

University Admit Eligibility Predictor
Professional Readiness for Innovation, Employability and
Entrepreneurship

Team ID: PNT2022TMID21562

Team Members

DHARSHINI SHREE C Y	(19IT022)
SARVESH V S	(19IT084)
SOWMYA V	(19IT099)
VIBHISHEAK L S	(19IT116)

Dissertation submitted in partial fulfillment of the requirements for the degree of

BACHELOR OF TECHNOLOGY

Branch: INFORMATION TECHNOLOGY



NOVEMBER 2022

DEPARTMENT OF INFORMATION TECHNOLOGY

THIAGARAJAR COLLEGE OF ENGINEERING

(Autonomous Institution)

Madurai – 625 015

TABLE OF CONTENTS

1.	INTRODUCTION	-4
1.1	Project Overview	
1.2	Purpose	
2.	LITERATURE SURVEY	-5
2.1	Existing problem	
2.2	References	
2.3	Problem Statement Definition	
3.	IDEATION & PROPOSED SOLUTION	-8
3.1	Empathy Map Canvas	
3.2	Ideation & Brainstorming	
3.3	Proposed Solution	
3.4	Problem Solution fit	
4.	REQUIREMENT ANALYSIS	-11
4.1	Functional requirement	
4.2	Non-Functional requirements	
5.	PROJECT DESIGN	-12
5.1	Data Flow Diagrams	
5.2	Solution & Technical Architecture	
5.3	User Stories	
6.	PROJECT PLANNING & SCHEDULING	-14
6.1	Sprint Planning & Estimation	
6.2	Sprint Delivery Schedule	
6.3	Reports from JIRA	
7.	CODING & SOLUTIONING	-16
7.1	Feature 1	
7.2	Feature 2	
7.3	Database Schema	

8.	TESTING	-18
8.1	Test Cases	
8.2	User Acceptance Testing	
9.	RESULTS	-22
9.1	Performance Metrics	
10.	ADVANTAGES & DISADVANTAGES	-23
11.	CONCLUSION	-23
12.	FUTURE SCOPE	-23
13.	APPENDIX	-23
14.	GITHUB AND DEMO LINK	-29

University Admit Eligibility Predictor

INTRODUCTION

1. Project Overview

PROBLEM STATEMENT:

Students are often worried about their chances of admission to the university. There are so many universities. Surfing through all the universities' past details is a time-consuming process.

OBJECTIVES:

- To build a model which gives accurate results
- To build a user-friendly web-based application for the users
- To create a simple user interface getting all the required input from the users
- To integrate the model with the web application using flask
- To deploy the model on IBM cloud

DATASET:

The dataset is taken from Kaggle. The dataset contains several parameters which are considered important during the application for Masters Programs.

- GRE Scores (out of 340)
- TOEFL Scores (out of 120)
- University Rating (out of 5)
- Statement of Purpose and Letter of Recommendation Strength (out of 5)
- Undergraduate GPA (out of 10)
- Research Experience (either 0 or 1)
- Chance of Admit (ranging from 0 to 1)

APPROACH:

The dataset is trained using different models and the model which gives high accuracy is considered. Random forest model gives the highest accuracy of 94%.

2. Purpose

- A web-based application in which students can register and enter their grades to determine whether or not they will be admitted to a university with a specific ranking can help the students proceed further.
- The analysis can help students who are currently preparing for exams or who have received their results gain a better understanding.
- It helps the students to know the weightage of each category of marks and to concentrate more in it.

2. LITERATURE SURVEY

1. Existing problem

S.NO	RESEARCH PAPER	AUTHOR	ALGORITHM USED	DATASET	INFERENCE
1	Prediction of the admission lines of college entrance examination based on machine learning	Zhenru Wang, Yijie Shi	Adaboost algorithm is used to study and forecast, which belongs to ensemble learning.	The data were selected from 2006 to 2015 in Sichuan Province. TotalNumber: total number of indicates of CEE in province Sichuan. NumberOtKeyUniversity: enrolment plan of first batch of undergraduate. NumberOfUniversity: enrolment plan of second batch of undergraduate. NumberOfMath: the number of science students who take part in CEE. Param: difficulty of test question.	AdaBoost also called Adaptive Boosting is a technique in Machine Learning used as an Ensemble Method. The most common algorithm used with AdaBoost is decision trees with one level that means with Decision trees with only 1 split. These trees are also called Decision Stumps.

S.NO	RESEARCH PAPER	AUTHOR	ALGORITHM USED	DATASET	INFERENCE
2	Research on Prediction of College Students' Performance Based on Support Vector Machine	Peng Wang , Yinshan Jia	Support vector machine was used to establish a college course performance prediction model, and cross-validation methods were used to obtain the best parameters and a reliable and stable model	The 2016 college computer science and technology and communication engineering students of a university were selected as the experimental data.	The prediction accuracy rate reached 73.6%. The prediction results show that the support vector machine can accurately predict college course performance based on the college entrance examination results.

S.NO	RESEARCH PAPER	AUTHOR	ALGORITHM USED	DATASET	INFERENCE
3	A University Admission Prediction System using Stacked Ensemble Learning	Sashank Sridhar, Siddhartha Mootha, Santosh Kolagati	MULTI LAYER PERCEPTRON AND THE STACKED ENSEMBLE MODEL. The proposed model takes into consideration various factors related to the student including their research experience, industry experience etc. The system proposed has been evaluated against various other machine learning algorithms including other deep learning methods. It is observed that the proposed model easily outperforms all other models and provides a very high accuracy.	The dataset consists of the applicant scores such as TOEFL, IELTS, GRE, GMAT, CGPA. web crawler has been used to extract student details from Edulix for 45 universities shortlisted. The scraped data had a total of 22 features and 50,000 samples.	The proposed ensemble neural network is evaluated by comparing it to other supervised algorithms such as Decision Trees, Random Forest, K-Nearest Neighbor, Naive Bayes Classifier, Logistic Regression, Support Vector Machine, (SVM), Linear Discriminant Analysis and Quadratic Discriminant Analysis. Ensemble NN has the highest accuracy.

S.NO	RESEARCH PAPER	AUTHOR	ALGORITHM USED	DATASET	INFERENCE
4	Engineering & Technology Admission Analysis and Prediction	Sachin Bhimrao Bhoite, Ajit More	Logistic Regression, K Nearest Neighbours', Decision Tree Classifier, Random Forest Classifier, Naive Bayes & Support Vector Machine Supervised Machine Learning Algorithms. Out all six models Decision Tree classifier & Random Forest always give great accuracy.	Various non-aided but affiliated to Savitribai Phule Pune University's engineering colleges admission record of academic year 2015-16 are considered. This Dataset has various attributes, which are: 'Main Serial No.', 'Sr. No.', 'College Name', 'College Code', 'Merit No', 'Merit Marks', 'Candidate Name', 'Gender', 'Candidate Type', 'Category', 'Home University', 'PH Type', 'Defence Type', 'HSC Eligibility', 'Seat Type', 'Fees Paid', 'CAP Round', 'Admitted/Uploaded Late', 'BRANCH' and 'NATIONALITY'.	Feature engineering is very essential part while implementing & building predictive models using machine learning techniques. The results have been more improved after feature engineering.

