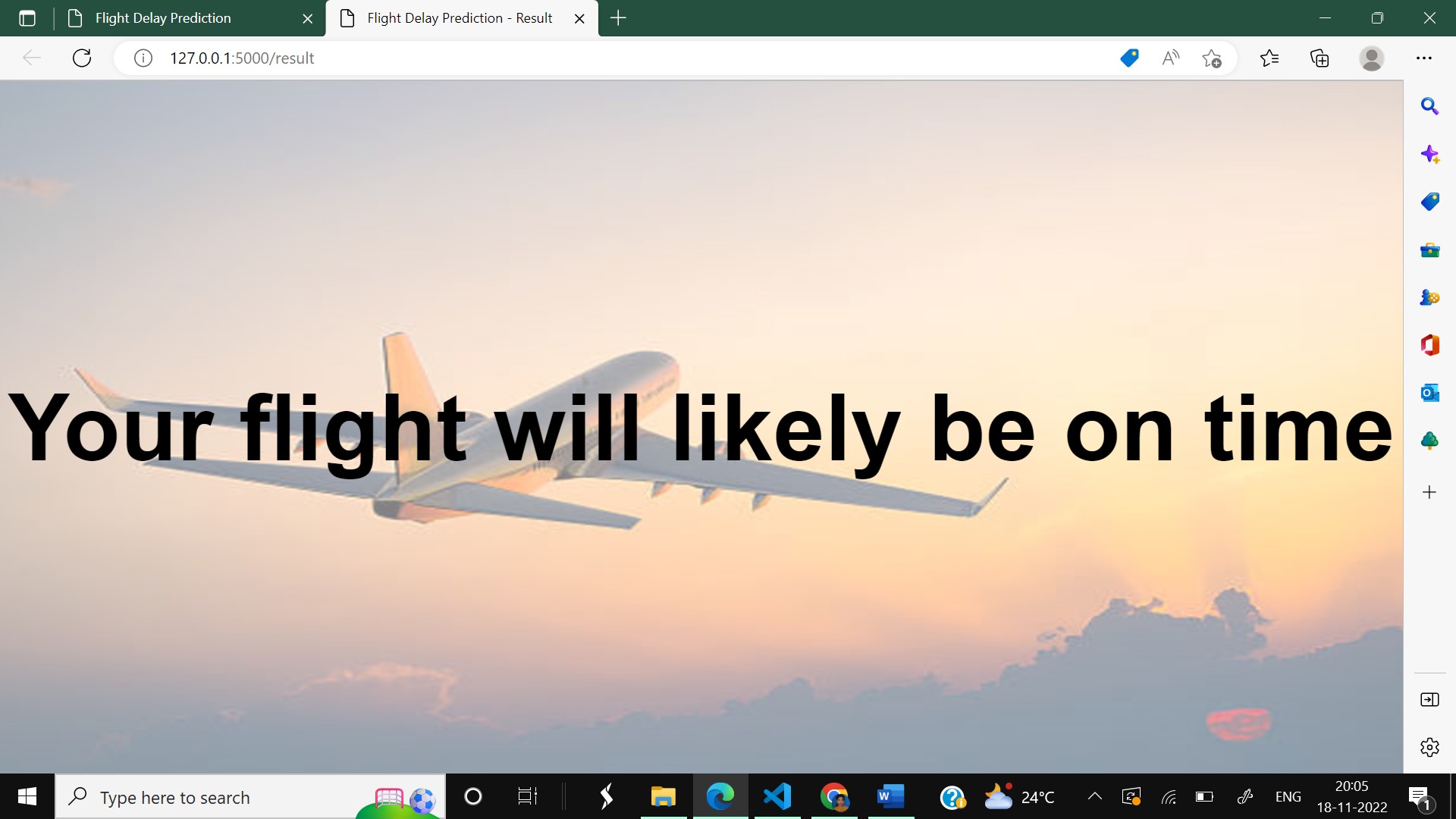
**Test case 1**: The designed application predicts if the flight is delayed.





