



A

MINI-PROJECT REPORT

ON

“Evocator: College Admission Predictor”

Submitted in partial fulfillment of the requirement for the award of

Bachelor of Engineering

In

Computer Science and Engineering

Punyashlok Ahilyadevi Holkar Solapur University

By

Sandeep Gone	TECSE - 39
Anekant Madrap	TECSE - 40
Prathamesh Manakapure	TECSE - 41
Ritika Kurapati	TECSE - 42
Priyanka Dandi	TECSE - 43

Under Guidance Of
Mrs. K. A. Khedikar



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
WALCHAND INSTITUTE OF TECHNOLOGY
SOLAPUR - 413006
(2021-2022)



CERTIFICATE

This is to certify that the Mini-Project entitled
“Evocator: College Admission Predictor”

Is

Submitted By

Sandeep Gone	39
Anekant Madrap	40
Prathamesh Manakapure	41
Ritika Kurapati	42
Priyanka Dandi	43

(Prof. Mrs. K. A. Khedikar)
Project Guide

(Dr. Mrs.A.M.Pujar)
Head
Dept of Computer Sc. & Engg

(Dr. V .A. Athavale)
Principal

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
WALCHAND INSTITUTE OF TECHNOLOGY
SOLAPUR
(2021-2022)

INDEX

Sr No.	Topic	Page No.
1	Abstract	4
2	Introduction	5
3	Background	6
4	Technologies Used	7
5	Description and Working of Project	9
6	Screenshots & Results	12
7	Advantages and Disadvantages or Applications	16
8	Future scope	17
9	Overall Project Cost	17
10	Conclusion/Summary	18
11	References	19

Abstract

College Admission Predictor System is a web-based application system in which students can register their marks along with their personal information. This helps to predict their admissions in colleges. It helps students to make right decisions for choosing their college. In which students can register with their details as well as marks details for prediction of admission in colleges.

Sometimes the students are seeking the admission to the college for she or he are not eligible as per the merit of that college. So here we built predictive model to guide the students about their admissibility in the desired college & also suggest the college where they will get the admission. So, to achieve this objective we may include machine learning capabilities that allow to improve their performance based on experience, just as humans do.

A machine learning model is built in order to provide results. The dataset contains historical MHT-CET cutoff percentile of various universities with branches. Various algorithms have been used to train the model for given dataset and predict results. Algorithms used are Random Forest, Decision Tree Classifier. The model analyzes student academic merits, category and college admission criteria. Based on that, it predicts the likelihood of a university college that a student may enter. Our system primarily focuses on building an efficient and user-friendly web application.

Introduction

After HSC examination, most of the students choose engineering to make their career. For engineering admission, students have to give MHT-CET entrance exam, which plays vital role in admission process. During admission process, students have to give their preferable colleges list. Many of students make mistake while listing college due to various reasons like inaccurate analysis of colleges, lack of knowledge, inappropriate guidance, increasing competition etc. This leads to various issues like a student may not get dream college, allotted a college which is not affordable, college may have no core placements or few placements which makes student to struggle more to get placed in industry.

The main principle behind the need of College Admission Prediction system is to make better choices of college before allotment. It can handle the details of students such as merit details. This Student Database has been designed that holds student first name, last name, username, password, branch, category.

Admissions in engineering colleges in the state of Maharashtra or any state is based upon common entrance test (CET) and since more than 1.5lakh seats are to be allotted in more than 200 engineering colleges and over 35 different branches of engineering , for students belonging to many categories like open, home university, outside home university, reserved category(SC,ST, OBC etc.) the problem becomes more serious and students struggle to understand which colleges they are likely to get admitted in, even after going through cut-off data of previous years. Many students fill wrong Options and fail to get admission. To minimize the stress of students we came up with the idea of a computer aided method which will help the students get the list of all colleges in which they could get the admission at the click of a button.

Background

Several research papers have been studied for development of college admission predictor system.

A. Predictive Analytics Models for Student Admission and Enrollment

In this paper, they have developed analytical model for a local university based on historical built on neural networks, decision trees and logistic regression. However, this model cannot be self-determined and only assists to compliment university administrators' decision-making process to manage admissions and enrolment.

B. An Automated Prediction Model for College Admission System

In this paper, they have done the necessary research using K-CET (Karnataka Common Entrance Test) data. Our system is developed taking K-CET into consideration. Use data analytic techniques to prepare a preference list based on the user's input. The preference list varies depending on user input.