2. References

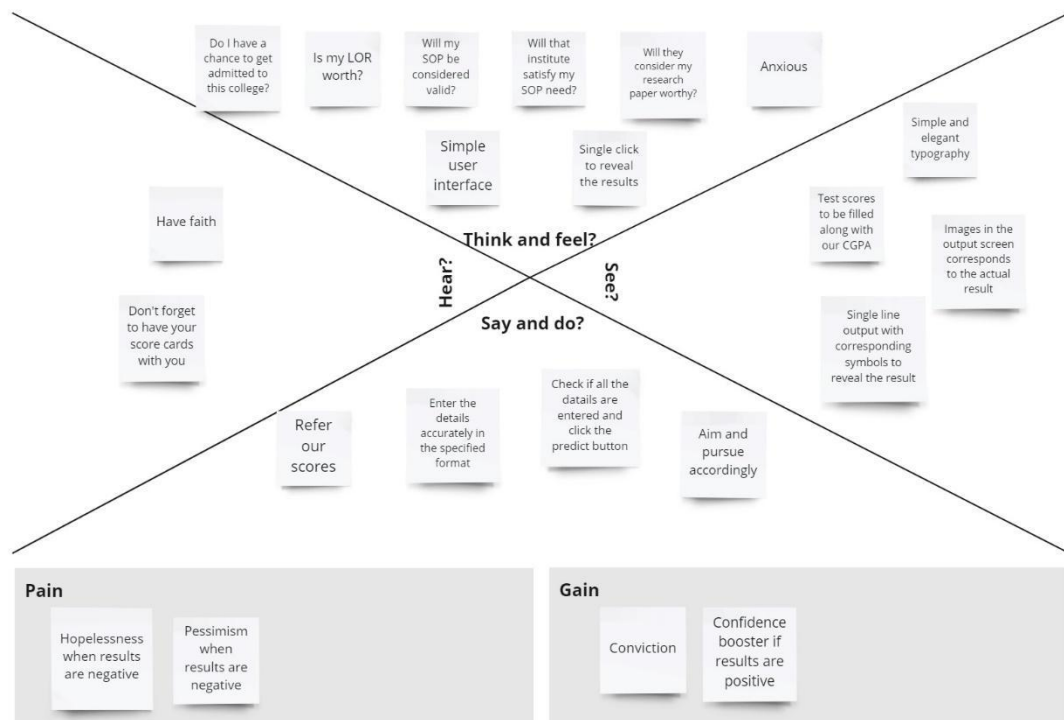
<https://ieeexplore.ieee.org/document/7924718>
<https://ieeexplore.ieee.org/document/9213205>
https://www.researchgate.net/publication/341740217_ENGINEERING_TECHNOLOGY_ADMISSION_ANALYSIS_AND_PREDICTION
<https://www.jncet.org/Manuscripts/Volume-8/Issue-4/Vol-8-issue-4-M-32.pdf>

3. Problem Statement Definition

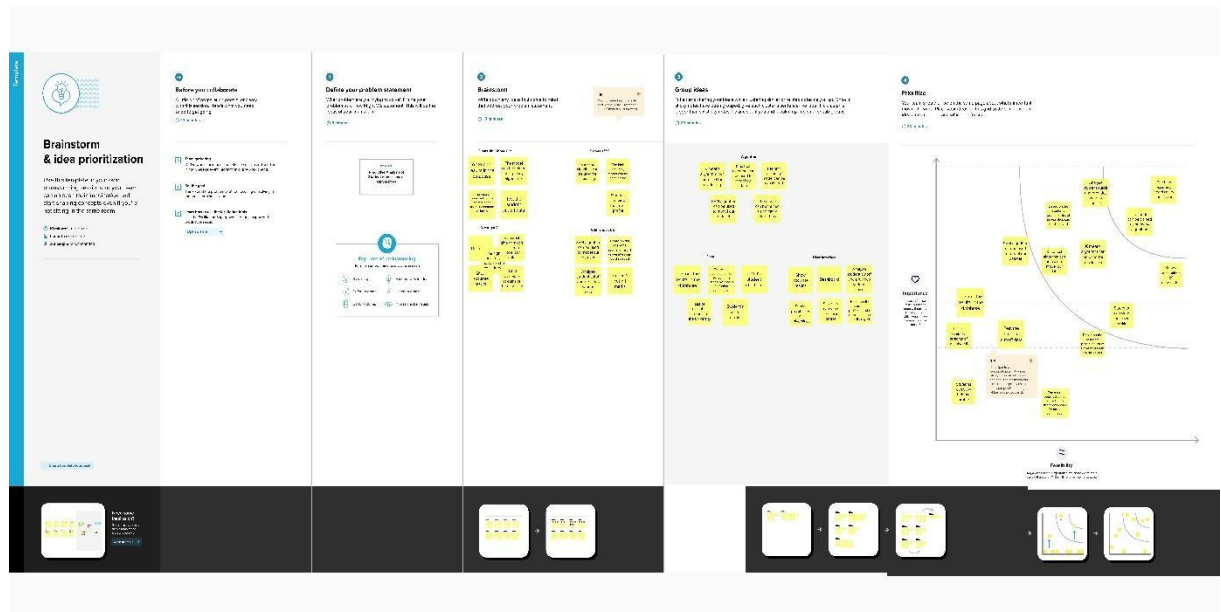


3. IDEATION & PROPOSED SOLUTION

1. Empathy Map Canvas



2. Ideation & Brainstorming



3. Proposed Solution

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Most of the people especially from the rural area are not that much aware of the standards which has been followed in various universities. At the time of completion of their higher secondary, they are having various stream willingness but not everybody is getting into the actual stream they have been wished. Same in the case of college too.
2.	Idea / Solution description	The aim of our project is to bring a new platform for the students who are in the phase of university admission, to predict the college in which they can get in to the stream they have been decided and wished already based on their performances in their academics includes cut-offs and quota. Here we are bringing the output which has good percentage of accuracy.
3.	Novelty / Uniqueness	In our University Admit Eligibility Predictor, student can able to get the complete insights about all the possible colleges and branches based on their cut-offs and quota. It will be like the practise session for them before attending the counselling conducted by various universities.