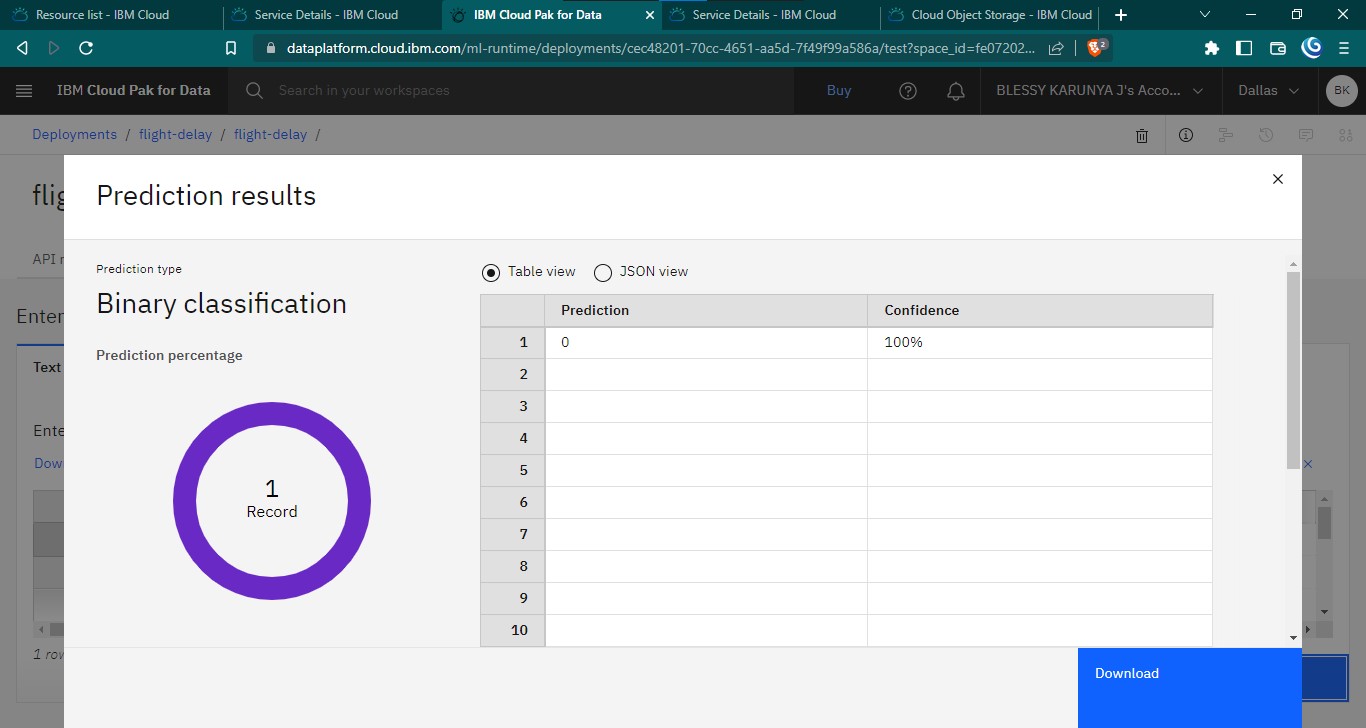
**Test case 2:** The designed application predicts if the flight is not delayed.



