UNIVERSITY ADMIT ELIGIBILITY PREDICTOR
TEAM ID - PNT2022TMID14377

1.INTRODUCTION

1.1 Project overview:

- In today's era we see a lot of students pursuing their education away from their home countries. The main country targeted by these international students is the United States of America.
- Majority of the international students in the United States of America are from India and China.
- In the past decade the number of Indian students pursuing post graduate education from the USA has rapidly increased.
- With the increase in the number of international students studying in the USA, each applicant has to face a tough competition to get admission in their dream university.
- Our project will assist students in obtaining a list of universities to which they may be admitted based on their marks.
- This System is a web based application system in which students can register their marks this helps to predict their admissions in colleges.

1.2 Purpose:

- The main objective of our system is to make the right choice of colleges. Whatever may be their scores, this application helps to find the best colleges for them.
- This is to make the admission process easy but also minimises stress for students.
- An Admission Point Score (APS) is the score universities and colleges use to determine whether or not your college application will be approved.

2. LITERATURE SURVEY:

2.1 Existing problem:

- Prediction is not accurate.
- Not cost efficient.
- System doesn't verifies the correct input data and make the inaccurate results.
- Required active internet connection.
- It is not reliable.
- Low computer literacy and security concerns.

2.2 References:

[1] Abdul Fatah S; M, A. H. (2012). Hybrid Recommender System for Predicting College Admission, pp. 107–113.

[2] Bibodi, J., Vadodaria, A., Rawat, A. and Patel, J. (n.d.). Admission Prediction System Using Machine Learning.

[3] Eberle, W., Simpson, E., Talbert, D., Roberts, L. and Pope, A. (n.d.). Using Machine Learning and Predictive Modeling to Assess Admission Policies and Standards.

[4] Jamison, J. (2017). Applying Machine Learning to Predict Davidson College ' s Admissions Yield, pp. 765–766.

[5] Mane, R. V. (2016). Predicting Student Admission decisions by Association Rule Mining with Pattern Growth Approach, pp. 202–207.

[6] Waters, A. and Miikkulainen, R. (2013). G RADE : Machine Learning Support for Graduate Admissions.

2.3 Problem Statement Definition:

1. The problem statement is to design a college prediction system and to provide a probabilistic insight into college administration for overall rating, cut offs of the colleges, admission intake and preferences of students.

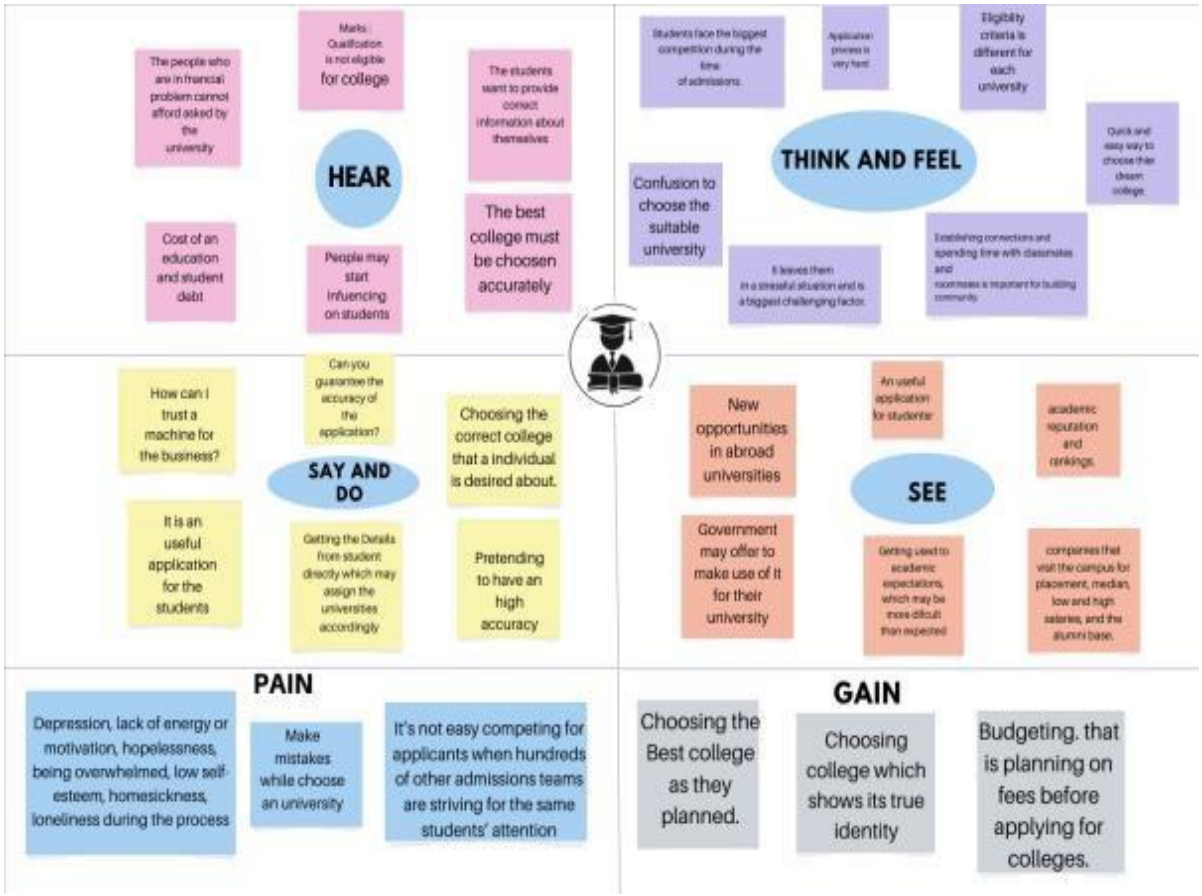
2. It has always been a troublesome process for students in finding the perfect university and course for their further studies.