4.	Social Impact / Customer Satisfaction	Our project let the students to know about the possible colleges and streams based on their cut-offs and quota. So, it will be very helpful for them in their counselling processes.
5.	Business Model (Revenue Model)	We can have two models for revenue, one is the subscription model. In here user will be asked to get the subscription in order to get output from our prediction. Subscription may be monthly or weekly. Actually, this model is for admission centres. The next revenue model is pay per each prediction model, here the individual has to pay for each prediction. This model has been designed for the individual

4. Problem Solution fit

Project Title: University Admit Eligibility Predictor		Project Design Phase-I - Solution Fit		Team ID: PNT2022TMID21562	
Define CS, fit into	1. CUSTOMER SEGMENT(S) Who is your customer? i.e. working parents of 0-5 y.o. kids Students are our primary users other than students the persons or the organizations who is giving career guidance also using our predictor.	6. CUSTOMER CONSTRAINTS What constraints prevent your customers from taking action or limit their choices of solutions? i.e. spending power, budget, no cash, network connection, available devices. Customer constraints includes 1. Poor network connectivity. 2. System with very low processing speed. 3. Improper data feeding.	5. AVAILABLE SOLUTIONS Which solutions are available to the customers when they face the problem or need to get the job done? What have they tried in the past? What pros & cons do these solutions have? i.e. pen and paper is an alternative to digital notetaking 1. Some websites are available in the internet in order to predict the universities but they are not even 50 percent accurate. 2. Can able to predict the college and the stream by the own manual calculation.	Explore AS.	
	2. JOBS-TO-BE-DONE / PROBLEMS Which jobs-to-be-done (or problems) do you address for your customers? There could be more than one; explore different sides. Predicting the possible colleges and streams for the students who have been completed their higher secondary with higher percent of accuracy.	9. PROBLEM ROOT CAUSE What is the real reason that this problem exists? What is the back story behind the need to do this job? i.e. customers have to do it because of the change in regulations. 1. Unawareness about the possible colleges and streams based on the cut-offs. 2. Unavailability of high accurate university prediction model	7. BEHAVIOUR What does your customer do to address the problem and get the job done? i.e. directly related: find the right solar panel installer, calculate usage and benefits; indirectly associated: customers spend free time on volunteering work (i.e. Greenpeace) Students who have been completed their higher secondary has to feed their academic data into our university predictor, then only they will come to know about the possible colleges and streams based on the fed data.		
Focus on J&P, tap into BE, understand	3. TRIGGERS What triggers customers to act? i.e. seeing their neighbor installing solar panels, reading about a more efficient solution in the news. 1. Plans about their graduation and degree. 2. Confusions in college and stream selection 3. Low seats availability 3. EMOTIONS: BEFORE / AFTER How do customers feel when they face a problem or a job and afterwards? i.e. lost, insecure > confident, in control - use it in your communication strategy & design Before: Students are totally confused with their selection and they are so stresses too. After: Students can able to select their willing one confidently by seeing all other possibilities with full satisfaction.	10. YOUR SOLUTION If you are working on an existing business, write down your current solution first, fill in the canvas, and check how much it fits reality. If you are working on a new business proposition, then keep it blank until you fill in the canvas and come up with a solution that fits within customer limitations, solves a problem and matches customer behaviour. Developing the user-friendly prediction model with higher percentage of accuracy by considering the historical data and by having various algorithms.	8. CHANNELS of BEHAVIOUR 8.1 ONLINE What kind of actions do customers take online? Extract online channels from #7 Need to feed the data about their academic performance includes cut-offs and quota 8.2 OFFLINE What kind of actions do customers take offline? Extract offline channels from #7 and use them for customer development. Do calculate manually by having all the required academic details by comparing it with the historical data.		
	Identify strong TR &				

4. REQUIREMENT ANALYSIS

1. Functional requirement

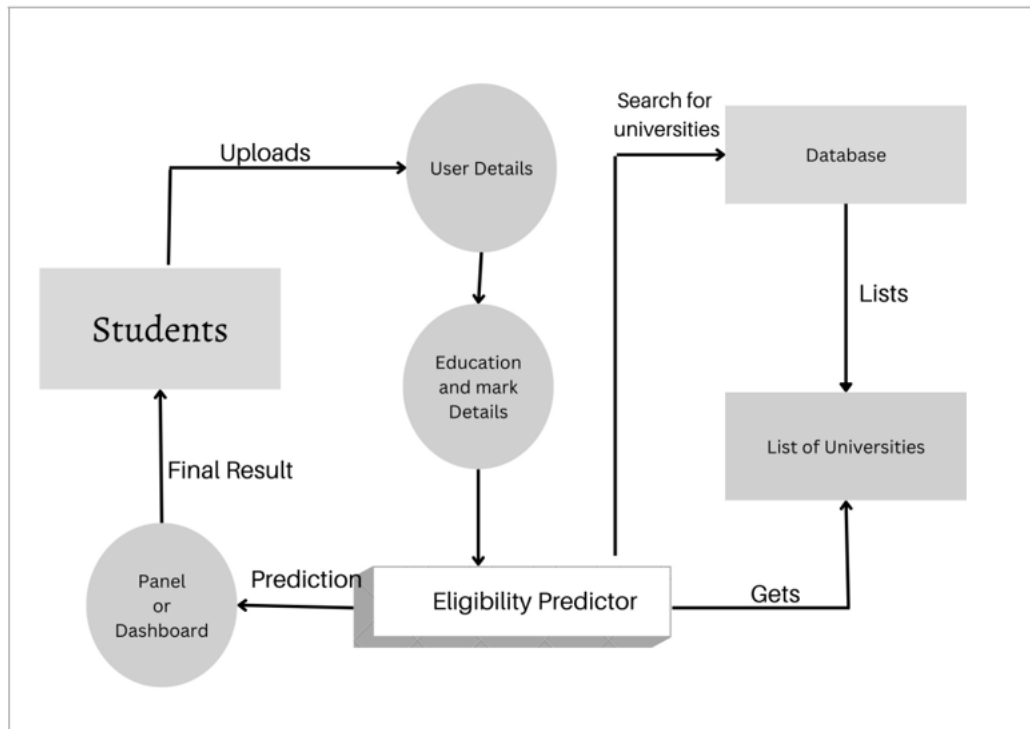
FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through Gmail Registration through LinkedIn
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	User Requirements	All the needed files are been asked to feed in the website. By having the file, it will do all the pre-processing and shows all the required information to the student(user). The information includes the list of all the possible universities and streams.
FR-4	User Details	Has to feed some documents <ol style="list-style-type: none">1. Score Sheets2. Letter of Recommendation (LOR)3. Statement of Purpose (SOP)4. Curriculum Vitae (CV)