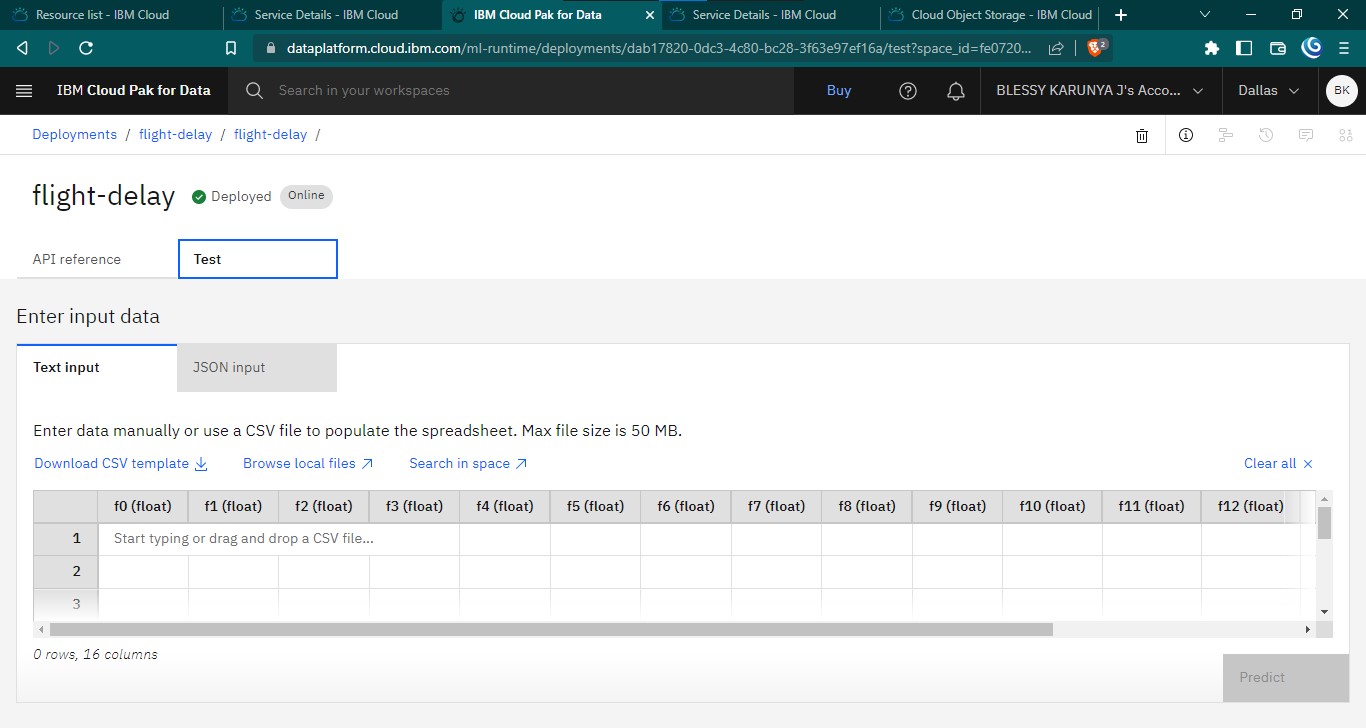


### 8.2 User Acceptance Testing

1. Is the model tested in IBM cloud? YES



1. Deployment testing phase: DONE



1. Can the user be able to make use of cloud-deployed ML model and run it in local machine using the API key? YES

The below table shows the number of test cases that have passed, failed and untested.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Section** | **Total cases** | **Not Tested** | **Fail** | **Pass** |
| Home Screen | 1 | 0 | 0 | 1 |
| User input | 4 | 0 | 0 | 4 |
| Flight delay testing | 2 | 0 | 0 | 2 |
| No flight delay testing | 2 | 0 | 0 | 2 |
| Version control | 1 | 0 | 0 | 1 |

