

**PROJECT REPORT**  
**UNIVERSITY ADMIT ELIGIBILITY PREDICTOR**  
**TEAM ID:PNT2022TMID37049**

**TABLE OF CONTENTS**

**1. INTRODUCTION**

- 1.1 Project Overview
- 1.2 Purpose

**2. LITERATURE SURVEY**

- 2.1 Existing System
- 2.2 References
- 2.3 Problem Statement

**3. IDEATION & PROPOSED SOLUTION**

- 3.1 Empathy Map Canvas
- 3.2 Ideation & Brainstorming
- 3.3 Proposed Solution
- 3.4 Problem Solution fit

**4. REQUIREMENT ANALYSIS**

- 4.1 Functional Requirements
- 4.2 Non-Functional Requirements

**5. PROJECT DESIGN**

- 5.1 Data Flow Diagram
- 5.2 Solution & Technical Architecture

**6. PROJECT PLANNING & SCHEDULING**

- 6.1 Sprint Planning & Estimation
- 6.2 Sprint Delivery Schedule
- 6.3 Reports from JIRA

## **7. CODING & SOLUTIONING (Explain the features added in the project along with code)**

7.1 Feature 1

7.2 Feature 2

## **8. TESTING**

8.1 Test Cases

8.2 User Acceptance Testing

## **9. RESULTS**

9.1 Performance Metrics

## **10.ADVANTAGES AND DISADVANTAGES**

## **11.CONCLUSION**

## **12.FUTURE SCOPE**

## **13.APPENDIX**

Source Code

Github & Project Demo Link

# **1.Introduction**

## **1.1 Project Overview**

Students are often worried about their chances of admission to University. The aim of this project is to help students in shortlisting universities with their profiles. The predicted output gives them a fair idea about their admission chances in a particular university. This analysis should also help students who are currently preparing or will be preparing to get a better idea.

## **1.2 Purpose**

there are many number of students who want to pursue higher education after engineering or any graduate degree many students want to do master's in abroad Universities. we are focusing on only the students who want to pursue their higher education in abroad universities. Students who want to do masters in Abroad have to write GRE (Graduate Records Examination) and TOEFL (Test of English as a Foreign Language/International English Language) Once they have attended the exams they have to prepare their SOP(statement of purpose) and LOR(letter of reccomendation) which are one of the crucial factors they have to consider. students want to know eligibility chances based on their marks our website will definitely help students to know and Predict the chance Students required to fill the details like GRE, TOEFL, SOP,LOR According to the students marks detail. students may get know to which type of University rating they have chance to get admit.

## **2.LITERATURE SURVEY**

### **2.1 EXISTING PROBLEM**

Students want to know about their chances of getting admission in Universities based on their score on General Record Examination (GRE), command over the English language is judged based on their score in English competency test like Test Of English as a Foreign Language (TOEFL) OR International English Language Testing System (IELTS), their work experience in same or other fields, the quality of the Letters Of Recommendation (LOR) and the Statement Of Purpose documents provided by the student etc. for the prediction process sometimes it seems the data might be invalid if the data entered incorrectly then the prediction process get error then the user's can't able to understand this problem may occurs is the system model is not properly designed and trained well so the user's faces the problem of prediction

### **2.2 REFERENCES**

1. Borah M.D., Application of knowledge based decision technique to predict student enrolment decision. Recent Trends in Information Systems (Re TIS), 21-23 Dec. 2011,180- 184.
2. Ragab, A.H.M., Hybrid recommender system for predicting college admission, Intelligent Systems Design and Applications (ISDA).107-113.
3. Rensong Dong. The module of prediction of College Entrance Examination aspiration, Fuzzy Systems and Knowledge Discovery (FSKD),1559-1562.
4. E. Gottlieb, "Using integer programming to guide college admissions decisions: a preliminary report", Journal of Computing Sciences in Colleges, Volume 17. Issue 2, Pages: 271-279, 2001.
5. J.S. Moore, "An expert system approach to graduate school admission decisions and academic performance prediction". ScienceDirect Volume 26, Issue 5, October 1998, Pages 659-670.
6. L. Chang. Applying Data Mining to Predict College Admissions Yield, Chapter 4 in J. Luan and C. Zhao (Eds.), Data mining in action: Case studies, Spring 2008 - College of Education.
7. S. Nadi, M.H. Sarace, and A. Bagheri," Hybrid Recommender System for Dynamic Web Users", International Journal Multimedia and Image Processing (UMIP), Vol. 1, Issue 1, March 2011.
8. J. A. Freeman, and D. M. Skapura, "Neural Networks: Algorithms. Applications. And Programming". Addison Wesley Pub (Sd), June 1991.

9. S. Vinnik, and H. Marc, "Decision Support System for Managing Educational Capacity Utilization in Universities", Int. Conf. on Engineering and Computer Education, ICECE05.

10. W. C. Lou, "A Hybrid Model of Tree Classifier and Neural Network for University Admission Recommender System," Master of Science Thesis, University of Macau, Faculty of Science and Technology, 2008.