2. Non-Functional requirements

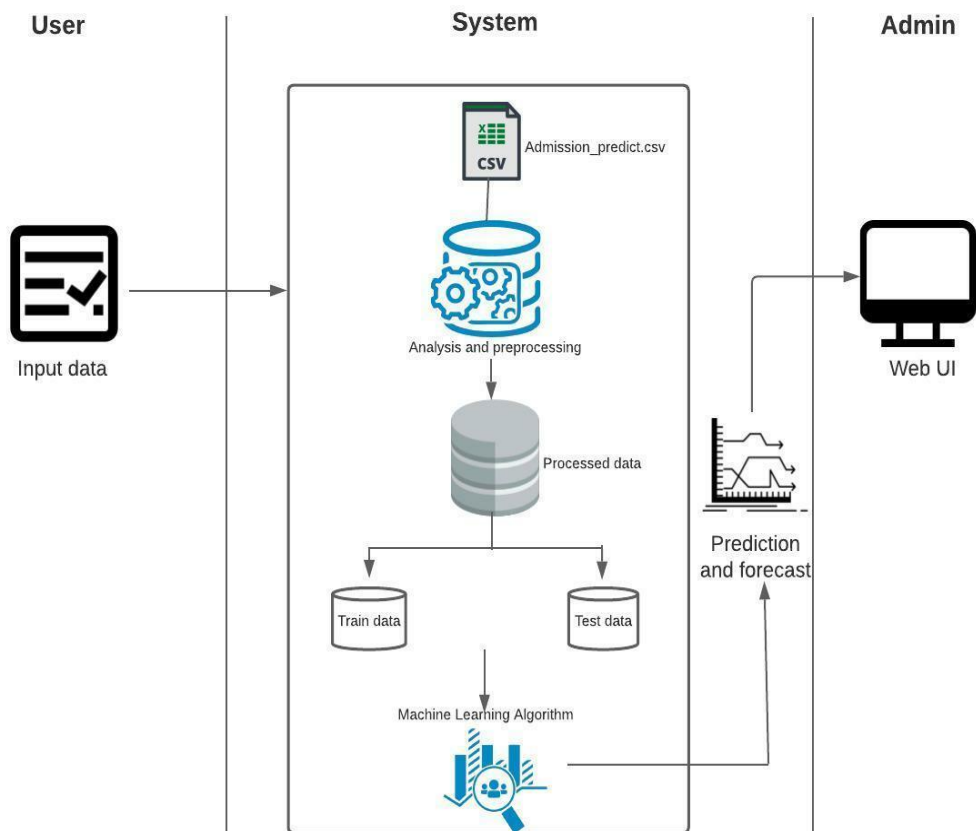
FR No.	Non-Functional Requirement	Description
NFR-1	Usability	<ul style="list-style-type: none">● Our website is very user friendly so even the layman can able to access our website.● There is no need for any pre requisite technical skill in order to access our website.● Each and every content of the page will be in synchronous way. Thus, it will not take much time to refresh or reload.
NFR-2	Security	<ul style="list-style-type: none">● The user who is having the valid credentials can able to access our site.● Data they are feeding into our website will not be accessed by any one of them.
NFR-3	Reliability	<ul style="list-style-type: none">● Our website is more reliable. Since nobody can able to see the data fed by the user.● The user can get the result with higher percent of accuracy.
NFR-4	Performance	<ul style="list-style-type: none">● User can able to handle the process in our website even by having internet connection

5. PROJECT DESIGN

1. Data Flow Diagrams



2. Solution & Technical Architecture



3. User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Student)	Dashboard	USN-1	As a user, I can view the cut off marks of previous years in my dashboard	I can access and download the files	High	Sprint-1
		USN-2	As a user, I can view university details and their rankings	I can only view(read-only)	Medium	Sprint-1
		USN-3	As a user, I can review the experience of the students in the university	I can access the review sections	Medium	Sprint-2
		USN-4	As a user, I can upload my documents	I have read and write access to upload files	High	Sprint-1
		USN-5	As a user, I can fill out the general and educational details in the form provided	I have read and write access to the forms filled	High	Sprint-2
	Predictor	USN-6	I can view the list of universities in which I am eligible to get an admission	I can receive the final result as whether eligible or not	High	Sprint-2
		USN-7	I can view the list of universities I am eligible with the same cut-off but in previous years	I can access the files with read-only permission	Medium	Sprint-2
Administrator	Dashboard	USN-8	As an administrator,I can have access to update the latest updates of the universities	I can have access to read and write the university information in the dashboard	High	Sprint-3
		USN-9	As an administrator,I can access any resources available in the page	I can access the resources that are available	Medium	Sprint-3
		USN-10	As an administrator,I can have a track on the universities the student is eligible to get admission at	I can access the list of the universities obtained as final result	High	Sprint-3

6. PROJECT PLANNING & SCHEDULING

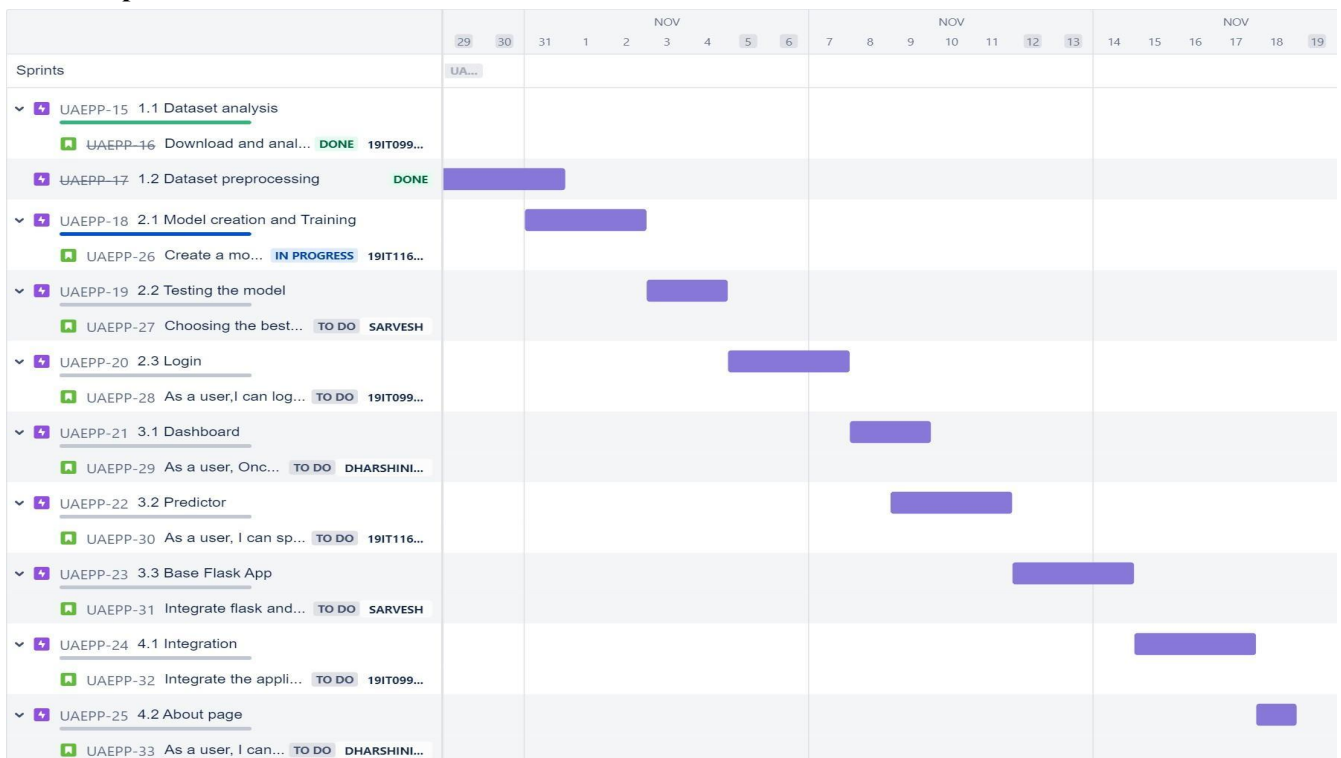
1. Sprint Planning & Estimation

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Dataset Analysis	USN-1	Download the admission_predict dataset and analyze it.	2	High	Sowmya V
	Dataset preprocessing	USN-2	Examine the dataset and perform preprocessing steps	1	Medium	Dharshini Shree C Y
Sprint-2	Model Creation and Training	USN-3	Create a model from the training data	2	High	Vibhisheak L S
	Testing the model	USN-4	Choosing the best model from the obtained accuracy	2	Medium	Sarvesh V S
	Login	USN-5	As a user, I can log into the application by entering email & password	1	High	Sowmya V
Sprint-3	Dashboard	USN-6	As a user, once I log in, I can view the Admission Prediction page	1	High	Dharshini Shree C Y
	Predictor	USN-7	As a user, I can specify all the values for prediction and get accurate results	1	High	Vibhisheak L S
	Base Flask App	USN-8	Integrate Flask and the built model	2	High	Sarvesh V S
Sprint-4	Integration	USN-9	Integrate the app on IBM cloud	2	High	Sowmya V
	About page	USN-10	As a user, I can know about the predictor system	1	Medium	Dharshini Shree C Y