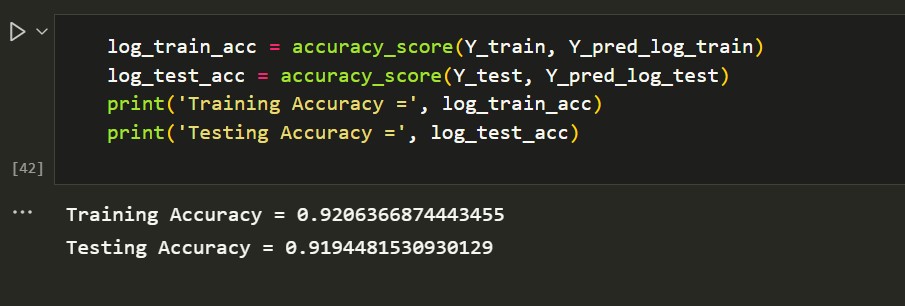
**CHAPTER 9**

## 9 RESULTS

### 9.1 Performance metrics

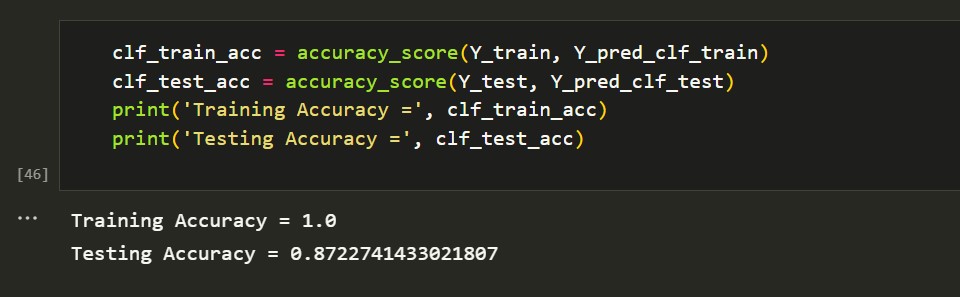
#### Model: Logistic Regression performance values

There is no big variation in the training and testing accuracy. Therefore, the Logistic Regression model is not overfit or underfit.



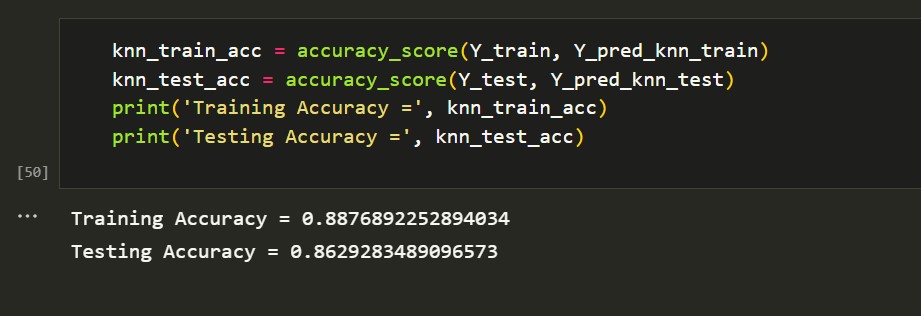
#### Model: Decision tree Classifier

There is a variation in the training and testing accuracy. The Decision tree classifier model is overfit. So, it is not chosen for best results.



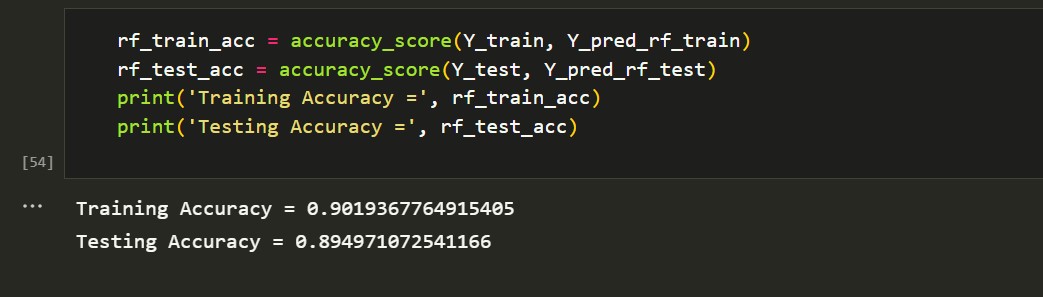
#### Model: KNN Classifier

There is no big variation in the training and testing accuracy. Therefore, the KNN Classifier model is not overfit or underfit.



#### Model: Random Forest Classifier

There is no big variation in the training and testing accuracy. Therefore, the Random Forest Classifier model is not overfit or underfit.



On comparing the four models built, based on the performance metrics it is clear that logistic regression gives the highest performance. Hence, that model is chosen for deployment.

## 10 ADVANTAGES & DISADVANTAGES

### 10.1 Advantages

The application is fast and offers great accuracy in predicting the flight delay.

Less maintenance is required.

It is user friendly.

It helps in reducing the tension of the passengers in knowing how long they will have to wait and lets passengers plan their schedule accordingly, thus in a way saving their time.

### 10.2 Disadvantages

It requires an internet connection for the website to work.

## 11 CONCLUSION

Flight delays not only anger and disturb air travelers' plans, but they also reduce efficiency, raise capital costs, reallocate flight crews and aircraft, and add to crew costs.