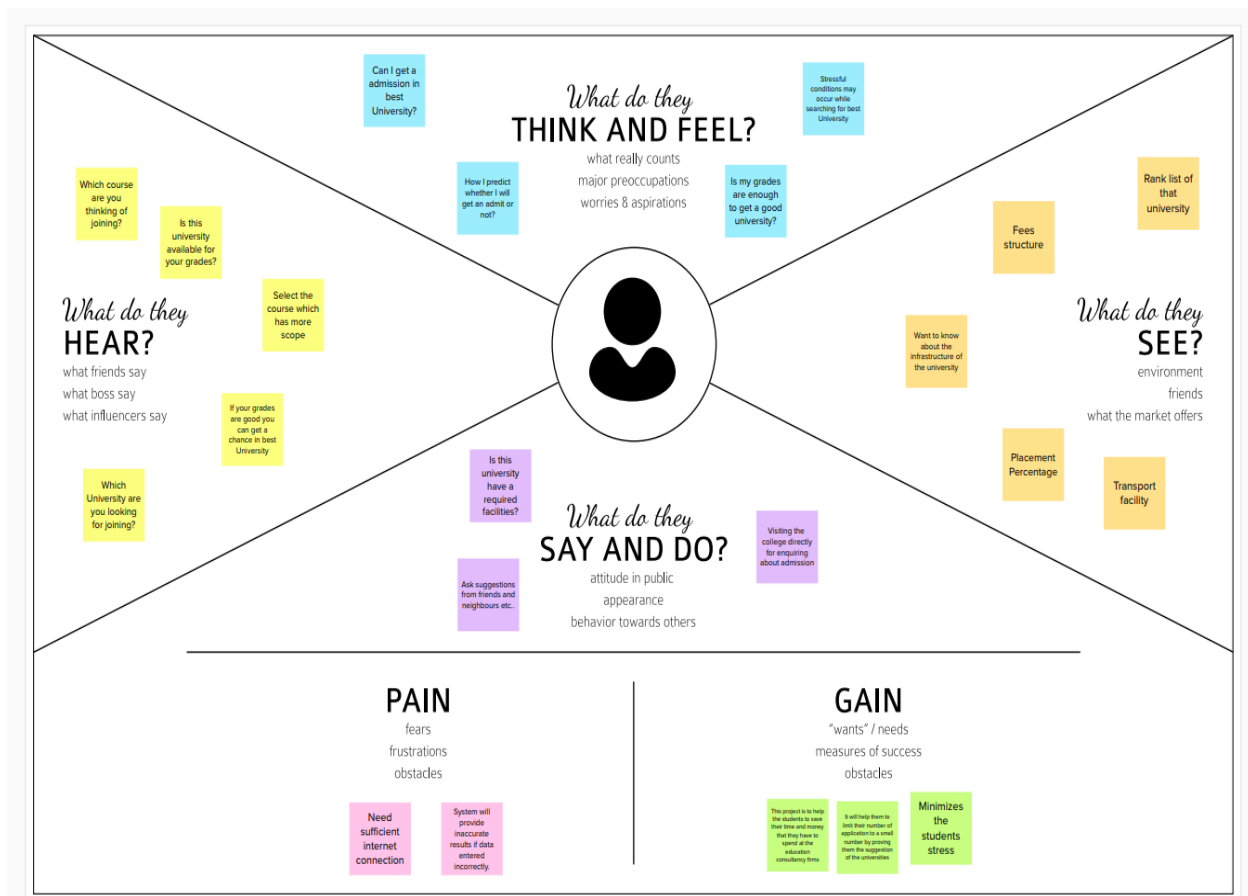
## **2.3 PROBLEM STATEMENT DEFINITION**

I am a Student. I have to get a Admission in University but I don't know is this my marks are capable for get a what rating type of University. For get a admission in university I came to know whether my marks are eligible to get a what type of rating University. Confusion state occurs while knowing about my prediction chances. This system makes me to help and Predict the chance or no chance based on my marks.

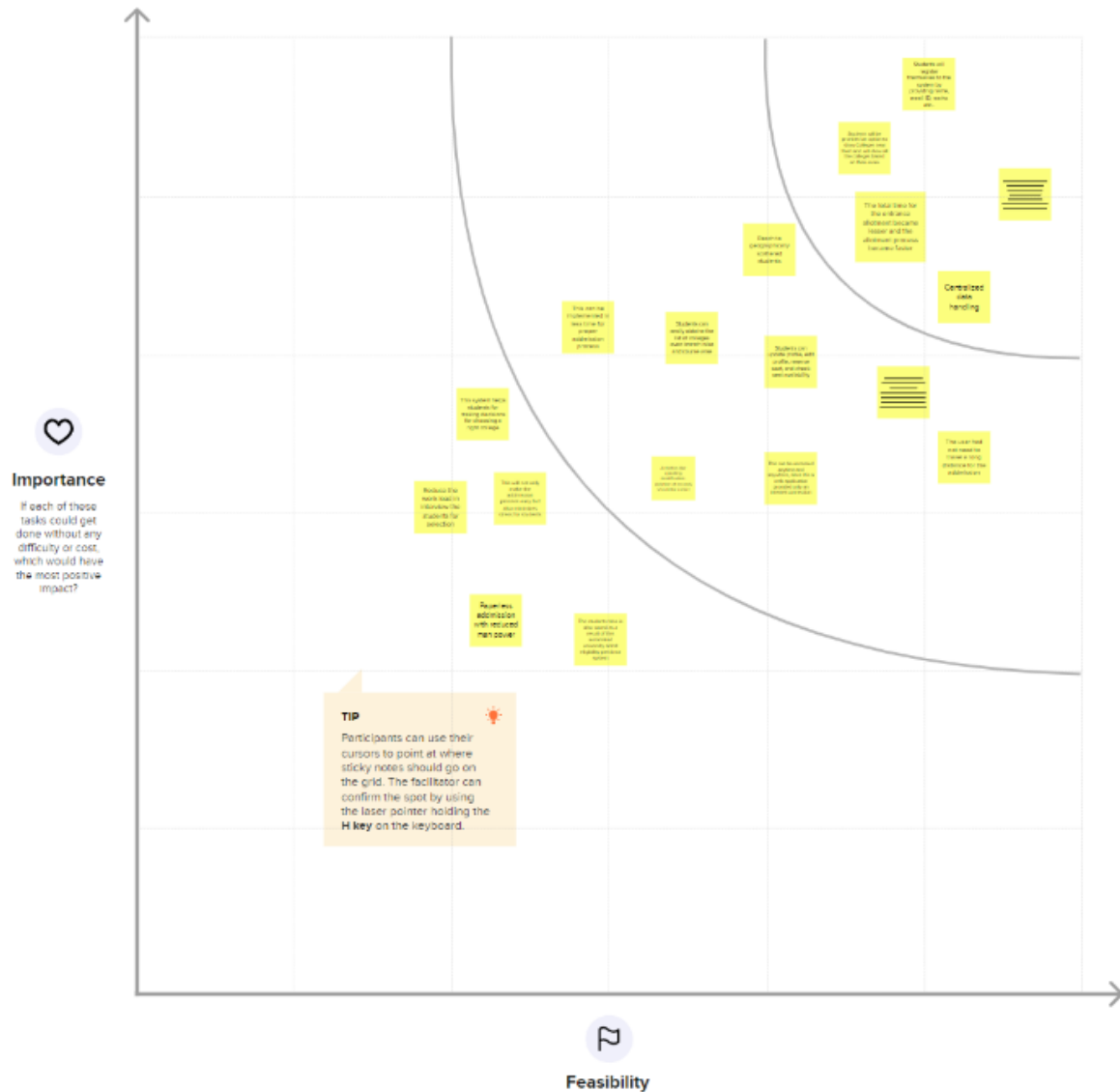
## 3.IDEATION &PROPOSED SOLUTION

### 3.1 EMPATHY MAP CANVAS

An empathy map is a simple, easy-to-digest visual that captures knowledge about a user's behaviours. It is a useful tool to help teams better understand their users. Creating an effective solution requires understanding the true problem and the person who is experiencing it. The exercise of creating the map helps participants consider things from the user's perspective along with his or her goals and challenges.



## 3.2 IDEATION AND BRAINSTORMING



### **3.3 PROPOSED SOLUTION**

For the proposed system, the machine learning algorithm gradient boosting was used to classify the dataset for the application model. The prediction functionality is implemented in the back-end process using the Flask framework and machine learning algorithms are integrated in the front-end and back-end Flask framework. This well structured application helps to predict correct and accurate data, this model avoids error, imprecise data, missing values, this application is user friendly and easy to use

### **3.4 PROBLEM SOLUTION FIT**

#### **1.CUSTOMER SEGMENTS**

The customer of this project will be students who have completed their graduate of any degree and aspire to get admitted into the abroad universities

#### **2.JOBS-TO-BE-DONE/PROBLEMS**

Students can register their marks details for predicting the admission in university Data collection is probably the most important step in designing the predictor hence it must be ensured that it is done properly Need sufficient internet connection System will provide inaccurate results if data entered incorrectly.

#### **3.TRIGGERS**

- The customer can be accessed anytime and anywhere by using this web application
- The prediction process is fair and easy to understandable

#### **4.EMOTIONS BEFORE/AFTER**

- This application predicts the best choices for the user, it makes the user confident.
- Trust and worthy application to use

#### **5.AVAILABLE SOLUTIONS**

Centralized data handling. It can handle the details of students such as mark details, This students database has been designed taking into account the practical needs to manage a student data.



## **6. CUSTOMER CONSTRAINTS**

The constraints which the customer would face may be the fear of data misuse Lack of network connection and guidance

## **7. BEHAVIOUR**

- Students will fill details themselves to the system by using their scores.
- Activities like updating,modification, deletion of records should be easier.

## **8.CHANNELS OF BEHAVIOUR**

- The customer can Predict theri marks by using the web application.
- it can be accessed by anytime and anywhere.

## **9.PROBLEM ROOT CAUSE**

The reliability of the predictor might be affected if the collected data is found to be inaccurate are considered to judge the Predicting Secondly, customers might stop from using our website they find it to be prone to cyber attacks.