2. Sprint Delivery Schedule

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

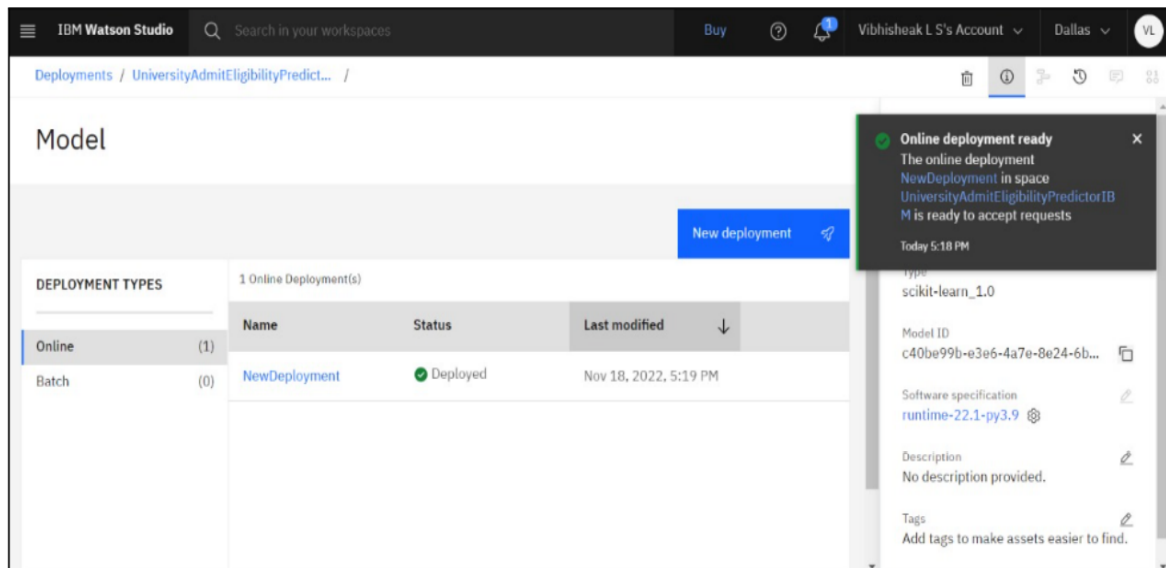
3. Reports from JIRA



7. CODING & SOLUTIONING

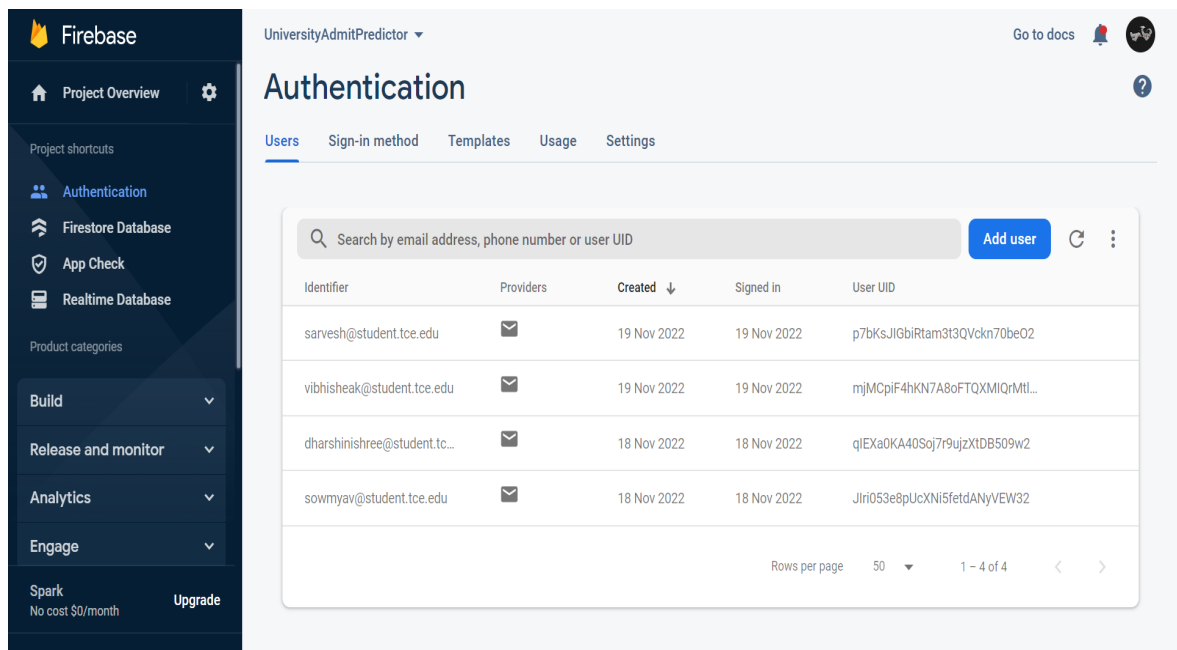
1. Feature 1

The model has been deployed using IBM Watson Machine learning service.



2. Feature 2

Login and registration page for users has been created using firebase authentication service.