The goal of the flight delay prediction model is to forecast aircraft delays caused as a lot of passengers have become dependent on flights these days for their mode of transportation. The dataset which has all important information about the flights and its delay is made use for developing the model. A lot of steps are performed right from importing the data, then pre-processing it till training and testing the model. First the necessary packages were imported then the missing values in the data were handled and it was checked for outliers and then one hot encoding was performed and scaling was done. Then the data was split and given for training.

Four different models were used for training and out of it the best one was chosen based on the performance metric which is the Logistic regression model. Once the model was built it was integrated along with the Flask framework so that the users can enter their flight details and see if the flight would be on time or get delayed. Then this model is trained and deployed in the IBM Cloud.

As a result, anticipating delays can enhance airline operations and passenger satisfaction, which will benefit the economy and bring a positive impact.

## 12 FUTURE SCOPE

In the future, the application can be included with an user authentication model. Apart from checking if the flight would get delayed or not, their search history can be maintained and personalized flight recommendations can be done. A section where the users can give their feedback can also be implemented.

**APPENDIX**

**13.1 Source code:**

***13.1.1.1Flask file:***

### 13.1.1.1.1 app.py

from flask import Flask, render\_template, request

import pandas as pd

import joblib

import numpy as np

app = Flask(\_\_name\_\_)

@app.route('/')

def home():

return render\_template('Flightdelay.html')

@app.route('/result', methods = ['POST'])

def predict():

fl\_num = int(request.form.get('fno'))

month = int(request.form.get('month'))

dayofmonth = int(request.form.get('daym'))

dayofweek = int(request.form.get('dayw'))

sdeptime = request.form.get('sdt')

adeptime = request.form.get('adt')

arrtime = int(request.form.get('sat'))

depdelay = int(adeptime) - int(sdeptime)

inputs = list()

inputs.append(fl\_num)

inputs.append(month)

inputs.append(dayofmonth)

inputs.append(dayofweek)

if (depdelay < 15):

inputs.append(0)

else:

inputs.append(1)

inputs.append(arrtime)

origin = str(request.form.get("org"))

dest = str(request.form.get("dest"))

if(origin=="ATL"):

a=[1,0,0,0,0]

inputs.extend(a)

elif(origin=="DTW"):

a=[0,1,0,0,0]

inputs.extend(a)

elif(origin=="JFK"):

a=[0,0,1,0,0]

inputs.extend(a)

elif(origin=="MSP"):

a=[0,0,0,1,0]

inputs.extend(a)

elif(origin=="SEA"):

a=[0,0,0,0,1]

inputs.extend(a)

if(dest=="ATL"):

b=[1,0,0,0,0]

inputs.extend(b)

elif(dest=="DTW"):

b=[0,1,0,0,0]

inputs.extend(b)

elif(dest=="JFK"):

b=[0,0,1,0,0]

inputs.extend(b)

elif(dest=="MSP"):

b=[0,0,0,1,0]

inputs.extend(b)

elif(dest=="SEA"):

b=[0,0,0,0,1]

inputs.extend(b)

prediction = preprocessAndPredict(inputs)

#Pass prediction to prediction template

print(inputs)

return render\_template('/result.html', prediction = prediction)

def preprocessAndPredict(inputs):

test\_data = np.array(inputs).reshape((1,16))

model=open('E:\\NM Smartinternz\\Flight Delay\\flight.pkl','rb')

trained\_model = joblib.load(model)

df = pd.DataFrame(data=test\_data[0:, 0:], columns=['FL\_NUM', 'MONTH', 'DAY\_OF\_MONTH', 'DAY\_OF\_WEEK', 'DEP\_DEL15', 'CRS\_ARR\_TIME', 'ORIGIN\_ATL', 'ORIGIN\_DTW', 'ORIGIN\_JFK', 'ORIGIN\_MSP', 'ORIGIN\_SEA', 'DEST\_ATL', 'DEST\_DTW', 'DEST\_JFK', 'DEST\_MSP', 'DEST\_SEA'])

data = df.values

result = trained\_model.predict(data)

print(result)

return result

if \_\_name\_\_ == '\_\_main\_\_':

app.run(debug=True)