## **10.YOUR SOLUTION**

Design a predictor with the help of the data collected, and ensure that it is accurate. Also make sure that the data collected from the users is safe and secure.

## 4. REQUIREMENT ANALYSIS

### 4.1 Functional Requirements

FR NO.	Functional Requirements(Epic)	Sub Requirement (Story/Sub-Task)
FR-1	Users Data Collection	Collecting the TOEFL, GRE scores from the user.
FR-2	User Registration	Registration through Form Registration through Gmail
FR-3	Predicting the Data	Analysing the given data with the previous year cut-off of the universities and then system provides the list of universities based on the student cut-off.
FR-4	Users Preference	Users can select the universities based on their convenient and preference from the predicted list.
FR-5	Output	The Universities are listed based on the Student marks where the universities will be listed in the rankwise, So the predicted output gives them a fair idea about their admission chances in a particular university.

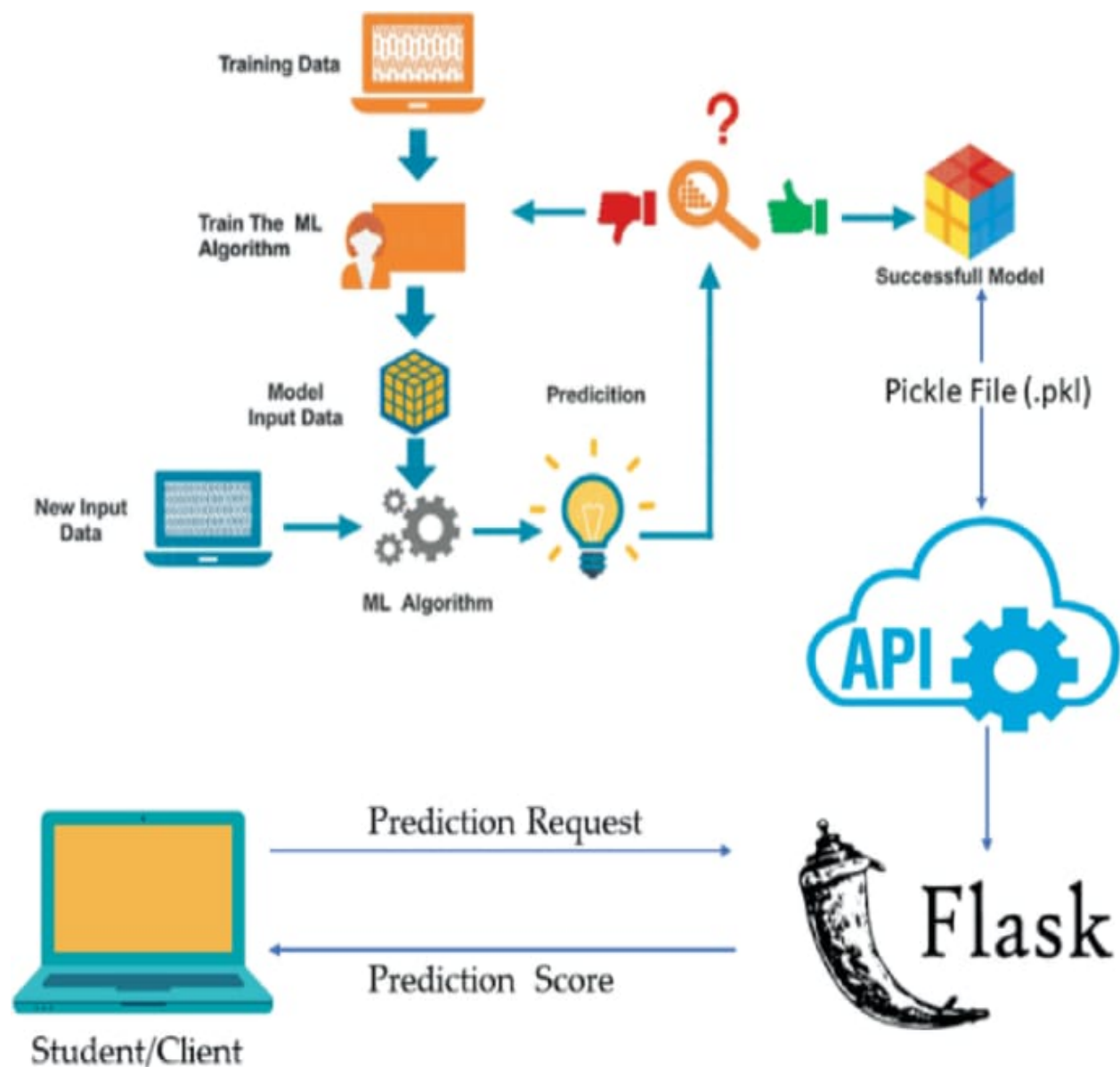
## 4.2 NON FUNCTIONAL REQUIREMENTS

FR No.	Non-Functional Requirement	Description
NFR-1	Usability	The predictor platform should provide the capacity to perform the right options for the users based on their profiles.
NFR-2	Security	The student profile and data should be maintained in a secured manner.
NFR-3	Reliability	<ul style="list-style-type: none"><li>• The user can find universities based on their preferred locations and results.</li><li>• The predictor system should be consistent and the system will give accurate and reliable resource.</li></ul>
NFR-4	Performance	The system can supply any number of users at a time and provides the list of universities, the predictor platform gives the good performance criteria.
NFR-5	Availability	The system predictor will available to users to accessed anytime and anywhere whenever they required.
NFR-6	Scalability	The system must be scalable to support many users at a time.