C. Prediction for University Admission using Machine Learning

In this paper, they created a Machine Learning model which could be used by students who want to pursue their education in the US, they considered only few universities with different rankings.

D. Hybrid Recommender System for Predicting College Admission

In this paper, this system consists of two cascade hybrid recommenders working together with the help of college predictor. Students it uses previous students admission data of colleges GPA for predicting most probable colleges. It looks over student academic merits, background, student records, and the college admission criteria. Then, predicts the possibility of university colleges that a student may enter. The design is proposed only of Saudi Arabian Universities.

E. CAPSLG: College Admission Predictor and Smart List Generator

In this paper, the Ensemble AdaBoost Classifier from the scikit-learn library of Python is used for classification of the data. The model is trained for the individual colleges under the Mumbai region. The project includes application of machine learning algorithm on the data-set of cut-off lists of colleges for past 3 years.

Technologies Used

HTML: Hyper Text Markup Language is the standard markup language, used for creating Web application and describing the structure of our Web application. It consists of a series of elements, which tell the browser how to display the content. HTML code ensures the proper formatting of text and images for our Internet browser. Without HTML, a browser would not know how to display text as elements or load images or other elements.

CSS: Cascading Style sheet is the language for describing the presentation of our Web application including colors, layout, and fonts. It allows one to adapt the presentation to different types of devices, such as large screens, small screens. CSS offers several significant advantages over alternative approaches to web design.

Python: Python is open source, interpreted, high level language and provides great approach for object-oriented programming and widely used by data scientist for various data science application. It provides great functionality to deal with mathematics, statistics and scientific function. For implementing machine learning model python has been used.

JavaScript: JavaScript is a light weight object-oriented programming language and is one of the core technologies. Developers can use JavaScript to fetch data from other sources and display it on their own site. We used java script to create interactive web content and for scripting our web pages.

Django: Django is a high-level Python web framework that enables rapid development of secure and maintainable websites. Django follows MVT (Model View Template) architecture. It is used for developing clean web application. For Simplicity and Scalability Django is used develop web application.

PostgreSQL: PostgreSQL is a free and open-source relational database management system. It is used as the primary data store for many web, mobile applications. It has the richest set of features that are supported by Django. It is faster when dealing with massive datasets, complicated queries, and read-write operations. PostgreSQL is used to deal with user data.

Machine Learning Algorithms: Machine learning algorithms can be applied on dataset for cost savings, improved time, and performance. machine learning techniques can handle large and complex data to draw interesting patterns or trends in them such as anomalies. Machines are needed to process information fast and make decisions when it reaches the threshold. There are many machine learning algorithms that help to do better data analysis. We used two different types of algorithms.

Description and Working of Project

Main Objectives of our application are

- To help students to fill their preferences during admission process.
- To ease of making better choices of college before allotment.
- To help students by giving appropriate college list as per their percentile.
- To deploy a web application for college admission predictor system.

System Architecture for our proposed methodology is

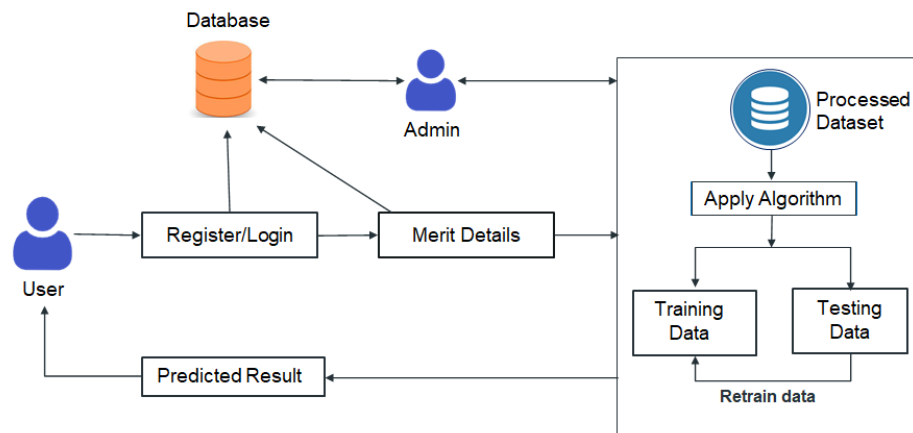


Fig 1. System Architecture

Main tools used are Django and Machine Learning to develop college admission predictor. Django is open-source python web framework used to develop User interface of website. Initially user has to register and login with valid credentials. For registration user's First name, Last name, City, Email Id and password are required fields, these details are stored in database and fetched from database while login. User have to give valid details in case of invalid details Error message is given to give valid credentials.