3. At times they do know which stream they want to get into, but it is not easy for them to find colleges based on their academic marks and other performances.

4. We aim to develop and provide a place which would give a probabilistic output of how likely it is to get into a university given their details.

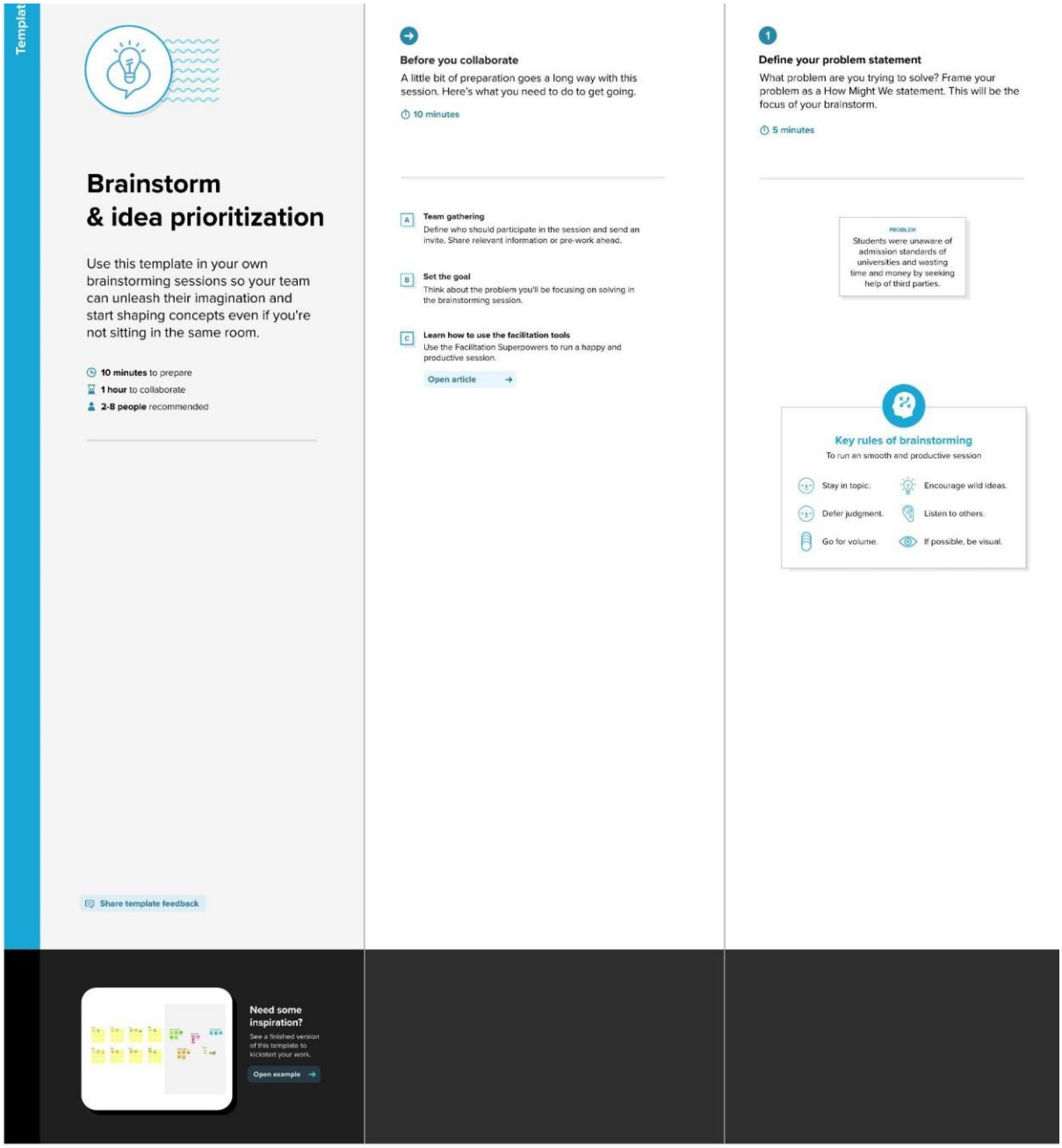
3.IDEATION & PROPOSED SOLUTION:

3.1 Empathy Map:

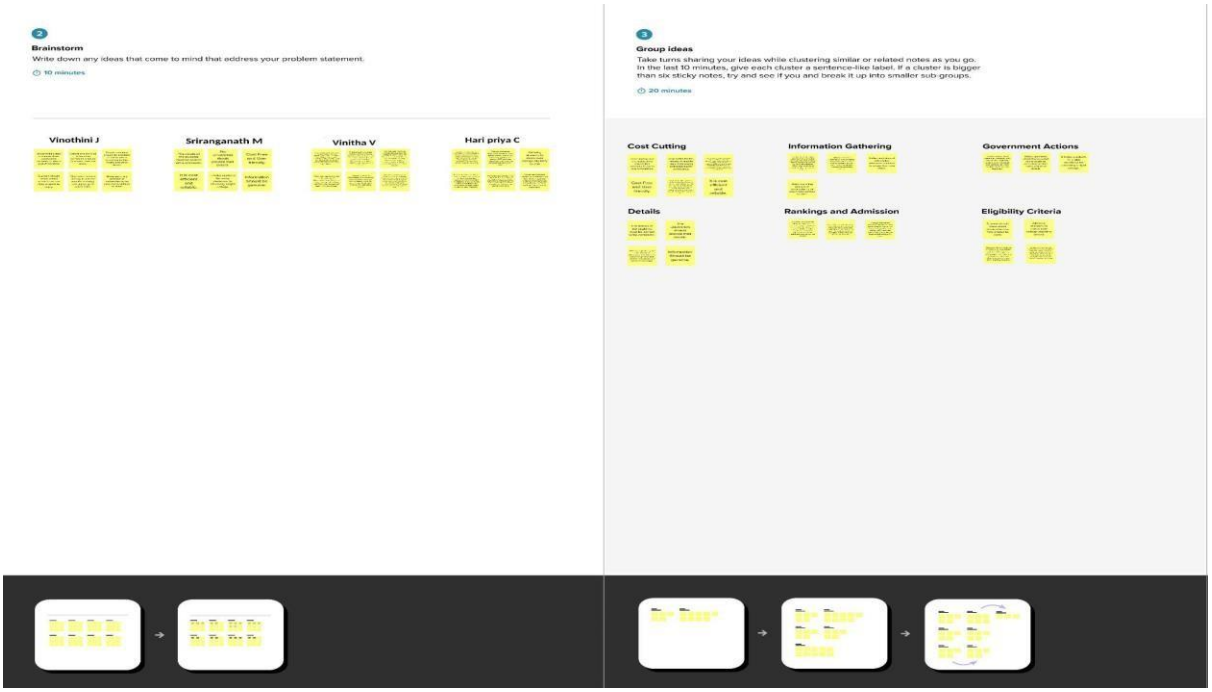


3.2 Ideation & Brainstorming:

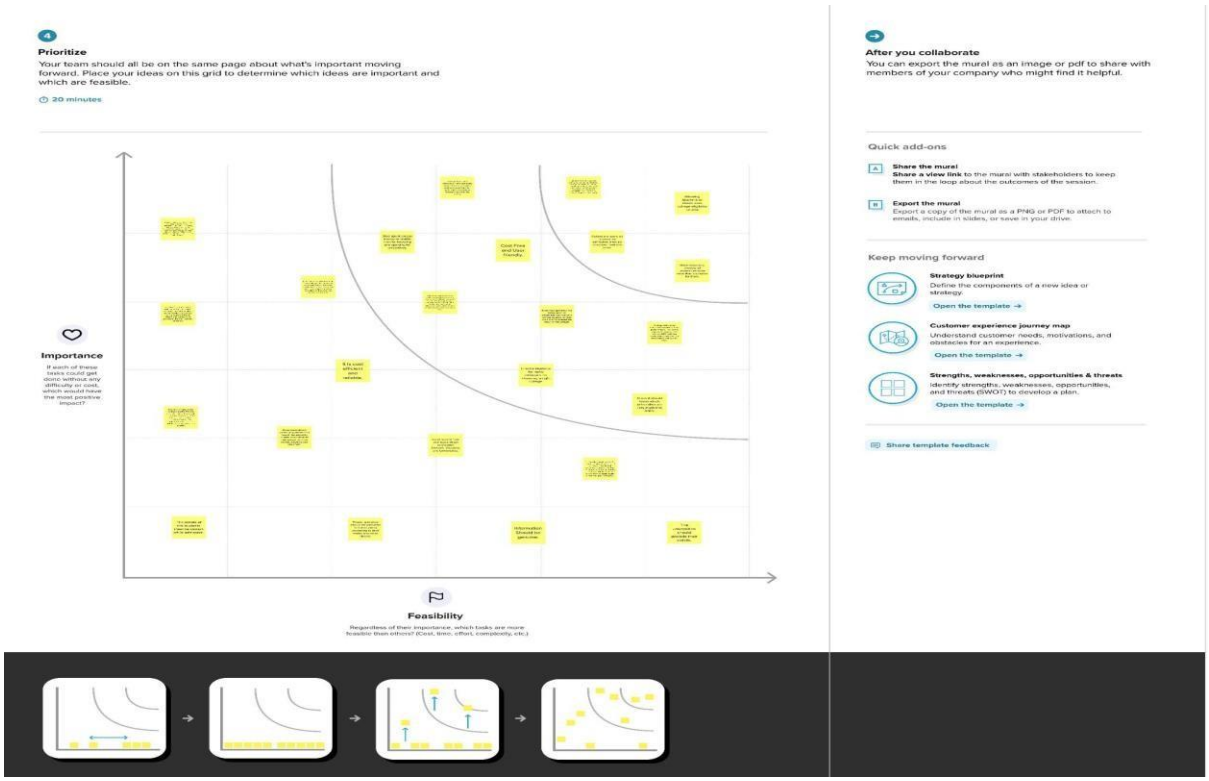
STEP-1:



STEP – II :



STEP-III:



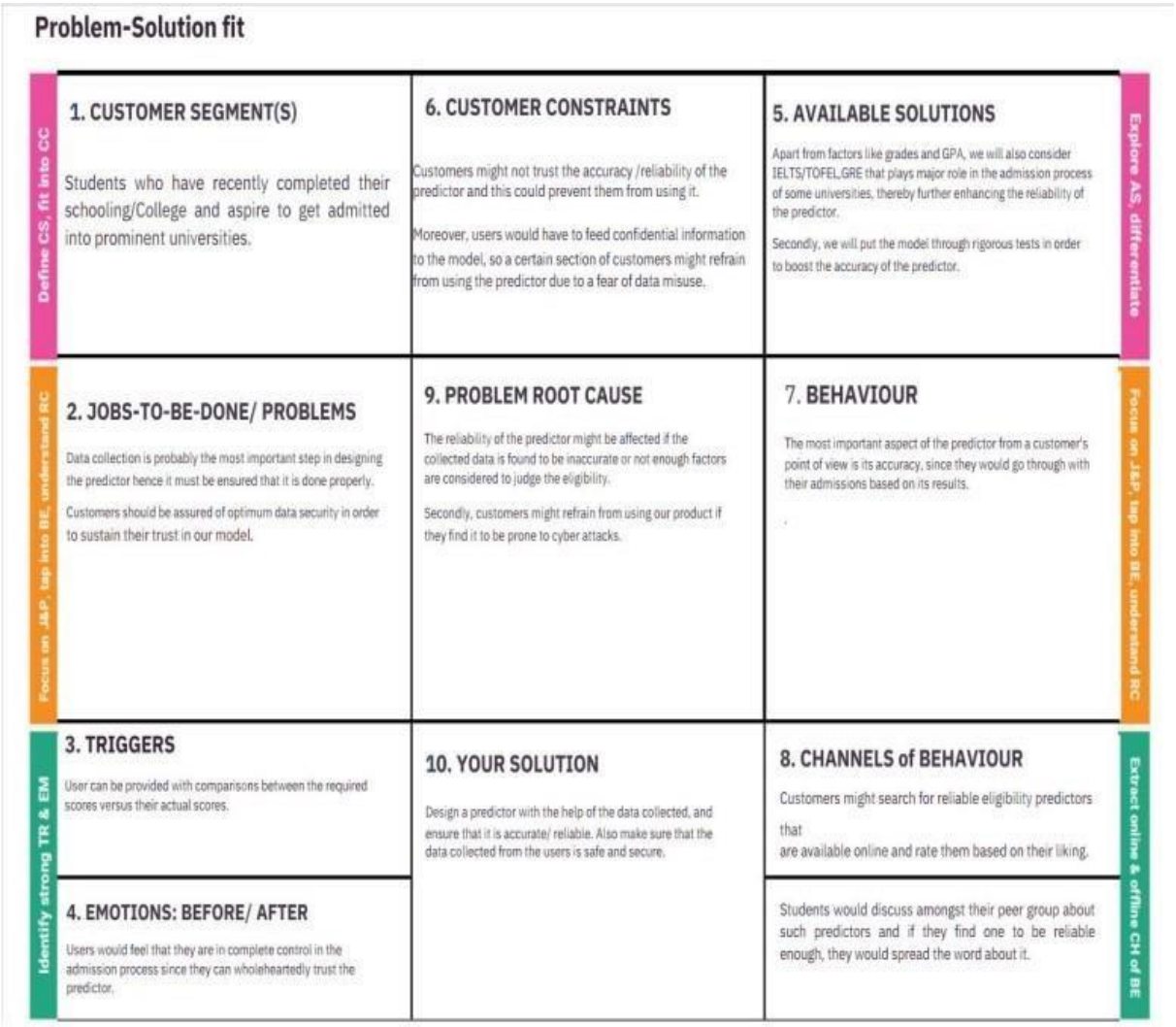
3.3 Proposed Solution:

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Students do not have much idea about the procedures, requirements and details of the universities they want to join, so they seek help from various educational consultancies to help them secure admission in the universities based on their profile, for which the students are supposed to pay a hefty amount as consultancy fee.
2.	Idea / Solution description	Providing an accurate prediction for the student's admission into the university of their choice based on various parameters like IELTS, GRE, Academic Performance, etc..
3.	Novelty / Uniqueness	It seems there are no web applications for predicting the eligibility criteria of a student for getting in to their dream university and also provide personalized insights on specific areas where they can improve.

4.	Social Impact / Customer Satisfaction	It helps student in making the right decision for choosing the universities. It cuts the cost of consultancy services by creating a direct connection between students and universities.
5.	Business Model (Revenue Model)	Universities are under immense pressure to admit more students and ensure student success. To overcome this pressure, they can make use of predictive models which help them to ease the intake process of students and improve efficiency.
6.	Scalability of the Solution	Further to reduce the immense pressure faced by the students to get admitted in a university, the model can also be evolved to consider university specific examinations and to maintain the latest eligibility criteria.

3.4 Problem Solution fit :

Problem-Solution fit



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 Gmail
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	User Login	Login through username and password Login through Gmail
FR-4	Administration work	Check qualified candidate detail Make allotment
FR-5	Admission Details	Check seat availability Check college infrastructure Check fees details
FR-6	Local councillor	Issue the final allotment order

4.2 Non-Functional requirements:

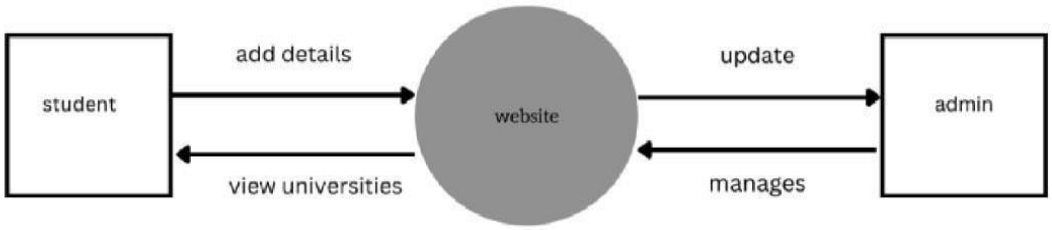
FR No.	Non-Functional Requirement	Description
NFR-1	Usability	i. A logical interface is essential to make easy use of the system, speeding up common tasks. ii. The product could be used by two categories of people, mainly administrator category and other users.
NFR-2	Security	Some of the factors that are identified to protect the software from accidental or malicious access, use, modification, destruction, or disclosure are described below: i. Keep specific log or history data sets. ii. Utilise certain cryptographic techniques. iii. Restrict the number of systems that can access the online admission system site. iv. Check data integrity for critical variables. v. Every user should be licensed to use the system under any of the four categories provided i.e. either verifier or advisor or local counsellor or administrator. vi. Communication needs to be restricted when the application is validating the user or licence.
NFR-3	Reliability	i. All data storage for user variables will be committed to the database at the time of entry. ii. Data corruption is prevented by applying the possible backup procedures and techniques.
NFR-4	Performance	i. The database should be able to accommodate a minimum of