***13.1.1.2HTML JS and CSS files:***

#### 13.1.1.2.1 Flightdelay.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<link rel="stylesheet" href="{{ url\_for('static',filename='styles/styles.css') }}">

<script src="{{url\_for('static', filename='styles/delaypredict.js')}}"></script>

<title>Flight Delay Prediction</title>

</head>

<body id="flight-form">

<h2 id="main-head" class="centered-head">FLIGHT DELAY PREDICTION</h2>

<img sr c="{{url\_for('static', filename='styles/images/Flight.jpg')}}" id="bgimg">

<form name="flightForm" action="/result" method="POST" target="\_blank">

<div id="form-content">

<div id="block1">

<div class="detail-container">

<label for="fno" class="label-item">Enter the Flight Number</label>

<br>

<input type="number" id="fno" name="fno" class="text-input">

</div>

<div class="detail-container">

<label for="month" class="label-item">Month</label>

<br>

<input type="number" id="month" name="month" class="text-input" onblur="checkValid('month');" placeholder="Enter the Month Number">

<div class="alert-text" id="month-valid">Enter a valid month between 1 to 12.</div>

</div>

<div class="detail-container">

<label for="daym" class="label-item">Day of Month</label>

<br>

<input type="number" id="daym" name="daym" class="text-input" onblur="checkValid('daym');">

<div class="alert-text" id="daym-valid">Enter a valid day of month.</div>

</div>

<div class="detail-container">

<label for="dayw" class="label-item">Day of Week</label>

<br>

<input type="number" id="dayw" name="dayw" class="text-input" onblur="checkValid('dayw');">

<div class="alert-text" id="dayw-valid">Enter a valid day between 1 to 7.</div>

</div>

<div class="detail-container">

<label for="org" class="label-item">Origin</label>

<br>

<select id="org" name="org" class="select-input">

<option value="ATL" class="option-item">ATL</option>

<option value="SEA" class="option-item">SEA</option>

<option value="DTW" class="option-item">DTW</option>

<option value="MSP" class="option-item">MSP</option>

<option value="JFK" class="option-item">JFK</option>

</select>

</div>

<div class="detail-container">

<label for="dest" class="label-item">Destination</label>

<br>

<select id="dest" name="dest" class="select-input" onblur="checkValid('dest');">

<option value="ATL" class="option-item">ATL</option>

<option value="SEA" class="option-item">SEA</option>

<option value="DTW" class="option-item">DTW</option>

<option value="MSP" class="option-item">MSP</option>

<option value="JFK" class="option-item">JFK</option>

</select>

<div class="alert-text" id="dest-valid">Enter different Origin and Destination.</div>

</div>

<div class="detail-container">

<label for="sdt" class="label-item">Scheduled Departure Time</label>

<br>

<input type="number" id="sdt" name="sdt" class="text-input" onblur="checkValid('sdt');" placeholder="Enter in the format HHMM">

<div class="alert-text" id="sdt-valid">Enter a valid time between 500 to 2359.</div>

</div>

<div class="detail-container">

<label for="sat" class="label-item">Scheduled Arrival Time</label>

<br>

<input type="number" id="sat" name="sat" class="text-input" onblur="checkValid('sat');" placeholder="Enter in the format HHMM">

<div class="alert-text" id="sat-valid">Enter a valid time between 500 to 2359.</div>

</div>

<div class="detail-container">

<label for="adt" class="label-item">Actual Departure Time</label>

<br>

<input type="number" id="adt" name="adt" class="text-input" onblur="checkValid('adt');" placeholder="Enter in the format HHMM">

<div class="alert-text" id="adt-valid">Enter a valid time between 500 to 2359.</div>

</div>

</div></div>

<div id="submit-button">

<input type="submit" value="Submit" id="submit" class="button" onclick="validateForm()">

</div>

</form>

</body>

</html>

### 13.1.1.2.2 styles.css

body {

font-family: Arial, Helvetica, sans-serif;

margin: 0;

}

.content {

padding: 10px;

display: block;