3. Database Schema

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 400 entries, 0 to 399
Data columns (total 8 columns):
#   Column                Non-Null Count  Dtype
---  -
0   GRE Score              400 non-null    int64
1   TOEFL Score            400 non-null    int64
2   University Rating      400 non-null    int64
3   SOP                    400 non-null    float64
4   LOR                    400 non-null    float64
5   CGPA                   400 non-null    float64
6   Research                400 non-null    int64
7   Chance of Admit        400 non-null    float64
dtypes: float64(4), int64(4)
memory usage: 25.1 KB
```

An additional column is created based on the chance of admit column values which changes the continuous value to categorical value giving Yes/No output predictions.

Changing the target column from continuous variable to categorical variable.If the chance of admit is greater than 80, then the probability is yes. Else the result is no									
<pre>df['result']=pd.cut(df.Chance_of_Admit_,bins=[0,0.80,1],labels=['No','Yes'])</pre>									
df									
	GRE_Score	TOEFL_Score	University_Rating	SOP	LOR_	CGPA	Research	Chance_of_Admit_	result
0	337	118	4	4.5	4.5	9.65	1	0.92	Yes
1	324	107	4	4.0	4.5	8.87	1	0.76	No
2	316	104	3	3.0	3.5	8.00	1	0.72	No
3	322	110	3	3.5	2.5	8.67	1	0.80	No
4	314	103	2	2.0	3.0	8.21	0	0.65	No
...
395	324	110	3	3.5	3.5	9.04	1	0.82	Yes
396	325	107	3	3.0	3.5	9.11	1	0.84	Yes
397	330	116	4	5.0	4.5	9.45	1	0.91	Yes
398	312	103	3	3.5	4.0	8.78	0	0.67	No
399	333	117	4	5.0	4.0	9.66	1	0.95	Yes

8. TESTING

1. Test Cases

Test case ID	Feature Type	Component	Test Scenario	Prerequisite	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Comments	TC for Automation (Y/N)	BUG ID	Executed by
Login Page_TC_01	UI	Login Page	Tested with all the UI components and input fields in the home page.	HTML, CSS	Need to give URL first Check with all the elements in the UI that it is displayed or not.		All the UI components and the input fields have to be function properly	Working as expected	Pass		N		Dhars hini Shree C Y
Login Page_TC_02	Functional	Login Page	Verifying whether the user has been authenticated to enter	HTML, CSS	Need to click on the predict or tab in the navigation bar. Check with all the elements in the UI that it is		All the UI components has to be function properly and users	Working as expected	Pass		N		Sarve sh V S

			into our predictor page.		displayed or not.		has to navigate to the predictor page properly						
Predict_Page_TC_003	UI	Predict Page	Verify whether all the UI elements in the Prediction page are functioning properly or not.	HTML, CSS, Flask	Check with all the elements in the UI that it is displayed or not		All the UI components has to be functioning properly	Working as expected	Pas s		N		Sowmya V
PredictPage_TC_004	Functional	Predict Page	Do enter the values in the input fields and click on predict	Flask	Need to enter 7 values for each attributes and click on predict.	2001004449 No Research	Navigate to the prediction page and do predict the accurate result	Working as expected	Pas s		N		Vibhishak L S

OutputPage_TC_005	Functional	Chance Page	Verify whether it is navigates to chance page only if the appropriate values are entered.	Flask	Need to enter 7 values for each attributes and click on predict. If prediction equals one, chance page is displayed	Prediction = You have a chance	Do navigate to chance page	Working as expected	Pass		N		Dharsini Shree C Y
OutputPage_TC_006	Functional	No chance Page	Checking whether it is redirected to no chance page or not		Need to enter the URL Need to enter the values and click on predict. It has to be redirected to no chance page if the predictor result is zero	Prediction = You don't have a chance	Do redirect to no chance page	Working as expected	Pass		N		Sowmya V

2. User Acceptance Testing

1. Purpose of Document

The purpose of this document is to briefly explain the test coverage and open issues of the University Admit Eligibility Predictor project at the time of the release to User Acceptance Testing (UAT).

2. Defect Analysis

This report shows the number of resolved or closed bugs at each severity level, and how they were resolved

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	11	4	2	3	20
Duplicate	1	0	3	0	4
External	3	3	0	1	7
Fixed	10	2	4	20	36
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won't Fix	0	5	2	1	8
Totals	25	14	13	26	78

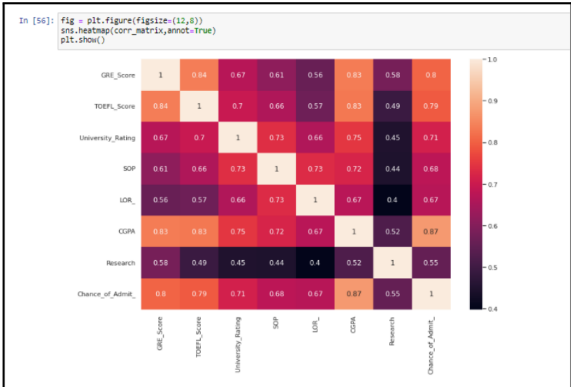
3. Test Case Analysis

This report shows the number of test cases that have passed, failed, and untested

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	7	0	0	7
Client Application	50	0	0	50
Security	2	0	0	2
Outsource Shipping	2	0	0	2
Exception Reporting	9	0	0	9
Final Report Output	4	0	0	4

9. RESULTS

1. Performance Metrics

S.No.	Parameter	Values	Screenshot																																																																																	
1.	Metrics	<p>Regression Model:</p> <p>MAE - , MSE - , RMSE - , R2 score -</p> <p>Classification Model:</p> <p>Confusion Matrix - , Accuracy Score- & Classification Report -</p>	<div><pre>In [118]: from sklearn.metrics import r2_score,mean_squared_error,mean_absolute_error from math import sqrt RMSE=float(format(np.sqrt(mean_squared_error(y_test,rf_y_pred)))) MAE=mean_absolute_error(y_test,rf_y_pred) MSE=mean_squared_error(y_test,rf_y_pred) R2=r2_score(y_test,rf_y_pred) print('RMSE :', RMSE, '\nMSE :', MSE, '\nMAE :',MAE, '\nR2 score:',R2) RMSE : 0.25 MSE : 0.0625 MAE : 0.0625 R2 score: 0.6865203761755486</pre></div> <div><pre>In [55]: corr_matrix = df.corr() corr_matrix</pre><p>Out[55]:</p><table><thead><tr><th></th><th>GRE_Score</th><th>TOEFL_Score</th><th>University_Rating</th><th>SOP</th><th>LOR_</th><th>CGPA</th><th>Research</th><th>Chance_of_Admit_</th></tr></thead><tbody><tr><th>GRE_Score</th><td>1.000000</td><td>0.835977</td><td>0.668976</td><td>0.612831</td><td>0.557555</td><td>0.833060</td><td>0.580391</td><td>0.802610</td></tr><tr><th>TOEFL_Score</th><td>0.835977</td><td>1.000000</td><td>0.695590</td><td>0.657981</td><td>0.567721</td><td>0.828417</td><td>0.488858</td><td>0.791594</td></tr><tr><th>University_Rating</th><td>0.668976</td><td>0.695590</td><td>1.000000</td><td>0.734523</td><td>0.660123</td><td>0.748479</td><td>0.447783</td><td>0.711250</td></tr><tr><th>SOP</th><td>0.612831</td><td>0.657981</td><td>0.734523</td><td>1.000000</td><td>0.729593</td><td>0.718144</td><td>0.444029</td><td>0.675732</td></tr><tr><th>LOR_</th><td>0.557555</td><td>0.567721</td><td>0.660123</td><td>0.729593</td><td>1.000000</td><td>0.670211</td><td>0.396859</td><td>0.669889</td></tr><tr><th>CGPA</th><td>0.833060</td><td>0.828417</td><td>0.748479</td><td>0.718144</td><td>0.670211</td><td>1.000000</td><td>0.521654</td><td>0.873289</td></tr><tr><th>Research</th><td>0.580391</td><td>0.488858</td><td>0.447783</td><td>0.444029</td><td>0.396859</td><td>0.521654</td><td>1.000000</td><td>0.553202</td></tr><tr><th>Chance_of_Admit_</th><td>0.802610</td><td>0.791594</td><td>0.711250</td><td>0.675732</td><td>0.669889</td><td>0.873289</td><td>0.553202</td><td>1.000000</td></tr></tbody></table></div> <div><pre>In [56]: fig = plt.figure(figsize=(12,8)) sns.heatmap(corr_matrix,annot=True) plt.show()</pre></div> <div><pre>In [116]: from sklearn.ensemble import RandomForestClassifier from sklearn.metrics import accuracy_score rf = RandomForestClassifier() trained_model = rf.fit(X_train, y_train) In [117]: import pickle pickle.dump(rf, open('university.pkl','wb')) In [118]: rf_y_pred = rf.predict(X_test) rf_accuracy=accuracy_score(rf_y_pred,y_test) print("Accuracy : ",rf_accuracy*100,"%") Accuracy : 93.75 %</pre></div>		GRE_Score	TOEFL_Score	University_Rating	SOP	LOR_	CGPA	Research	Chance_of_Admit_	GRE_Score	1.000000	0.835977	0.668976	0.612831	0.557555	0.833060	0.580391	0.802610	TOEFL_Score	0.835977	1.000000	0.695590	0.657981	0.567721	0.828417	0.488858	0.791594	University_Rating	0.668976	0.695590	1.000000	0.734523	0.660123	0.748479	0.447783	0.711250	SOP	0.612831	0.657981	0.734523	1.000000	0.729593	0.718144	0.444029	0.675732	LOR_	0.557555	0.567721	0.660123	0.729593	1.000000	0.670211	0.396859	0.669889	CGPA	0.833060	0.828417	0.748479	0.718144	0.670211	1.000000	0.521654	0.873289	Research	0.580391	0.488858	0.447783	0.444029	0.396859	0.521654	1.000000	0.553202	Chance_of_Admit_	0.802610	0.791594	0.711250	0.675732	0.669889	0.873289	0.553202	1.000000
	GRE_Score	TOEFL_Score	University_Rating	SOP	LOR_	CGPA	Research	Chance_of_Admit_																																																																												
GRE_Score	1.000000	0.835977	0.668976	0.612831	0.557555	0.833060	0.580391	0.802610																																																																												
TOEFL_Score	0.835977	1.000000	0.695590	0.657981	0.567721	0.828417	0.488858	0.791594																																																																												
University_Rating	0.668976	0.695590	1.000000	0.734523	0.660123	0.748479	0.447783	0.711250																																																																												
SOP	0.612831	0.657981	0.734523	1.000000	0.729593	0.718144	0.444029	0.675732																																																																												
LOR_	0.557555	0.567721	0.660123	0.729593	1.000000	0.670211	0.396859	0.669889																																																																												
CGPA	0.833060	0.828417	0.748479	0.718144	0.670211	1.000000	0.521654	0.873289																																																																												
Research	0.580391	0.488858	0.447783	0.444029	0.396859	0.521654	1.000000	0.553202																																																																												
Chance_of_Admit_	0.802610	0.791594	0.711250	0.675732	0.669889	0.873289	0.553202	1.000000																																																																												
2.	Tune the Model	Hyperparameter Tuning - Validation Method -	<div><pre>In [128]: from sklearn.linear_model import LogisticRegression from sklearn.model_selection import GridSearchCV c_space = np.logspace(-5, 8, 15) param_grid = {'C': c_space} logreg = LogisticRegression() logreg_cv = GridSearchCV(logreg, param_grid, cv = 5) logreg_cv.fit(X_train, y_train) print("Tuned Logistic Regression Parameters: {}".format(logreg_cv.best_params_)) print("Best score is {}".format(logreg_cv.best_score_)) Tuned Logistic Regression Parameters: {'C': 0.4393970560760795} Best score is 0.9</pre></div>																																																																																	

10. ADVANTAGES & DISADVANTAGES

ADVANTAGES:

- It gives an overall accuracy of 94%, which is really high.
- The dataset consists of all possible attributes needed for prediction.
- Confidence booster if results are positive.
- Students can also change their scores to see how they affect the overall prediction results and focus more on that area.

DISADVANTAGES:

- The model is built in such a way that the prediction is positive only if the chance of admit percent is greater than 80%. Even if the result is 79%, the prediction would be NO.
- The complexity of the examinations is not considered. Therefore the results may vary every year for the same set of attribute.
- The dataset used for training the model is of comparatively small size. Therefore, the model cannot be relied on to take accurate real-time decisions.

11. CONCLUSION

The dataset is trained with different ML model. The ML models used to train our dataset are KNN, Logistic Regression, Random Forest, SVM each having accuracy of 86%, 88%, 93%, 89% respectively. Random forest algorithm is finally selected to be used in our model. The Machine Learning model is integrated using flask for our web application. At long last, understudies can have an open-source AI model which will assist the understudies with knowing their opportunity of entrance into a specific college with high exactness

12. FUTURE SCOPE

A real-time project can be developed by gathering data from institutions. The data can be processed and trained using big data frameworks like spark and MLLib can be used to train the data using different machine learning models.

13. APPENDIX

Source Code

App.py