## 5.PROJECT DESIGN

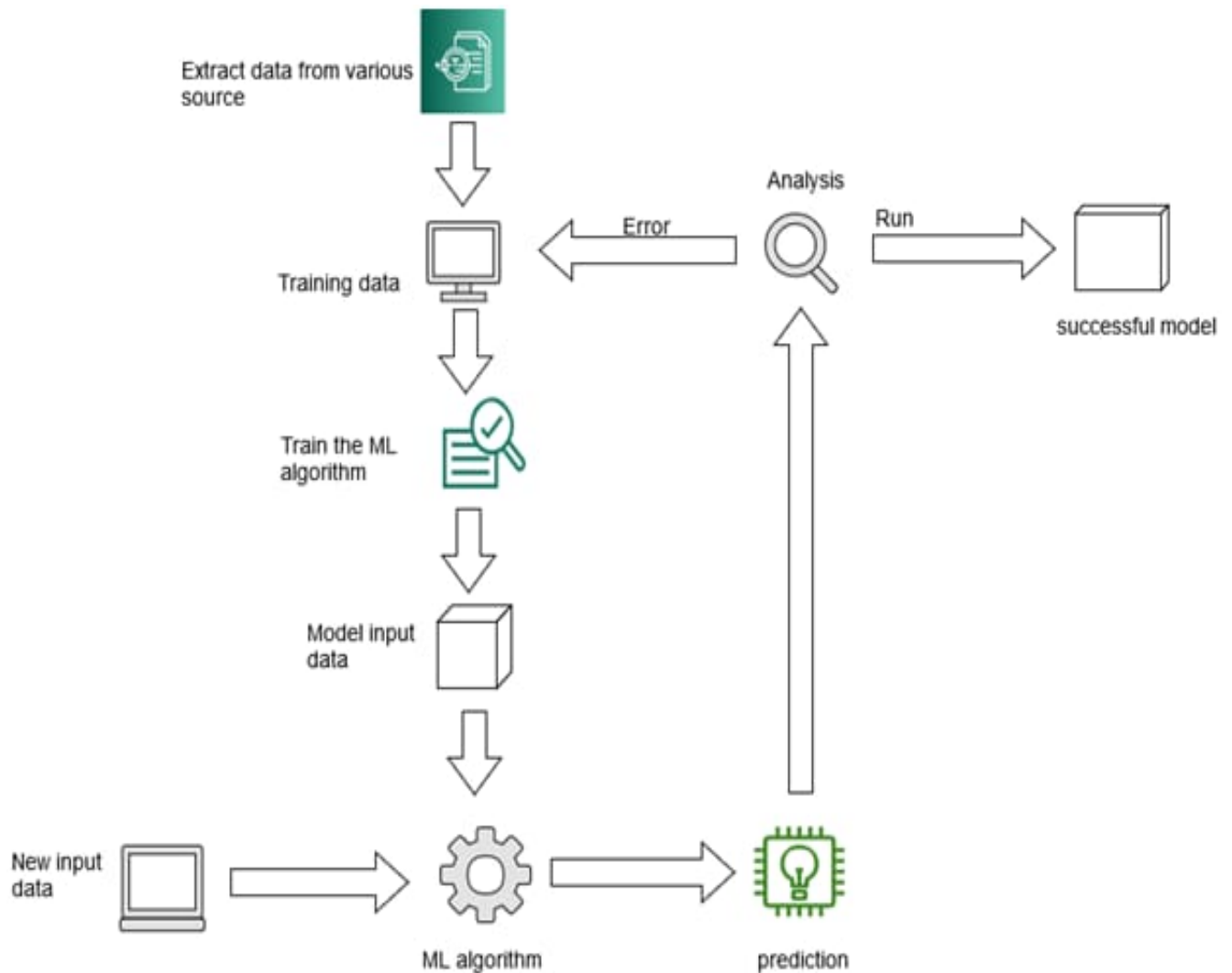
### 5.1 DATA FLOW DIAGRAM

A data-flow diagram is a way of representing a flow of data through a process or a system. The DFD also provides information about the outputs and inputs of each entity and the process itself. A data-flow diagram has no control flow-there are no decision rules and no loops.



## 5.2 SOLUTION AND TECHNICAL ARCHITECTURE

Based on the complexity of the deployment, a solution architecture diagram may actually be a set of diagrams documenting various levels of the architecture. The diagram relates the information that you gather on the environment to both physical and logical choices for your architecture in an easily understood manner.



## 5.3 USER STORIES

- **Customer**

As a user, I can register into the application by setting password and I login through the application by using email id and password after login into the application I entered my mark details finally I click into Predict I came to know whether this scores are eligibility to the what rating type of University.

- **Server**

Performing Data Analysis, Choosing the perfect Algorithm/model(ML),Checking Error Matrix.The backend process is implemented by the flask framework and machine learning algorithms Integrated frontend and backend by using flask framework.

## 6.PROJECT PLANNING AND SCHEDULING

### 6.1 SPRINT PLANNING AND SCHEDULING

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	University Registration	USN-1	As a student, I can register for the application by entering my email, password by confirming my password.	2	High	2
Sprint-1		USN-2	As a student, I will receive confirmation email once I have registered for the application from the university.	1	High	1
Sprint-2		USN-3	As a student, I can register for the application through university by uploading my mark statements. Upload original copy of the Marksheets.	2	Low	2
Sprint-3		USN-4	As a Students, I can register for the application through Gmail with all eligibility. Students can upload extra course completion certificates.	2	Medium	2
Sprint-4	Login by user name	USN-5	As a Student, I can log into the application by entering email & password	1	High	2
	Dashboard		Check dashboard and upload the details according to university criteria.			4

## 6.2 Sprint Delivery Schedule

### Project Tracker, Velocity & Burndown Chart

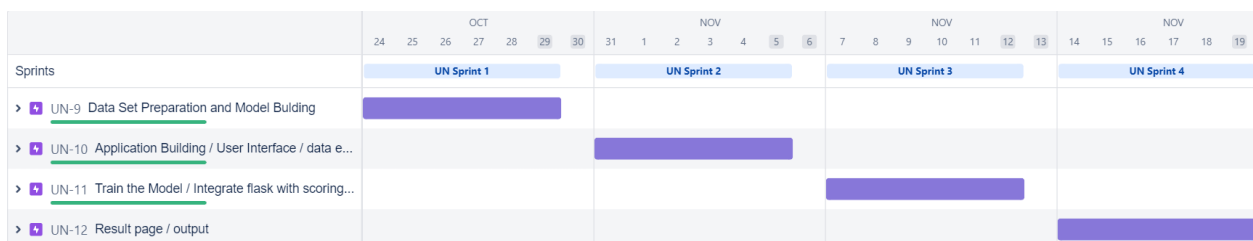
Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

#### Velocity:

Imagine we have a 6-day sprint duration, and the velocity of the team is 20 (points per sprint). Let's calculate the team's average velocity (AV) per iteration unit (story points per day)