For database management PostgreSQL is used. PostgreSQL is the primary data store or data warehouse for many web, mobile, geospatial, and analytics applications. PostgreSQL is an advanced, enterprise-class, and open-source relational database system and supports both SQL (relational) querying.

After login, Merit and other details required for prediction are provided. Merit details include HSC percentage, CET percentage, Math marks and whether user is of PCM (Physics, Chemistry, Math) or PCMB (Physics, Chemistry, Math, Biology) stream. If a student has HSC percentage less than 50% (less than 45% for reserved/ backward category/ Persons with Disability) then student will not be eligible for Engineering hence in that case proper message is shown.

UML Use case Diagram:

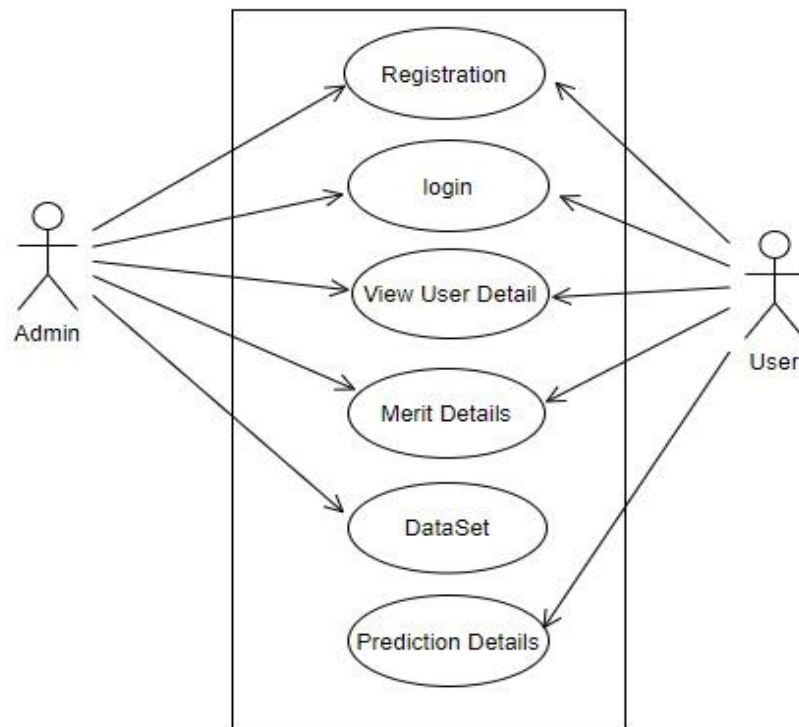


Fig 2. Use case Diagram

In Machine Learning model, we two different machine algorithms have been implemented:

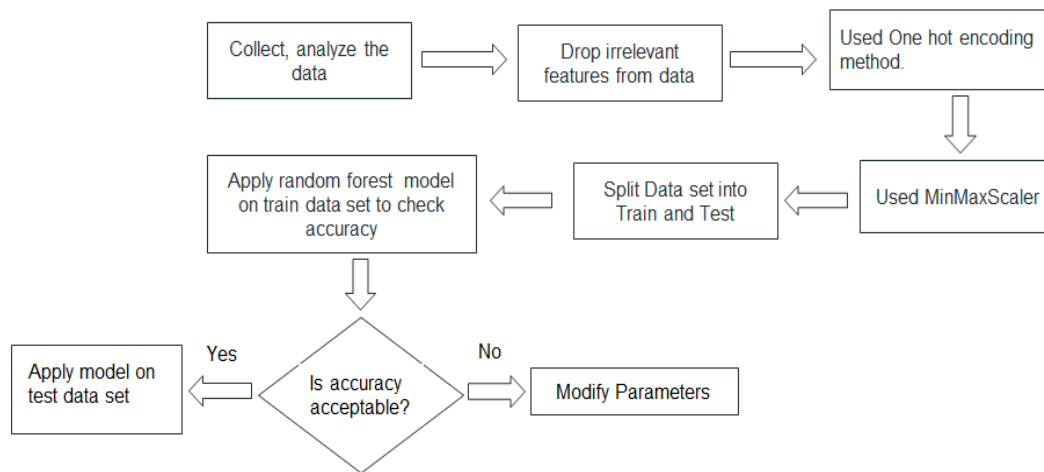


Fig 3. Methodology of our project

1. Decision Tree: A decision tree is non-parametric supervised learning. It is used for both classification and regression problems. It is a flowchart-like structure in which each internal node represents a “test” on an attribute, each branch represents the outcome of the test, and each leaf node represents a class label. The path between root and leaf represents classification rules.

2. Random Forest: Random Forest builds decision trees on different samples and takes their majority vote for classification and average in case of regression. It is an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time. For classification tasks, the output of the random forest is the class selected by most trees. It provides an effective way of handling missing data. It can produce a reasonable prediction without hyper-parameter tuning. It solves the issue of overfitting in decision trees. In every random forest tree, a subset of features is selected randomly at the node’s splitting point.

About the Dataset: To achieve all the objectives we need past admission data of multiple Engineering colleges to work on. So, we have collected various University’s engineering colleges admission record. As we are focusing on MHT-CET so we collected only the colleges of Maharashtra state. This Dataset has various attributes, which are: Sr. No., College Name, College Code and with different Category, Branch.

Data pre-processing: It is an important task in machine learning. It converts raw data into clean data. Missing Value are those values that failed to load information or the data itself was corrupted.

There are different techniques to handle missing values. One of which we have applied is deleting rows because some of the rows were blank. Handling Categorical Data (Label Encoder) – This is one of the most frequently used techniques for the categorical variable. Label encoder converts labels into a numeric format so that the machine can recognize it. In our data, there are two categorical columns like Category, Branch.

Model Implementation: After cleaning all the data, removing all the noise, selecting relevant features and encoded it into machine learning form, the next step is building a predictive model by applying various ML techniques to find out the best model which gives us more accuracy for train and test both. But before that, we must split our data into 2 parts.

Train-Test Split: The training data set is used to create the model while testing the data set is used to check the performance. Training data's output is available to model while test data is unseen data. So, in our data, we have split data into 70% for training data and 30% for testing data because it makes the classification model better.

Comparison of Result (Actual Vs Fitted): we have used 2 standard algorithms for prediction of college which includes Decision Tree, Random Forest. The chosen appropriate algorithms run through test harness of number of splits 30 and performance major accuracy.

After observing accuracy results of the model, we can simply understand the accuracy of classifier Decision Tree & Random Forest are seen more practical. After finalizing classifier of the model sometimes we required tuning the model for better accuracy. Tuning is the method for increasing a model's performance without over fitting the data or making the variance too high. We have taken single record to classify the college & tested. For this single record we have again build 2 predictive model and tested for different input variable. So, we found every time that Random Forest always give great accuracy than Decision Tree.

Implementation:

Model.py

```
import pandas as pd
import numpy as np
from sklearn.ensemble import RandomForestClassifier
from sklearn.preprocessing import OneHotEncoder
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import MinMaxScaler
from sklearn.metrics import accuracy_score, recall_score, precision_score, f1_score
from sklearn.metrics import accuracy_score, recall_score, precision_score, f1_score

df=pd.read_csv('withbranch.csv')
df['category'] = df['category'].astype('category')
df['branch'] = df['branch'].astype('category')

enc=OneHotEncoder()
enc_df=pd.DataFrame(enc.fit_transform(df[['category', 'branch']]).toarray())
enc_df.columns=['obc','open','sc','st','vj','it','civil','cse','etc','mech']
df=df.drop(['category','branch'],axis=1)
final_df=pd.concat([enc_df,df],axis=1)
sc=MinMaxScaler()

X = final_df.iloc[:, 0:-1].values
y = final_df.iloc[:, -1].values
X_train, X_test, y_train, y_test = train_test_split(X, y, train_size=0.85, random_state=42)

clf_model = RandomForestClassifier()#(random_state=41, max_depth=15, n_estimators=30)
clf_model.fit(X_train,y_train)
Ry_predT = clf_model.predict(X_test)
Ry_predTe = clf_model.predict(X_train)
```

```
accuracy_score(Ry_predTe, y_train)
```