		10,000 records of students. ii. At any instant the system should support use of multiple users at a time. iii. Availability results of the requested college should be presented to the student in max of two seconds, so retrieving data should be reliable. iv. As each student will be given a maximum time of 10min, accessing from the database should be done at relevant speed.
NFR-5	Availability	The system should be available at all the time meaning that the user can access easily. Increase of the hardware and database failure a replacement page will be shown and for database back should be retrieved from the data folder.
NFR-6	Scalability	The system should be available at all times, meaning that the user can access it easily. Increase of the hardware and database failure a replacement page will be shown and for database back should be retrieved from the data folder.

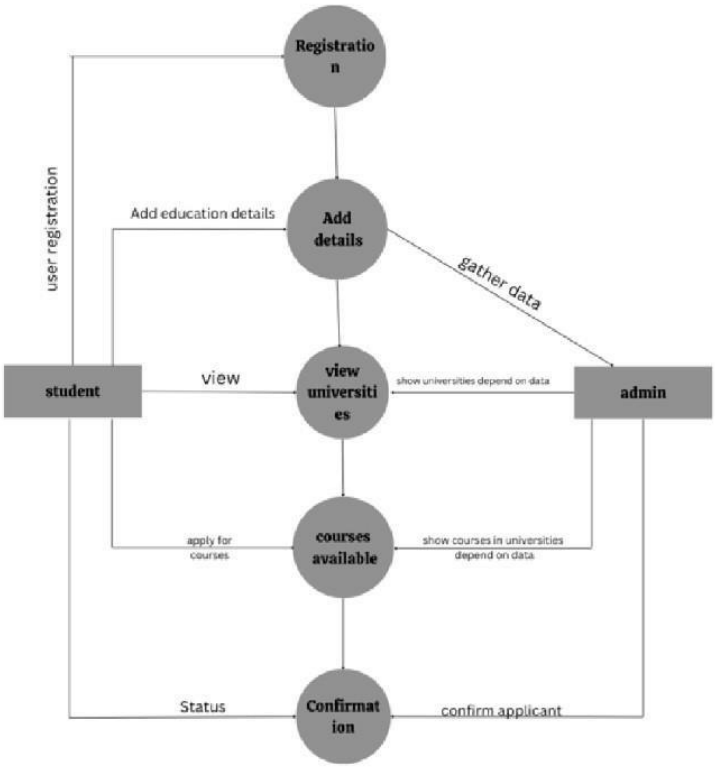
5. PROJECT DESIGN:

5.1 Data Flow Diagrams:

Level – 1:

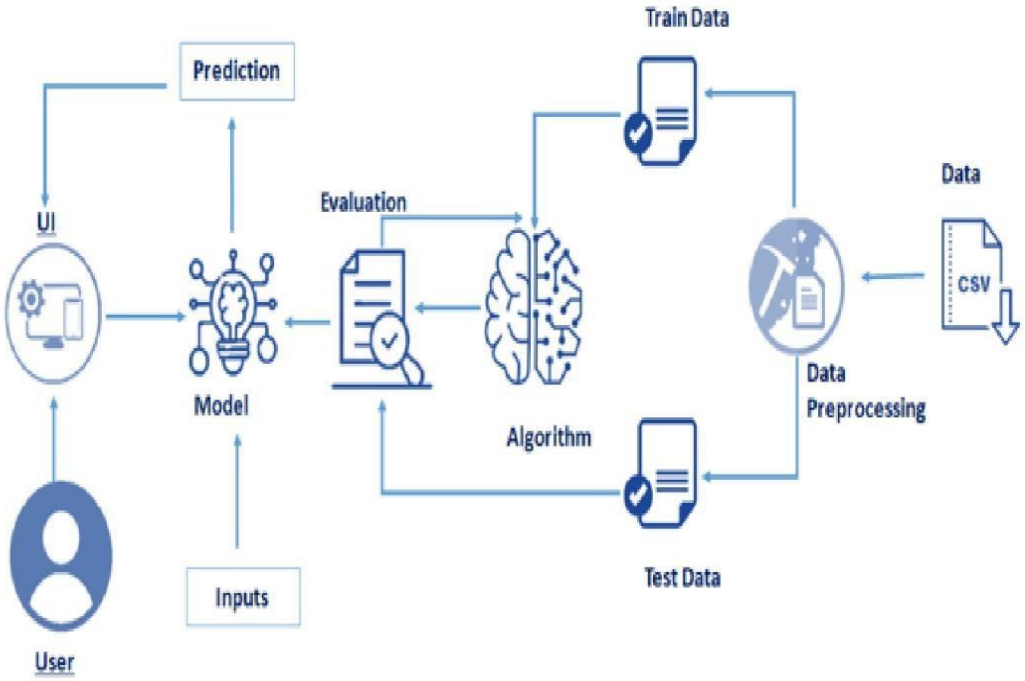


Level-2:



5.2 Solution & Technical Architecture:

Technical Architecture:



5.3 User Stories:

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Web user)	Personal Details	USN-1	As a user, I can Give my academic information 65tin the profile section	I can access my dashboard	High	Sprint-1
		USN-2	As a user, I will be able to select a locationthat I prefer	I can receive the list of location in the dropdownto select	High	Sprint-1
	Search	USN-3	As a user I can search for my preferred university	I can use the search bar	Medium	Sprint-2
	User Preference	USN-4	As a user, I can select my preferred universityfrom the list to check my eligibility for the particular university	I can use the dropdown list provided to select the university	Medium	Sprint-2
		USN-5	As a user, I can select my preferred location	I can select my preferred location	High	Sprint-1
		USN-6	As a user, I will be able to select my preferred Course	I can select a course from the dropdown list	Medium	Sprint-1
	Result	USN-7	As a user, I can view the list of	I can view the list of universities	High	Sprint-3

			universities that I am eligible in accordance to my preferred location	filtered by the model		
		USN-8	As a user, I can access the link to the university that I am eligible from the list	I can access the university link	Medium	Sprint-3
		USN-9	As a user, I can access the location link of theuniversity that I am eligible from the list	I can access the university location link	Low	Sprint-3
		USN-10	From the list of universities, I can select andview the eligibility for the particular university	I can view the eligible university	Medium	Sprint-3

6. PROJECT PLANNING & SCHEDULING:

6.1 Sprint Planning & Estimation:

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	User Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	2	High	2
Sprint-1		USN-2	As a user, I will receive confirmation email once I have registered for the application	1	High	1
Sprint-2		USN-3	As a user, I can check the eligibility criteria for various universities by uploading the necessary documents.	2	Low	2
Sprint-3		USN-4	As a user, I can register	2	Medium	2

			for the desired university through Gmail and can also upload further course completion documents if necessary.			
Sprint-4	User Login	USN-5	As a user, I can log into the application by entering email & password.	1	High	2
	Dashboard		Check dashboard for further updates and upload the details according to the desired and eligible universities based on the eligibility criteria.			4

6.2 Sprint Delivery Schedules:

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	24-10-22	29-10-22	20	31 Oct 2022
Sprint-2	20	6 Days	31-10-22	05-11-22	20	07 Nov 2022
Sprint-3	20	6 Days	07-11-22	12-11-22	15	14 Nov 2022
Sprint-4	20	6 Days	14-11-22	19-11-22	25	19 Nov 2022

7. CODING & SOLUTIONING:

7.1 Feature 1:

Predict the accurate results:

It will predict the accuracy results that will result in the accurate chance for university.

You Have Chance

Predicted score that you have **79.36534%** chance

ALL THE BEST....!

Back

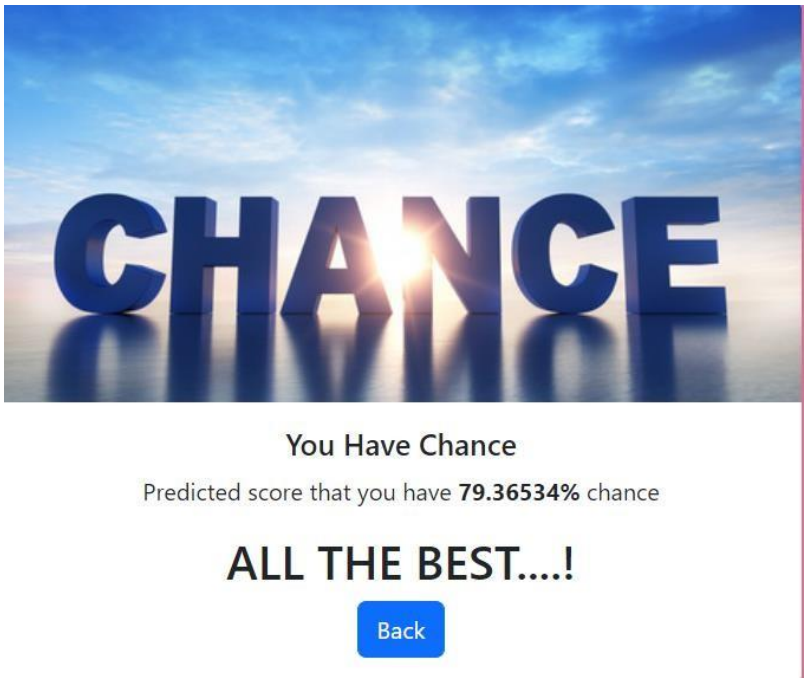
7.3 Database Schema:

Serial No.	GRE Score	TOEFL Sco	University	SOP	LOR	CGPA	Research	Chance of Admit
1	337	118	4	4.5	4.5	9.65	1	0.92
2	324	107	4	4	4.5	8.87	1	0.76
3	316	104	3	3	3.5	8	1	0.72
4	322	110	3	3.5	2.5	8.67	1	0.8
5	314	103	2	2	3	8.21	0	0.65
6	330	115	5	4.5	3	9.34	1	0.9
7	321	109	3	3	4	8.2	1	0.75
8	308	101	2	3	4	7.9	0	0.68
9	302	102	1	2	1.5	8	0	0.5
10	323	108	3	3.5	3	8.6	0	0.45
11	325	106	3	3.5	4	8.4	1	0.52
12	327	111	4	4	4.5	9	1	0.84
13	328	112	4	4	4.5	9.1	1	0.78
14	307	109	3	4	3	8	1	0.62
15	311	104	3	3.5	2	8.2	1	0.61
16	314	105	3	3.5	2.5	8.3	0	0.54
17	317	107	3	4	3	8.7	0	0.66
18	319	106	3	4	3	8	1	0.65
19	318	110	3	4	3	8.8	0	0.63
20	303	102	3	3.5	3	8.5	0	0.62
21	312	107	3	3	2	7.9	1	0.64
22	325	114	4	3	2	8.4	0	0.7
23	328	116	5	5	5	9.5	1	0.94
24	334	119	5	5	4.5	9.7	1	0.95
25	336	119	5	4	3.5	9.8	1	0.97
26	340	120	5	4.5	4.5	9.6	1	0.94
27	322	109	5	4.5	3.5	8.8	0	0.76