}

.content-head {

text-align: center;

font-weight: bold;

font-size: 36px;

}

.button {

background-color: #1C55A2;

color: aliceblue;

padding: 10px;

border-radius: 10px;

border-color: #0E0E0F;

border-width: 1.5px;

}

.button a {

color: aliceblue;

text-decoration: none;

font-weight: bold;

}

#feedback-button {

margin-top: 10px;

}

#feedback-button-section {

text-align: center;

}

#bgimg {

position: fixed;

z-index: -1;

width: 100%;

height: 100%;

padding: 0;

margin: 0;

top: 0;

opacity: 0.9;

}

.centered-head {

text-align: center;

color: #3700ff;

font-weight: bold;

}

.label-item {

color: #f4f6f8;

font-weight: bold;

}

.detail-container {

padding-bottom: 10px;

padding-top: 10px;

}

.text-input {

margin-top: 5px;

border-color: #1C55A2;

border-width: 1.5px;

width: 75%;

height: 20px;

padding-left: 5px;

padding-right: 5px;

padding-top: 2px;

padding-bottom: 2px;

}

.select-input {

margin-top: 5px;

border-color: #0E0E0F;

border-width: 1.5px;

width: 40%;

height: 30px;

background-color: #1C55A2;

color: aliceblue;

font-weight: bold;

cursor: pointer;

}

#form-content {

justify-content: space-evenly;

flex-direction: row;

}

#block1 {

display: block;

width: 40%;

padding: 20px;

}

#review {

height: 100px;

padding-top: 5px;

font-family: Arial, Helvetica, sans-serif;

}

#submit-button {

padding-left: 20px;

}

#submit {

background-color: #00c3ff;

color: rgb(1, 7, 14);

font-weight: bold;

}

#submit:hover {

cursor: pointer;

}

.choose-item {

font-weight: 600;

}

input[type="radio"], input[type="checkbox"] {

cursor: pointer;

}

.alert-text {

color: rgb(255, 79, 47);

font-size: small;

padding-left: 10px;

display: none;

}

#### 13.1.1.2.3 delaypredict.js

function validateForm() {

var fno = document.forms["flightForm"]["fno"].value;

var month = document.forms["flightForm"]["month"].value;

var daym = document.forms["flightForm"]["daym"].value;

var dayw = document.forms["flightForm"]["dayw"].value;

var org = document.forms["flightForm"]["org"].value;

var dest = document.forms["flightForm"]["dest"].value;

var sdt = document.forms["flightForm"]["sdt"].value;

var sat = document.forms["flightForm"]["sat"].value;

var adt = document.forms["flightForm"]["adt"].value;

if (fno == "" || fno == null || month == "" || month == null || daym == "" || daym == null || dayw == "" || dayw == null || org == "" || org == null || dest == "" || dest == null || sdt == "" || sdt == null || sat == "" || sat == null || adt == "" || adt == null) {

alert("The given fields must be filled out");

event.preventDefault();

}

if(month<1 || month>12)

{

alert("Enter a valid month");

event.preventDefault();

}

if(month==2)

{

if(daym<1 || daym>=29)

{

alert("Enter a valid day of month");

event.preventDefault();

}

}

else if(month==1 || month==3 || month==5 || month==7 || month==8 || month==10 || month==12)

{

if(daym<1 || daym>31)

{

alert("Enter a valid day of month");

event.preventDefault();

}

}

else if(month==4 || month==6 || month==9 || month==11)

{

if(daym<1 || daym>30)

{

alert("Enter a valid day of month");

event.preventDefault();

}

}

if(dayw<1 || dayw>7)

{

alert("Enter a valid day of week");

event.preventDefault();

}

if(org==dest)

{

alert("Enter different origin and destination");

event.preventDefault();

}

if(sdt<500 || sdt>2400)

{

alert("Enter a valid Departure time between 500 to 2400");

event.preventDefault();

}

if(sat<500 || sat>2400)

{

alert("Enter a valid Arrival time between 500 to 2400");

event.preventDefault();

}

if(sdt==sat)

{

alert("Departure and Arrival time must differ by atleast 1 hr");

event.preventDefault();