Evocator/Accounts/views.py

```
from django.http import HttpResponseRedirect
from django.shortcuts import render, redirect
from django.contrib import messages
from django.contrib.auth.models import User, auth
from joblib import load
```

```
model = load('./savedModels/model.joblib')
```

```
def login(request):
    if request.method == 'POST':
        username = request.POST['username']
        password = request.POST['password']
        user = auth.authenticate(username=username, password=password)
        if user is not None:
            auth.login(request, user)
            print("login successful")
            return render(request, 'home.html')
        else:
            messages.info(request, "Invalid credentials")
            return render(request, 'login.html')
    else:
        return render(request, 'login.html')
```

```
def register(request):

    if request.method == 'POST':
        first_name = request.POST['first_name']
        last_name = request.POST['last_name']
```

```

username = request.POST['username']
email = request.POST['email']
password1 = request.POST['password1']
password2 = request.POST['password2']

if password1==password2:
    if User.objects.filter(username=username).exists():
        messages.info(request, "Username taken")
    elif User.objects.filter(email=email).exists():
        messages.info(request, "Email taken")
    else:
        user = User.objects.create_user(first_name=first_name, last_name=last_name,
username=username, email=email, password=password1)
        user.save()
        print('User created')
        return render(request, 'login.html')
    else:
        print("Password not matching")
else:
    return render(request, 'register.html')

def home(request):
    college = { 1: [1002, "Government College Of Engineering, Amravati", "Amravati"],
                2: [6273, "Bansilal Ramnath Agarawal Charitable Trust's Vishwakarma Institute of
Technology, Pune", "Pune"],
                3: [6141, "Jaywant Shikshan Prasarak Mandal's,Rajarshi Shahu College of Engineering,
Pune", "Pune"],
                4: [6207, "Dr .D. Y. Patil Institute of Engineering & Technology, Pimpri,Pune", "Pune"],
                5: [5108, "Nashik District Maratha Vidya Prasarak Samaj's Karmaveer Adv.Babaurao
Ganpatrao Thakare College of Engineering, Nashik", "Nashik"] }

    cet = float(request.GET.get('cet'))

```

```
branch = request.GET.get('branch')
category = request.GET.get('category')
bset=[0.0, 0.0, 0.0, 0.0, 0.0]
cset=[0.0, 0.0, 0.0, 0.0, 0.0]
```

```
if branch=='it':
    bset[0]=1.0
elif branch=='civil':
    bset[1]=1.0
elif branch=='cse':
    bset[2]=1.0
elif branch=='etc':
    bset[3]=1.0
else:
    bset[4]=1.0
```

```
if category=='obc':
    cset[0]=1.0
elif category=='open':
    bset[1]=1.0
elif category=='sc':
    bset[2]=1.0
elif category=='st':
    bset[3]=1.0
else:
    bset[4]=1.0
```

```
details = cset + bset
details.append(float(cet))
ypred = model.predict([details])
print(ypred)
pred = college[ypred[0]]
```



```
return render(request, 'result.html', {'code':pred[0], 'name':pred[1], 'city':pred[2]})
```

```
def logout(request):
```

```
    auth.logout(request)
```

```
    return render(request, 'index.html')
```

Evocator/urls.py

```
from django.contrib import admin
```

```
from django.urls import path, include
```

```
urlpatterns = [
```

```
    path("", include('myapp.urls')),
```

```
    path('admin/', admin.site.urls),
```

```
    path('accounts/', include('accounts.urls')),
```

```
]
```

Accounts/urls.py

```
from django.urls import path
```

```
from . import views
```

```
urlpatterns = [
```

```
    path("register", views.register, name='register'),
```

```
    path("login", views.login, name='login'),
```

```
    path("logout", views.logout, name='logout'),
```

```
    path("home", views.home, name='home'),
```

```
    path("result", views.home, name='result'), ]
```