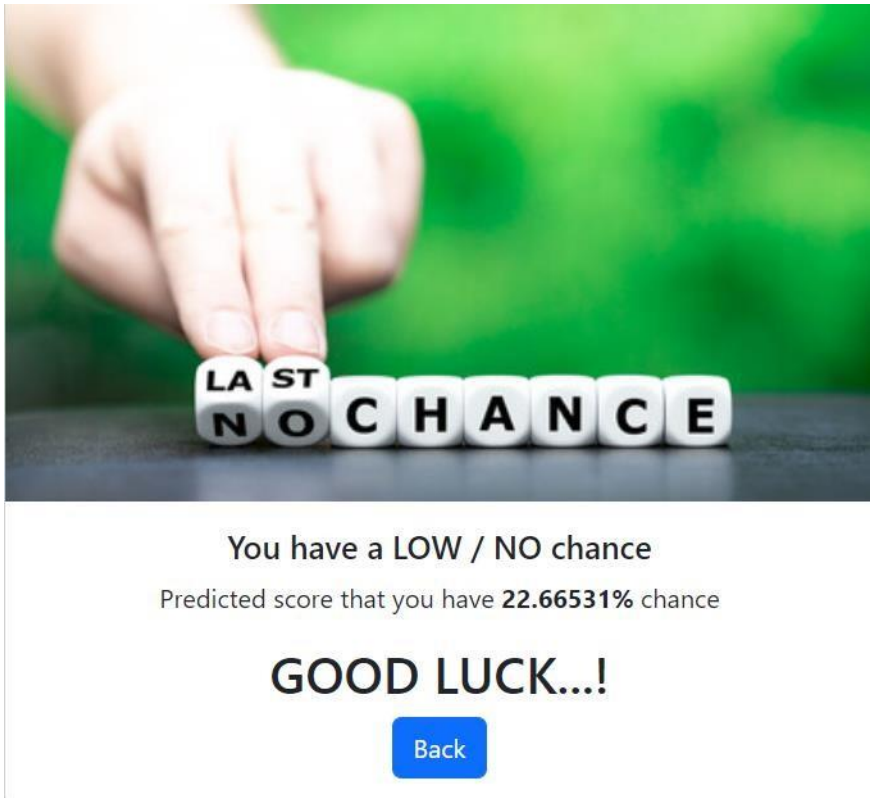
8. TESTING:

8.2 User Acceptance Testing:

Chance Acceptance



No chance



9. RESULTS:

9.1 Performance Metrics:

S.No	Parameter	Values
1	Metrics	**Regression Model:** MAE– 0.03790692243018498, MSE – 0.003058753436307664, RMSE – 0.05530599819465936, R2 score – 0.8647260941958439 **Classification Model:** Accuracy Score > 50 – eligible for university Accuracy Score < 50 – eligible for university
2	Tune the model	**Hyperparameter Tuning:** 0.9166666666666666 **Validation Method:** Random forest

10. ADVANTAGES:

- It helps students make decisions for choosing the right college.
- Here the chance of occurrence of error is less when compared with the existing system.
- It is fast, efficient and reliable.
- Avoids data redundancy and inconsistency.
- Very user-friendly.
- Easy accessibility of data.
- Increases Accuracy and Efficiency

DISADVANTAGES:

- ❖ Requires active internet connection.
- ❖ It can only prediction accuracy of the students doesn't give any recommendations of colleges.

11. CONCLUSION:

This system predicts the correct accuracy with the four decimal values and it will enable the progress that helps you to predict the chance of admission in the Abroad universities.

12. FUTURE SCOPE:

The future scope of this project is very broad. Few of them are:

- This can be implemented in less time for proper admission process.
- This can be accessed anytime anywhere, since it is a web application provided only an internet connection.
- The user had not need to travel a long distance for the admission and his/her time is also saved as a result of this automated system.

13. APPENDIX:

Source Code:

Universityprediction.py

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings('ignore')
%matplotlib inline
```

```
import os, types
import pandas as pd
from botocore.client import Config
import ibm_boto3

def __iter__(self): return 0

# @hidden_cell
# The following code accesses a file in your IBM Cloud Object Storage. It
includes your credentials.
# You might want to remove those credentials before you share the notebook.
cos_client = ibm_boto3.client(service_name='s3',
    ibm_api_key_id='OnXOxYJpFaS7YsiRvshtwtPLtBm60B0_lUknUsVZKqqN',
    ibm_auth_endpoint="https://iam.cloud.ibm.com/oidc/token",
    config=Config(signature_version='oauth'),
    endpoint_url='https://s3.private.us.cloud-object-storage.appdomain.cloud')

bucket = 'universityeligibilitypredictor-donotdelete-pr-xiux87vd9z54fy'
object_key = 'Admission_Predict.csv'

body = cos_client.get_object(Bucket=bucket,Key=object_key)['Body']
# add missing__iter__ method, so pandas accepts body as file-like object
if not hasattr(body, "_iter__"): body.__iter__= types.MethodType(__iter__,
body )

data= pd.read_csv(body)
data.head()

data.info()
corr_matrix = data.corr()
corr_matrix
#plotting the correlation matrix as a heatmap
```