$$AV = \text{SPRINT DURATION} / \text{VELOCITY} = 20/6 = 3.33$$

## 6.3 REPORTS FROM JIRA



### BURNDOWN CHART



## 7.CODING AND SOLUTIONING

### 7.1 FEATURE 1

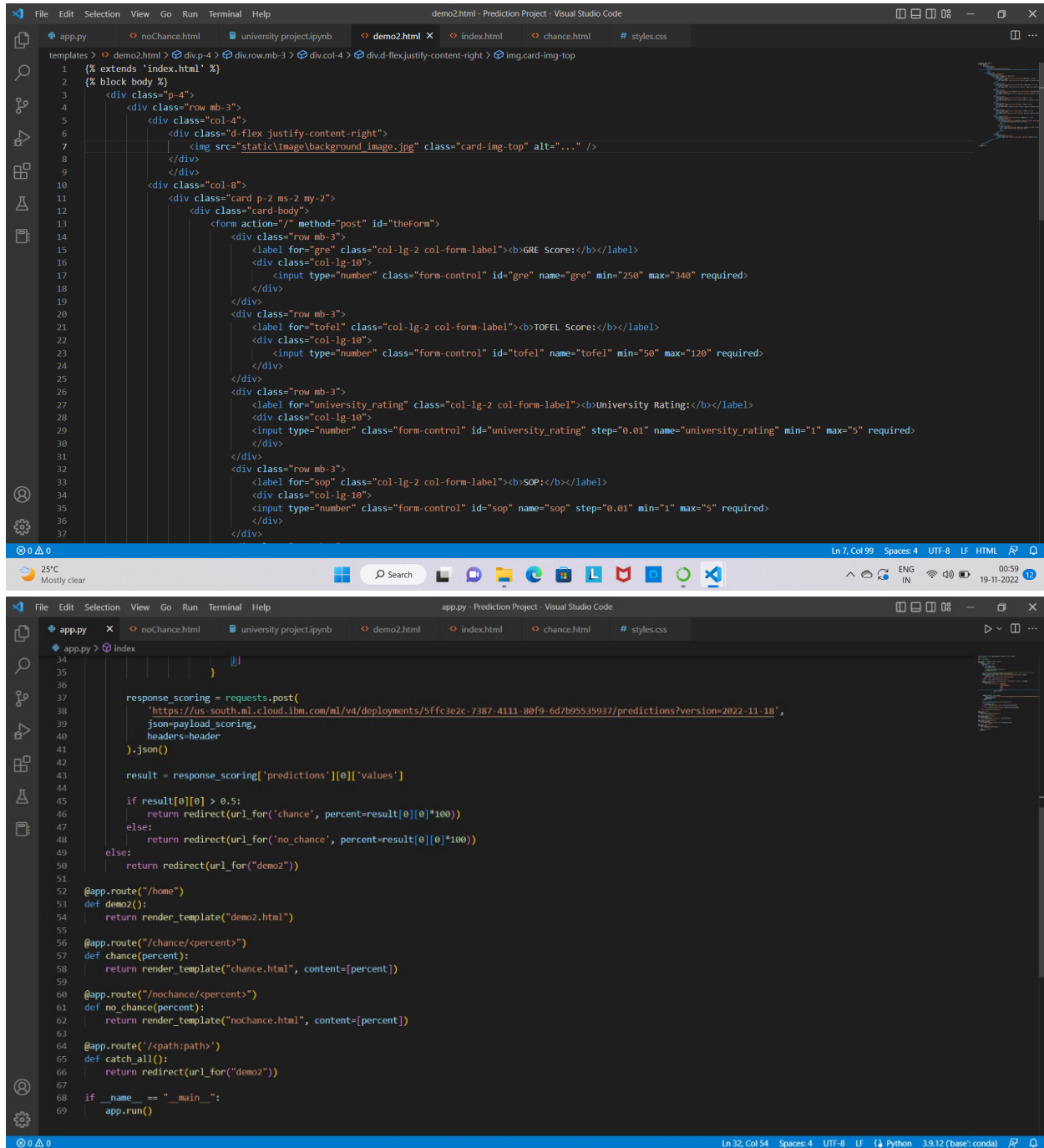
The following is the flask app coding

```
app.py - Prediction Project - Visual Studio Code
File Edit Selection View Go Run Terminal Help
app.py x noChance.html university project.ipynb demo2.html index.html chance.html # styles.css
app.py > index
1 from flask import Flask, render_template, redirect, url_for, request
2 import requests
3
4 app = Flask(__name__)
5
6 @app.route("/", methods = ['POST', 'GET'])
7 def index():
8     if request.method == 'POST':
9         arr = []
10        for i in request.form:
11            val = request.form[i]
12            if val == '':
13                return redirect(url_for("demo2"))
14            arr.append(float(val))
15
16        # you must manually set API_KEY below using information retrieved from your IBM Cloud account.
17        API_KEY = "3gc4tXNG1UINbp9Rps1_k6WRVw7uLARTtNwJ0UTTxMn"
18        token_response = requests.post('https://iam.cloud.ibm.com/identity/token', data={
19            'apikey':API_KEY,
20            'grant_type': 'urn:ibm:params:oauth:grant-type:apikey'
21        })
22        mitoken = token_response.json()["access_token"]
23
24        header = {'Content-Type': 'application/json', 'Authorization': 'Bearer ' + mitoken}
25        payload_scoring = {
26            "input_data": [{"fields": [
27                'GRE Score',
28                'TOEFL Score',
29                'University Rating',
30                'SOP',
31                'LOR ',
32                'CGPA',
33                'Research'
34            ],
35            "values": [arr]
36        }
37
38        response_scoring = requests.post(
```

```
app.py - Prediction Project - Visual Studio Code
File Edit Selection View Go Run Terminal Help
app.py x noChance.html university project.ipynb demo2.html index.html chance.html # styles.css
app.py > index
34
35
36
37 response_scoring = requests.post(
38     'https://us-south.ml.cloud.ibm.com/v4/deployments/5ffc3e2c-7387-4111-88f9-6d7b95535937/predictions?version=2022-11-18',
39     json=payload_scoring,
40     headers=header
41 ).json()
42
43 result = response_scoring['predictions'][0]['values']
44
45 if result[0][0] > 0.5:
46     return redirect(url_for('chance', percent=result[0][0]*100))
47 else:
48     return redirect(url_for('no_chance', percent=result[0][0]*100))
49
50 return redirect(url_for("demo2"))
51
52 @app.route("/home")
53 def demo2():
54     return render_template("demo2.html")
55
56 @app.route("/chance/<percent>")
57 def chance(percent):
58     return render_template("chance.html", content=[percent])
59
60 @app.route("/nochance/<percent>")
61 def no_chance(percent):
62     return render_template("noChance.html", content=[percent])
63
64 @app.route('/<path:path>')
65 def catch_all():
66     return redirect(url_for("demo2"))
67
68 if __name__ == "__main__":
69     app.run()
```

## 7.2 FEATURE 2

The Following Code is the Demo2.Html



The image displays two screenshots of a Visual Studio Code editor window, showing the development of a web application. The top screenshot shows the HTML file 'demo2.html' with a form for predicting a score. The bottom screenshot shows the Python file 'app.py' with the corresponding Flask application logic.

**Top Screenshot: demo2.html**

```
1 {% extends 'index.html' %}
2 {% block body %}
3 <div class="row mb-3">
4 <div class="col-4">
5 <div class="d-flex justify-content-right">
6 
7 </div>
8 </div>
9 <div class="col-8">
10 <div class="card p-2 ms-2 my-2">
11 <div class="card-body">
12 <form action="/" method="post" id="theForm">
13 <div class="row mb-3">
14 <label for="gre" class="col-lg-2 col-form-label"><b>GRE Score:</b></label>
15 <div class="col-lg-10">
16 <input type="number" class="form-control" id="gre" name="gre" min="250" max="340" required>
17 </div>
18 </div>
19 <div class="row mb-3">
20 <label for="tofel" class="col-lg-2 col-form-label"><b>TOFEL Score:</b></label>
21 <div class="col-lg-10">
22 <input type="number" class="form-control" id="tofel" name="tofel" min="50" max="120" required>
23 </div>
24 </div>
25 <div class="row mb-3">
26 <label for="university_rating" class="col-lg-2 col-form-label"><b>University Rating:</b></label>
27 <div class="col-lg-10">
28 <input type="number" class="form-control" id="university_rating" step="0.01" name="university_rating" min="1" max="5" required>
29 </div>
30 </div>
31 <div class="row mb-3">
32 <label for="sop" class="col-lg-2 col-form-label"><b>SOP:</b></label>
33 <div class="col-lg-10">
34 <input type="number" class="form-control" id="sop" name="sop" step="0.01" min="1" max="5" required>
35 </div>
36 </div>
37 </div>
</div>
</div>
</div>
```

**Bottom Screenshot: app.py**

```
34 }
35 }
36 }
37 response_scoring = requests.post(
38     'https://us-south.ml.cloud.ibm.com/v4/deployments/5ffc3e2c-7387-4111-88f9-6d7b95535937/predictions?version=2022-11-18',
39     json=payload_scoring,
40     headers=header
41 ).json()
42 result = response_scoring['predictions'][0]['values']
43
44 if result[0][0] > 0.5:
45     return redirect(url_for('chance', percent=result[0][0]*100))
46 else:
47     return redirect(url_for('no_chance', percent=result[0][0]*100))
48
49 else:
50     return redirect(url_for("demo2"))
51
52 @app.route("/")
53 def demo2():
54     return render_template("demo2.html")
55
56 @app.route("/chance/<percent>")
57 def chance(percent):
58     return render_template("chance.html", content=[percent])
59
60 @app.route("/nochance/<percent>")
61 def no_chance(percent):
62     return render_template("noChance.html", content=[percent])
63
64 @app.route('/<path:path>')
65 def catch_all():
66     return redirect(url_for("demo2"))
67
68 if __name__ == "__main__":
69     app.run()
```