Screenshots & Results

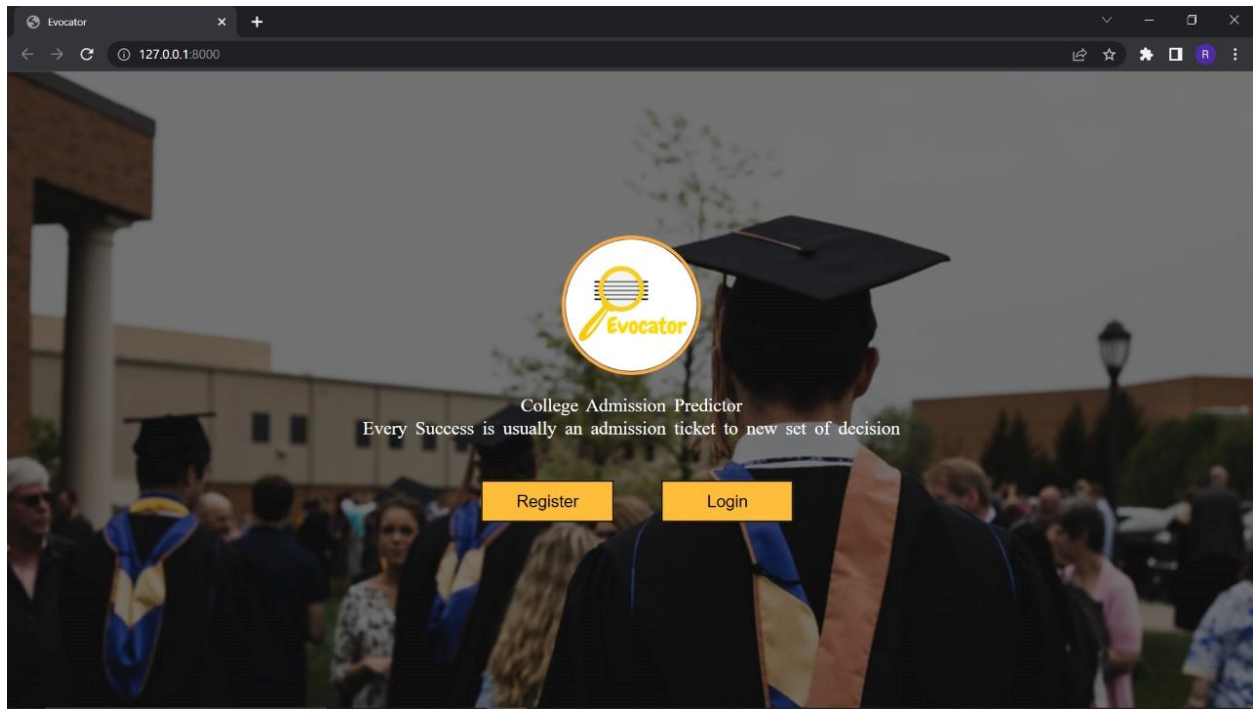


Fig 4. Home Page

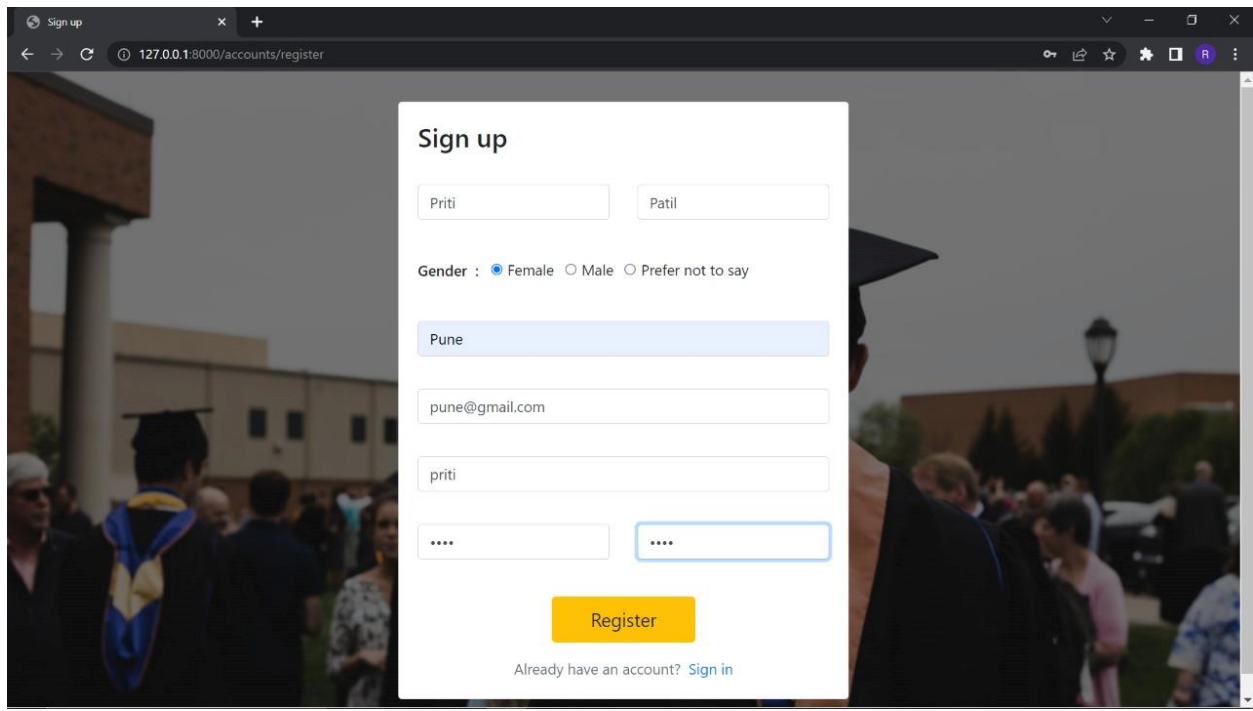