```
from flask import Flask, request, jsonify, render_template, redirect, url_for
import requests
import json
import pickle
model = pickle.load(open('university.pkl', 'rb'))
import pyrebase
```

```

API_KEY = "w7wZ3NDUKJjLg9ulwEFwDCKCnOurNNLrzp3gZ-SNrbGO"
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}

config = {
    "apiKey": "AIzaSyBLcGYGA82pCAHW4xKjgDYv_bsnEJEgo1E",
    "authDomain": "universityadmitpredictor.firebaseio.com",
    "projectId": "universityadmitpredictor",
    "storageBucket": "universityadmitpredictor.appspot.com",
    "messagingSenderId": "938493164189",
    "databaseURL":
"https://console.firebase.google.com/u/0/project/universityadmitpredictor/database/
universityadmitpredictor-default-rtdb/data/~2F",
}

firebase = pyrebase.initialize_app(config)
auth = firebase.auth()

model = pickle.load(open('university.pkl', 'rb'))

app = Flask(__name__)

@app.route('/', methods=['GET', 'POST'])
def homepage():
    if request.method == 'POST':
        unsuccessful = 'Please check your credentials'

        email = request.form['name']
        password = request.form['pass']

        try:
            auth.sign_in_with_email_and_password(email, password)
            return render_template('index.html')
        except:
            auth.create_user_with_email_and_password(email,password)
            auth.sign_in_with_email_and_password(email, password)
            return render_template('index.html')

    return render_template('login.html')

@app.route('/predict', methods=['GET', 'POST'])
def predict():
    if request.method == 'POST':
        gre = request.form['gre']
        toefl = request.form['toefl']
        universityNumber = request.form['universityNumber']
        sop = request.form['sop']
        lor = request.form['lor']
        cgpa = request.form['cgpa']
        research = request.form['research']

        y_pred = [[gre, toefl, universityNumber, sop, lor, cgpa, research]]

        payload_scoring = {"input_data": [
            {"field": ["GRE Score", "TOEFL Score", "University Rating", "SOP",
"LOR ", "CGPA", "Research"]}],

```