This Following Code is the Index.html

```
File Edit Selection View Go Run Terminal Help
demo2.html - Prediction Project - Visual Studio Code
app.py noChance.html university project.ipynb demo2.html X index.html chance.html styles.css
templates > demo2.html > div.p-4 > div.row.mb-3 > div.col-4 > div.d-flex.justify-content-right > img.card-img-top
41 <input type="number" class="form-control" id="lor" name="lor" step="0.01" min="1" max="5" required>
42 </div>
43 </div>
44 <div class="row mb-3">
45 <label for="cgpa" class="col-lg-2 col-form-label"><b>CGPA:</b></label>
46 <div class="col-lg-10">
47 <input type="number" class="form-control" id="cgpa" name="cgpa" step="0.01" min="5" max="10" required>
48 </div>
49 </div>
50 <fieldset class="row mb-3">
51 <legend class="col-form-label col-sm-2 pt-0"><b>Research:</b></legend>
52 <div class="col-sm-10">
53 <div class="form-check">
54 <input class="form-check-input" type="radio" name="yes_no_radio" id="gridRadios1" value="1">
55 <label class="form-check-label" for="yes_no_radio">
56 Yes
57 </label>
58 </div>
59 <div class="form-check">
60 <input class="form-check-input" type="radio" name="yes_no_radio" id="gridRadios2" value="0" checked>
61 <label class="form-check-label" for="yes_no_radio">
62 No
63 </label>
64 </div>
65 </div>
66 </fieldset>
67 <div class="row lg-3">
68 <div class="col-lg-2 mb-2 me-3">
69 <button type="submit" class="btn btn-primary" id="button">Predict</button>
70 </div>
71 </form>
72 </div>
73 </div>
74 </div>
75 </div>
76 </div>
77 {% endblock %}
```

This Following Code is the Chance.html

```
File Edit Selection View Go Run Terminal Help
index.html - Prediction Project - Visual Studio Code
app.py noChance.html university project.ipynb demo2.html index.html X chance.html styles.css
templates > index.html > html > body > nav.navbar.navbar-expand-lg.bg-light > div.container-fluid > a.navbar-brand.text-responsive-h > img.d-inline-block.align-text-top.
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <meta charset="UTF-8">
5 <meta http-equiv="X-UA-Compatible" content="IE=edge">
6 <meta name="viewport" content="width=device-width, initial-scale=1.0">
7 <meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1, user-scalable=no">
8 <link rel="stylesheet" type="text/css" rel="noopener" target="blank" href="static/css/styles.css">
9 <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-Zenh87qx5Jm2210Vw8CK2rdkQ2Bzep5IDxbcnCeu0Xj2
10 <script type="text/javascript" src="static/js/script.js" async></script>
11 <title>University Admit Eligibility Predictor</title>
12 </head>
13 <body>
14 <nav class="navbar navbar-expand-lg bg-light">
15 <div class="container-fluid">
16 <a class="navbar-brand text-responsive-h" href="/">
17 
18 <center><h3>University Admit Eligibility Predictor</h3></center>
19 </a>
20 </div>
21 </nav>
22 {% block body %}
23 <h1>Index Page </h1>
24 {% endblock %}
25 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/js/bootstrap.bundle.min.js" integrity="sha384-OERCA2EjJCMA+3y4gXIOqEjwXjY7qPCqsdltbnJuaoe923+mo//f6
26 </body>
27 </html>
28
```

The Following Code is the no chance.html

noChance.html - Prediction Project - Visual Studio Code

File Edit Selection View Go Run Terminal Help


app.py noChance.html X university project.ipynb demo2.html index.html chance.html styles.css

```

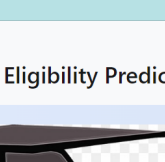
1 {% extends 'index.html' %}
2 {% block body %}
3
4 <div class="container text-center p-4">
5   <div class="d-flex justify-content-center">
6     <div class="card" style="width: 34rem;>
7       
9         <h5 class="card-title">You have a LOW / NO chance</h5>
10        <p class="card-text">The model has predicted that you have no chance</p>
11        <a href="/home" class="btn btn-primary">Go Back</a>
12      </div>
13    </div>
14  </div>
15
16 </div>
17
18 </div>
19
20 {% endblock %}
21

```

Ln 7, Col 48 Spaces: 4 UTF-8 LF HTML 01:00 19-11-2022



## University Admit Eligibility Predictor



GRE Score:

TOFEL Score:

University Rating:

SOP:

LOR:

CGPA:

Research:

☒ Yes

☐ No

Predict

## 8.TESTING

### 8.1 TEST CASES

To test and train the datasets

AutoSave Off Admission\_Predict1

File Home Insert Page Layout Formulas Data Review View Help

Undo Paste Copy Format Painter Clipboard Font Alignment

POSSIBLE DATA LOSS Some features might be lost if you save this workbook in the comma-delimited (.csv) format. To preserve t

L2

	A	B	C	D	E	F	G	H	I	J	K
1	Serial No.	GRE Score	TOEFL Sco	University	SOP	LOR	CGPA	Research	Chance of Admit		
2	1	337	118	4	4.5	4.5	9.65	1	0.92		
3	2	324	107	4	4	4.5	8.87	1	0.76		
4	3	316	104	3	3	3.5	8	1	0.72		
5	4	322	110	3	3.5	2.5	8.67	1	0.8		
6	5	314	103	2	2	3	8.21	0	0.65		
7	6	330	115	5	4.5	3	9.34	1	0.9		
8	7	321	109	3	3	4	8.2	1	0.75		
9	8	308	101	2	3	4	7.9	0	0.68		
10	9	302	102	1	2	1.5	8	0	0.5		
11	10	323	108	3	3.5	3	8.6	0	0.45		
12	11	325	106	3	3.5	4	8.4	1	0.52		
13	12	327	111	4	4	4.5	9	1	0.84		
14	13	328	112	4	4	4.5	9.1	1	0.78		
15	14	307	109	3	4	3	8	1	0.62		
16	15	311	104	3	3.5	2	8.2	1	0.61		
17	16	314	105	3	3.5	2.5	8.3	0	0.54		
18	17	317	107	3	4	3	8.7	0	0.66		
19	18	319	106	3	4	3	8	1	0.65		
20	19	318	110	3	4	3	8.8	0	0.63		
21	20	303	102	3	3.5	3	8.5	0	0.62		
22	21	312	107	3	3	2	7.9	1	0.64		
23	22	325	114	4	3	2	8.4	0	0.7		
24	23	328	116	5	5	5	9.5	1	0.94		
25	24	334	119	5	5	4.5	9.7	1	0.95		
26	25	336	119	5	4	3.5	9.8	1	0.97		
27	26	340	120	5	4.5	4.5	9.9	1	0.99		

Admission\_Predict1

Ready Accessibility: Unavailable

26°C Haze

Search

## **8.2 USER ACCEPTANCE TESTING**

User acceptance testing (UAT), also called application testing or end-user testing, is a phase of software development in which the software is tested in the real world by its intended audience. UAT is often the last phase of the software testing process and is performed before the tested software is released to its intended market. The goal of UAT is to ensure software can handle real-world tasks and perform up to development specifications.

UAT, users are given the opportunity to interact with the software before its official release to see if any features have been overlooked or if it contains any bugs. UAT can be done in-house with volunteers, by paid test subjects using the software or by making the test version available for download as a free trial. The results from the early testers are forwarded to the developers, who make final changes before releasing the software commercially. UAT is effective for ensuring quality in terms of time and software cost, while also increasing transparency with users.