```
fig = plt.figure(figsize=(12,8))
sns.heatmap(corr_matrix,annot=True)
plt.show()
#plotting data which have high correlation

sns.relplot(data=data,x="GRE Score",y="Chance of Admit ",hue="Research",
            palette="Set1",alpha=0.7)
plt.title("GRE Score vs Chance of Admit")
plt.show()
sns.relplot(data=data,x="CGPA",y="Chance of Admit ",hue="Research",
            palette="Set2",alpha=0.7)
plt.title("CGPA vs Chance of Admit")
plt.show()
sns.relplot(data=data,x="TOEFL Score",y="Chance of Admit ",hue="Research",
            palette="Set1",alpha=0.7,kind="line",ci=None)
plt.title("TOEFL Score vs Chance of Admit")
plt.show()
sns.barplot(data=data,x="SOP",y="Chance of Admit ",
            palette="Set2",hue="Research")
plt.title("SOP vs Chance of Admit")
plt.show()
sns.barplot(data=data,x="LOR ",y="Chance of Admit ",
            palette="Set3",hue="Research")
plt.title("LOR vs Chance of Admit")
plt.show()
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import r2_score
x = data[["GRE Score","TOEFL Score","University Rating","SOP","LOR ","CGPA",
"Research"]]
y = data["Chance of Admit "].values.reshape(-1,1)

x_train, x_test, y_train, y_test =
train_test_split(x,y,test_size=0.2,random_state=42)
#implying multiple linear regression and determining its score

multiple_lin_reg = LinearRegression()
multiple_lin_reg.fit(x_train,y_train)

y_pred_mlr = multiple_lin_reg.predict(x_test)

r2_score_mlr = r2_score(y_test,y_pred_mlr)
print("Mutiple Linear Regression's Score = {:.3f}".format(r2_score_mlr))
#implying decision tree regression and determining its score
```

```
tree_reg = DecisionTreeRegressor()
tree_reg.fit(x_train,y_train)

y_pred_tree = tree_reg.predict(x_test)

r2_score_tree = r2_score(y_test,y_pred_tree)
print("Decision Tree Regression's Score = {:.3f}".format(r2_score_tree))
#implying random forest regression and determining its score

ran_for_reg = RandomForestRegressor(n_estimators=100,random_state=42)
ran_for_reg.fit(x_train,y_train)

y_pred_rfr = ran_for_reg.predict(x_test)

r2_score_rfr = r2_score(y_test,y_pred_rfr)
print("Random Forest Regression's Score = {:.3f}".format(r2_score_rfr))
from ibm_watson_machine_learning import APIClient
wml_credentials = {
    "url": "https://us-south.ml.cloud.ibm.com",
    "apikey": "8cqf_Qc-PZbuOGpinZfs1SrsP-JbjMjye2X9oY4X-63H"
}

client = APIClient(wml_credentials)
def guid_from_space_name(client, space_name):
    space = client.spaces.get_details()
    return (next(item for item in space['resources'] if item['entity']['name']
== space_name)['metadata']['id'])
space_uid = guid_from_space_name(client, 'newspace')
print("Space UID = " + space_uid)
client.software_specifications.list()
#Set Python Version
software_spec_uid = client.software_specifications.get_uid_by_name("runtime-
22.1-py3.9")
software_spec_uid
model_details = client.repository.store_model(model = multiple_lin_reg,
meta_props={
    client.repository.ModelMetaNames.NAME: "UAEP_Multiple_Linear_Regression",
    client.repository.ModelMetaNames.TYPE: "scikit-learn_1.0",
    client.repository.ModelMetaNames.SOFTWARE_SPEC_UID: software_spec_uid
}
)

model_id = client.repository.get_model_id(model_details)
model_id
import requests

# NOTE: you must manually set API_KEY below using information retrieved from
your IBM Cloud account.
```

```
API_KEY = "8cqf_Qc-PZbuOGpinZfs1SrsP-JbjMjye2X9oY4X-63H"
token_response = requests.post('https://iam.cloud.ibm.com/identity/token',
data={"apikey":
 API_KEY, "grant_type": 'urn:ibm:params:oauth:grant-type:apikey'})
mltoken = token_response.json()["access_token"]

header = {'Content-Type': 'application/json', 'Authorization': 'Bearer ' +
mltoken}

# NOTE: manually define and pass the array(s) of values to be scored in the
next line
payload_scoring = {"input_data": [{"field": [["GRE Score","TOEFL
Score","University Rating","SOP","LOR ","CGPA", "Research"]], "values": [[326,
110, 2, 3.5, 4, 9.23, 1]]}]}

response_scoring = requests.post('https://us-
south.ml.cloud.ibm.com/ml/v4/deployments/35d1a0c8-7921-4798-bef9-
c20ebffd3b9d/predictions?version=2022-11-17', json=payload_scoring,
headers={'Authorization': 'Bearer ' + mltoken})
print("Scoring response")

pred = response_scoring.json()
print(pred)

probability = response_scoring.json()['predictions'][0]['values'][0][0][0]
probability
```

GitHub & Project Demo Link:

Github link - https://github.com/IBM-EPBL/IBM-Project-37651-1660315956

Video link - demo video link