```

        "values": y_pred}}
    response_scoring =
requests.post('https://us-south.ml.cloud.ibm.com/ml/v4/deployments/67f91885-c382-4d
94-9b23-60bbc3f65a47/predictions?version=2022-11-18',json=payload_scoring,headers={
'Authorization': 'Bearer ' + mltoken})
    print("Scoring response")
    print(response_scoring.json())
    predictions = response_scoring.json()
    output = predictions['predictions'][0]['values'][0][0]
    print(output)

    if output == 'Yes':
        return render_template('chance.html')
    if output == 'No':
        return render_template('Nochance.html')
    return render_template('index.html')

@app.route('/index.html',methods=['GET', 'POST'])
def index():
    return render_template('index.html')

@app.route('/about.html')
def about():
    return render_template('about.html')

if __name__ == '__main__':
    app.run(debug=True)

```

Model.py

```

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import pickle

data=pd.read_csv('C:/Users/Lenovo/OneDrive/Documents/Nalaiya Thiran/University
Admit Eligibility Predictor/dataset/Admission_predict.csv')
data.drop(["Serial No."],axis=1,inplace=True)
df = pd.DataFrame(data)
df.columns = df.columns.str.replace(' ', '_')

df['result']=pd.cut(df.Chance_of_Admit_,bins=[0,0.80,1],labels=['No','Yes'])

independent = data.iloc[:,0:7].values
dependent = data.iloc[:,8:].values

from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(independent, dependent,
random_state=0, train_size = .2)

from sklearn.ensemble import RandomForestClassifier
rf = RandomForestClassifier()
trained_model = rf.fit(X_train, y_train)

pickle.dump(rf, open('university.pkl','wb'))

```

Login.html

```
<!DOCTYPE html>
<html>
<head>
  <style>

  div.static{
    position: absolute;
    top: 250px;
    left:900px;
  }

  </style>
  <h2 style="color:black"><font size="+5"><b>UNIVERSITY ADMISSION PREDICTION
SYSTEM</b></font></h2>

  <title></title>
  <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css"
integrity="sha384-BVYiiSIFeK1dGmJRAkycuHAHRg32OmUcww7on3RYdg4Va+PmSTsz/K68vbdEjh4u"
crossorigin="anonymous">

  <!-- Optional theme -->
  <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap-theme.min.css"
integrity="sha384-rHyOnLiRsVXV4nD0JutlnGaslCJuC7uwjduW9SVrLvRYooPp2bWYgmgJQIXwl/Sp"
crossorigin="anonymous">

  <!-- Latest compiled and minified JavaScript -->
  <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"
integrity="sha384-Tc5IQib027qvyjSMfHjOMaLkfuWVxZxUPnCJA7l2mCWNIpG9mGCD8wGNlcpdTxa"
crossorigin="anonymous"></script>
</head>
<body>
  <div class="imgcontainer">
    
  </div>
  <div class="container">
    {% if s %}
      <div class="alert alert-success">
        <h2>{{s}}</h2>
      </div>
    {% endif %}

    {% if us %}
      <div class="alert alert-danger">
        <h2>{{us}}</h2>
      </div>
    {% endif %}
    <div class="static">
      <form action="{{url_for('homepage')}}" method="post">
        <h2>Sign in</h2><br>
        <font size="+1"><label for="inputEmail">Email address</label></font>
        <input type="email" id="inputEmail" name="name" placeholder="Email
address" required autofocus>
        <br><br><font size="+1"><label for="inputPassword">Password
</label></font>
```

```

        <input type="password" id="inputPassword" name="pass"
placeholder="Password" required>
        <div class="checkbox">
            <label>
                <input type="checkbox" value="remember-me"> Remember me
            </label>
        </div>
        <button style="background-color:black; color:white; width:100%"
type="submit">Sign in</button>

    </form>
</div>
</div>
<footer></footer>
</body>
</html>

```

Index.html

```

<!DOCTYPE html>
<html>

    <body
background="https://oge.tmu.edu.tw/admission/apply/graduation-hat-with-degree-paper
-on-a-stack-of-book/">
        <head>
            <style>
                body {
                    background-repeat: no-repeat;
                    background-attachment: fixed;
                    background-size: 100% 100%;
                }

                form,h2,h3 {
                    padding-top: 10px;

                    padding-left: 100px;
                    padding-right: 40px;
                }
            </style>
            <title>University Admit Eligibility Predictor</title>

        </head>
        <h2 style="color:blue"><font size="+2">UNIVERSITY ADMISSION PREDICTION
SYSTEM</font></h2>
        <h3><font size="+1">Enter your details and get probability of your
admission</font></h3>
        <form action="{url_for('predict')}"method="POST">
            <br>
            <font size="+1">Enter GRE score <input id="gre" type="number" name="gre"
required="required"></font>
            <br><br>
            <font size="+1">Enter TOEFL score <input id="toefl" type="number"
name="toefl" required="required"></font>
            <br><br>
            <font size="+1">Select University number</font><br>
            <input id="universityNumber" type="radio" id="1" name="universityNumber"
value="1">
            <label for="1">1</label><br>
            <input type="radio" id="2" name="universityNumber" value="2">

```

```

        <label for="2">2</label><br>
        <input type="radio" id="3" name="universityNumber" value="3">
        <label for="3">3</label><br>
        <input type="radio" id="4" name="universityNumber" value="4">
        <label for="4">4</label><br>
        <input type="radio" id="5" name="universityNumber" value="5">
        <label for="5">5</label><br>
        <br>
        <font size="+1">Enter SOP <input id="sop" type="number" name="sop" min="1"
max="5" required="required"></font>
        <br><br>
        <font size="+1">Enter LOR <input id="lor" type="number" name="lor" min="1"
max="5" required="required"></font>
        <br><br>
        <font size="+1">Enter CGPA <input id="cgpa" type="number" name="cgpa"
required="required"></font>
        <br><br>
        <font size="+1">Research</font><br>
        <input id="research" type="radio" id="researchDone" name="research"
value="1">
        <label for="research">Research</label><br>
        <input id="noresearch" type="radio" name="research" value="0">
        <label for="noresearch">No Research</label><br>
        <br>
        <button type="submit" value="Predict"><font
size="+1">Predict</font></button>
    </form>

    <h2><a href="about.html">Click here to know more about us...</a></h2>
</body>
</html>

```

Chance.html

```

<!DOCTYPE html>
<html>
<head>
    <style>

    div.static{
        position: absolute;
        top: 100px;
        left:100px;
    }

    </style>
</head>
<body>

<div class="static">
<h2>Predicting Chance of Admission</h2>
    <p>Prediction: <b ><u>You have a chance</u> </b><b
style='font-size:40px;'>128077</b> </p>
</div>



```

```
</body>
</html>
```

Nochance.html

```
<!DOCTYPE html>
<html>
<head>
  <style>

  div.static{
    position: absolute;
    top: 100px;
    left:100px;
  }
  img{
    position: absolute;
    top: 150px;
    right:200px;
  }

  </style>
</head>
<body>

<div class="static">
<h2>Predicting Chance of Admission</h2>
  <p>Prediction: <b ><u>You don't have a chance</u> </b><b
style='font-size:40px;'>#128078;</b> </p>
</div>



</body>
</html>
```

GitHub link: <https://github.com/IBM-EPBL/IBM-Project-34841-1660277975>

Project Demo Link: <https://youtu.be/nDXrbuxJznI>