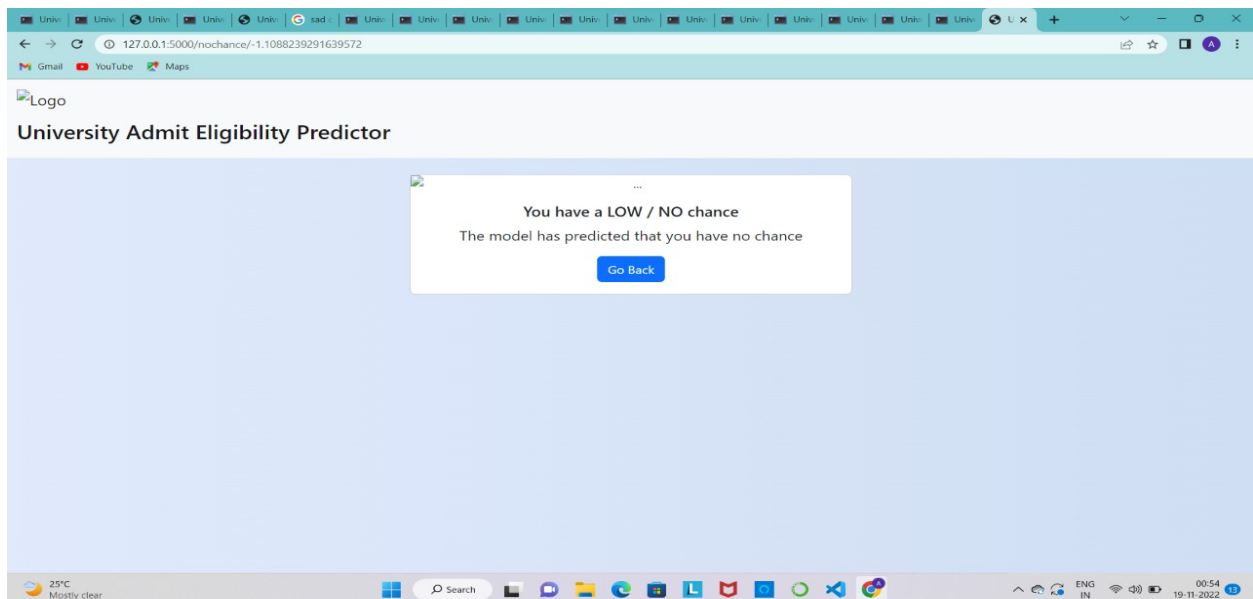
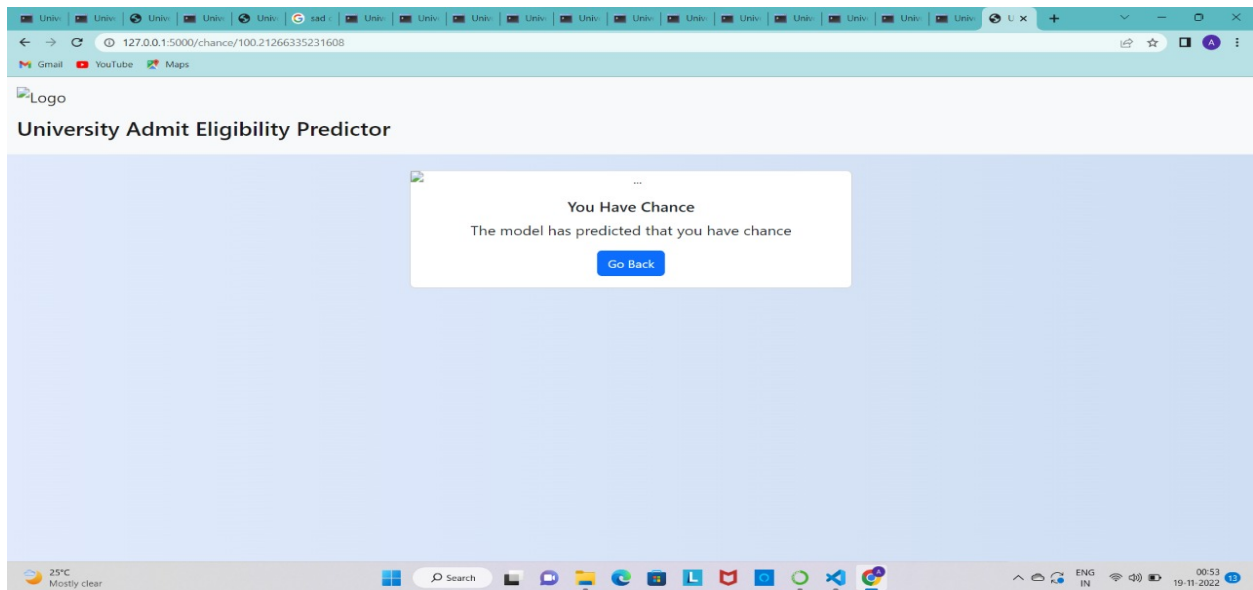
## 9.RESULTS

### 9.1 PERFORMANCE METRICS

The Performance is the Accuracy of the model trained

The Training accuracy of the model is 98.2126633523

The Testing accuracy of the model is 87.99628300



## **10.ADVANTAGES AND DISADVANTAGES**

### **ADVANTAGES:**

- This system helps the student for making the prediction for chance/no chance
- Avoids data redundancy and inconsistency
- Chance of occurrence of error is less
- User friendly to use
- Maintain the proper datasets
- easy accessibility of data

### **DISADVANTAGES**

- Security Concerns
- Need Sufficient Internet Connection
- System will provide inaccurate results if data entered incorrectly
- If there are no proper internet connection sometimes it seems the data might be invalid

## **11 CONCLUSION**

The prediction model web application is developed by using the machine learning algorithm. The backend process is implemented by using the Python flask. front end well designed by using HTML, CSS Integrated the front end and backend by using flask finally the web application designed and developed successfully the user can easily get to know about the prediction of chances.