}

if(adt<500 || adt>2400)

{

alert("Enter a valid Departure time between 500 to 2400");

event.preventDefault();

}

}

function checkValid(element)

{

var obj = document.getElementById(element);

var valid\_obj = document.getElementById(element + "-valid");

if(element=='month')

{

if(obj.value<1 || obj.value>12)

{

obj.style.borderColor = "rgb(255, 79, 47)"

valid\_obj.style.display = "block";

}

else {

obj.style.borderColor = "#1C55A2";

valid\_obj.style.display = "none";

}

}

if(element=='daym')

{

var monobj = document.getElementById('month');

if(monobj.value==2)

{

if(obj.value<1 || obj.value>=29)

{

obj.style.borderColor = "rgb(255, 79, 47)"

valid\_obj.style.display = "block";

}

else {

obj.style.borderColor = "#1C55A2";

valid\_obj.style.display = "none";

}

}

else if(monobj.value==1 || monobj.value==3 || monobj.value==5 || monobj.value==7 || monobj.value==8 || monobj.value==10 || monobj.value==12)

{

if(obj.value<1 || obj.value>31)

{

obj.style.borderColor = "rgb(255, 79, 47)"

valid\_obj.style.display = "block";

}

else {

obj.style.borderColor = "#1C55A2";

valid\_obj.style.display = "none";

}

}

else if(monobj.value==4 || monobj.value==6 || monobj.value==9 || monobj.value==11)

{

if(obj.value<1 || obj.value>30)

{

obj.style.borderColor = "rgb(255, 79, 47)"

valid\_obj.style.display = "block";

}

else {

obj.style.borderColor = "#1C55A2";

valid\_obj.style.display = "none";

}

}

else

{

obj.style.borderColor = "rgb(255, 79, 47)"

valid\_obj.style.display = "block";

}

}

if(element=='dayw')

{

if(obj.value<1 || obj.value>7)

{

obj.style.borderColor = "rgb(255, 79, 47)"

valid\_obj.style.display = "block";

}

else {

obj.style.borderColor = "#1C55A2";

valid\_obj.style.display = "none";

}

}

if(element=='dest')

{

var origin\_obj = document.getElementById('org');

if(obj.value==origin\_obj.value)

{

obj.style.borderColor = "rgb(255, 79, 47)"

valid\_obj.style.display = "block";

}

else

{

obj.style.borderColor = "#1C55A2";

valid\_obj.style.display = "none";

}

}

if(element=='sdt')

{

if(obj.value<500 || obj.value>2400)

{

obj.style.borderColor = "rgb(255, 79, 47)"

valid\_obj.style.display = "block";

}

else {

obj.style.borderColor = "#1C55A2";

valid\_obj.style.display = "none";

}

}

if(element=='sat')

{

if(obj.value<500 || obj.value>2400)

{

obj.style.borderColor = "rgb(255, 79, 47)"

valid\_obj.style.display = "block";

}

else {

obj.style.borderColor = "#1C55A2";

valid\_obj.style.display = "none";

}

}

if(element=='adt')

{

if(obj.value<500 || obj.value>2400)

{

obj.style.borderColor = "rgb(255, 79, 47)"

valid\_obj.style.display = "block";

}

else {

obj.style.borderColor = "#1C55A2";

valid\_obj.style.display = "none";

}

}

}

#### 13.1.1.2.4 result.html

<!doctype html>

<html>

<head>

<title>Flight Delay Prediction-Result</title>

<link rel="stylesheet" href="{{ url\_for('static',filename='styles/result\_styles.css') }}">

</head>

<body>

<img src="{{url\_for('static', filename='styles/images/Flight.jpg')}}" id="bgimg">

{% if prediction == 0.0 %}

<div class="pred\_result" id="result\_0">Your flight will likely be on time</div>

{% endif %}

{% if prediction == 1.0 %}

<div class="pred\_result" id="result\_1">Your flight is likely to be delayed</div>

{% endif %}

</body>

</html>

### 13.2 GitHub link & Project Demo Link

**GitHub link:**

**Project Demo Link:**