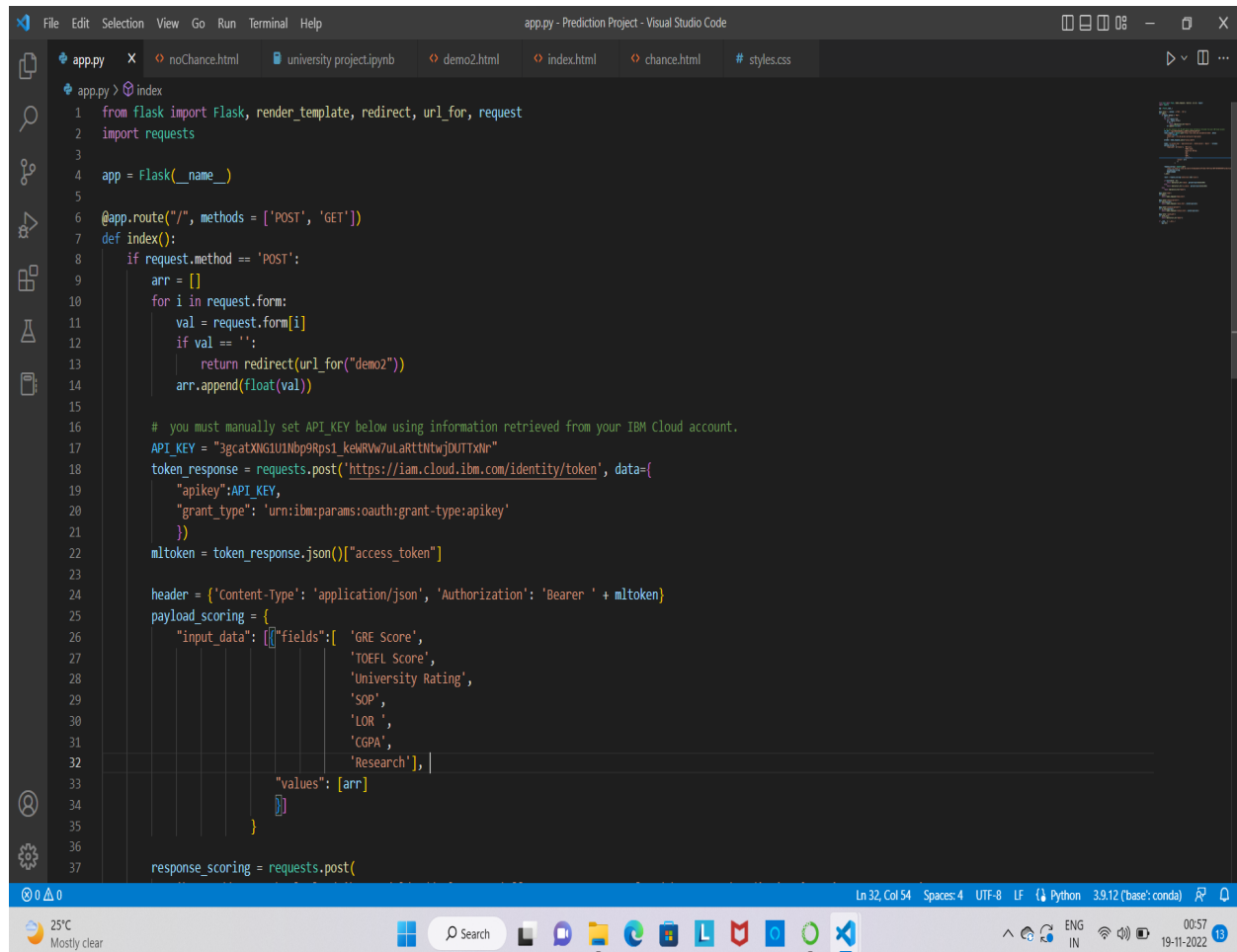
## **12.Future Scope**

Further enhancement can be made by secure the datasets in a proper manner. considering the existing model this system has been developed for the user's convenient. In future this model web application has develop by adding extra features by using the upcoming technology.



## 13.APPENDIX

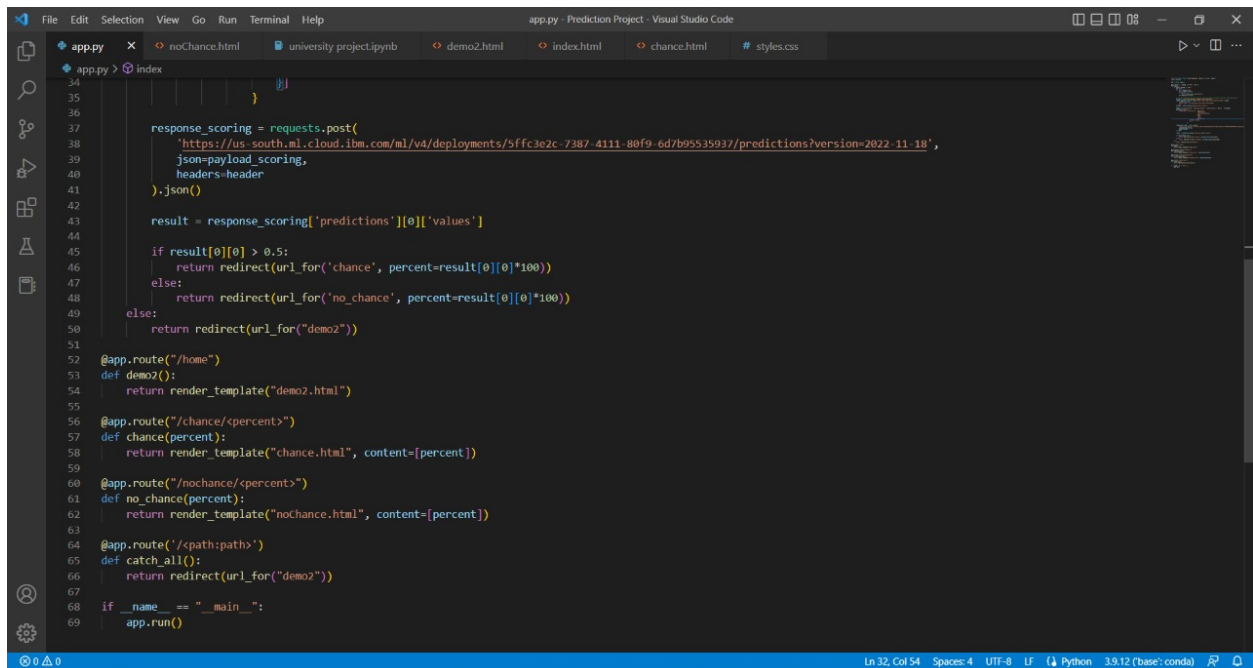
### SOURCE CODE



```
1 from flask import Flask, render_template, redirect, url_for, request
2 import requests
3
4 app = Flask(__name__)
5
6 @app.route("/", methods = ['POST', 'GET'])
7 def index():
8     if request.method == 'POST':
9         arr = []
10        for i in request.form:
11            val = request.form[i]
12            if val == '':
13                return redirect(url_for("demo2"))
14            arr.append(float(val))
15
16        # you must manually set API_KEY below using information retrieved from your IBM Cloud account.
17        API_KEY = "3gcatXNGIUINbp9Rpsi_keWRVw7uLarttNtwjDUTTxm"
18        token_response = requests.post("https://iam.cloud.ibm.com/identity/token", data={
19            "apikey":API_KEY,
20            "grant_type": 'urn:ibm:params:oauth:grant-type:apikey'
21        })
22        mltoken = token_response.json()["access_token"]
23
24        header = {'Content-Type': 'application/json', 'Authorization': 'Bearer ' + mltoken}
25        payload_scoring = {
26            "input_data": [{"fields": [
27                'GRE Score',
28                'TOEFL Score',
29                'University Rating',
30                'SOP',
31                'LOR ',
32                'CGPA',
33                'Research'],
34                "values": [arr]
35            }
36        ]
37        response_scoring = requests.post(
```

Ln 32, Col 54 Spaces: 4 UTF-8 LF Python 3.9.12 (base: conda)

25°C Mostly clear Search ENG IN 00:57 19-11-2022



```
34
35
36
37 response_scoring = requests.post(
38     'https://us-south.ml.cloud.ibm.com/ml/v4/deployments/5ffc3e2c-7387-4111-80f9-6d7b95535937/predictions?version=2022-11-18',
39     json=payload_scoring,
40     headers=header
41 ).json()
42
43 result = response_scoring['predictions'][0]['values']
44
45 if result[0][0] > 0.5:
46     return redirect(url_for('chance', percent=result[0][0]*100))
47 else:
48     return redirect(url_for('no_chance', percent=result[0][0]*100))
49 else:
50     return redirect(url_for("demo2"))
51
52 @app.route("/home")
53 def demo2():
54     return render_template("demo2.html")
55
56 @app.route("/chance/<percent>")
57 def chance(percent):
58     return render_template("chance.html", content=[percent])
59
60 @app.route("/nochance/<percent>")
61 def no_chance(percent):
62     return render_template("noChance.html", content=[percent])
63
64 @app.route('/<path:path>')
65 def catch_all():
66     return redirect(url_for("demo2"))
67
68 if __name__ == "__main__":
69     app.run()
```

**GITHUB LINK :**

**IBM-EPBL/IBM-Project-43914-1660720604**