Fig 5. Sign up page

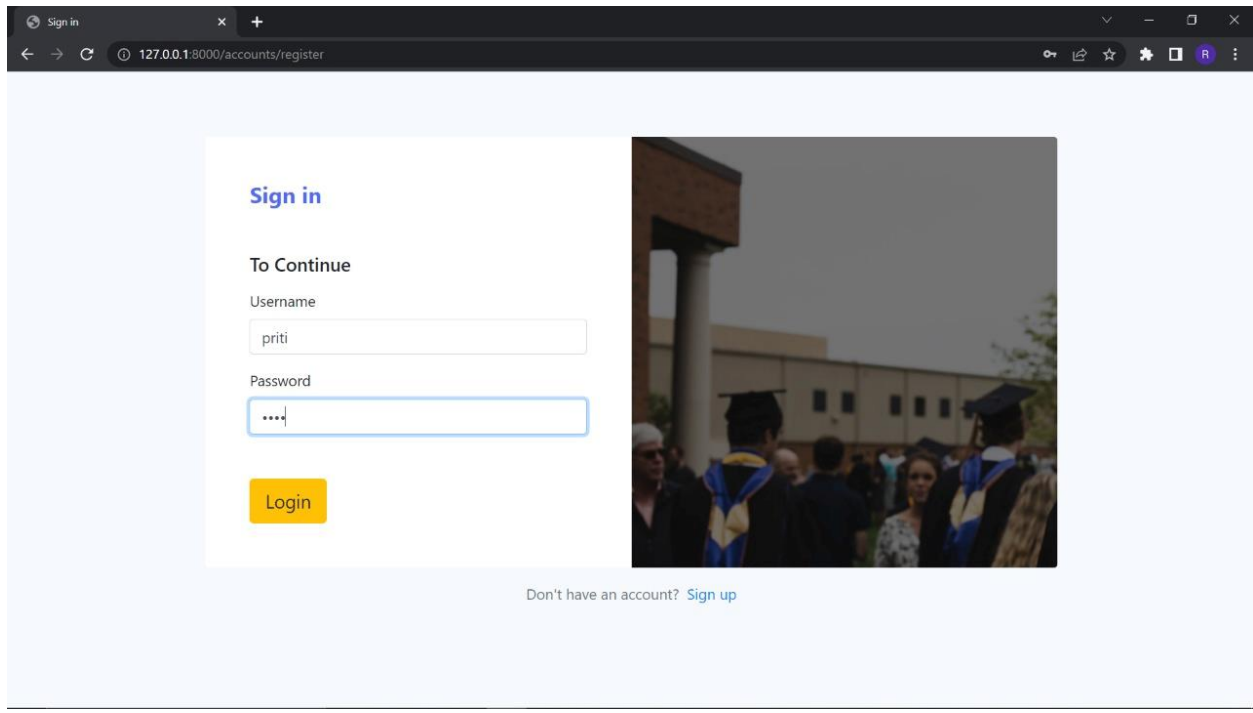


Fig 6. Sign in Page

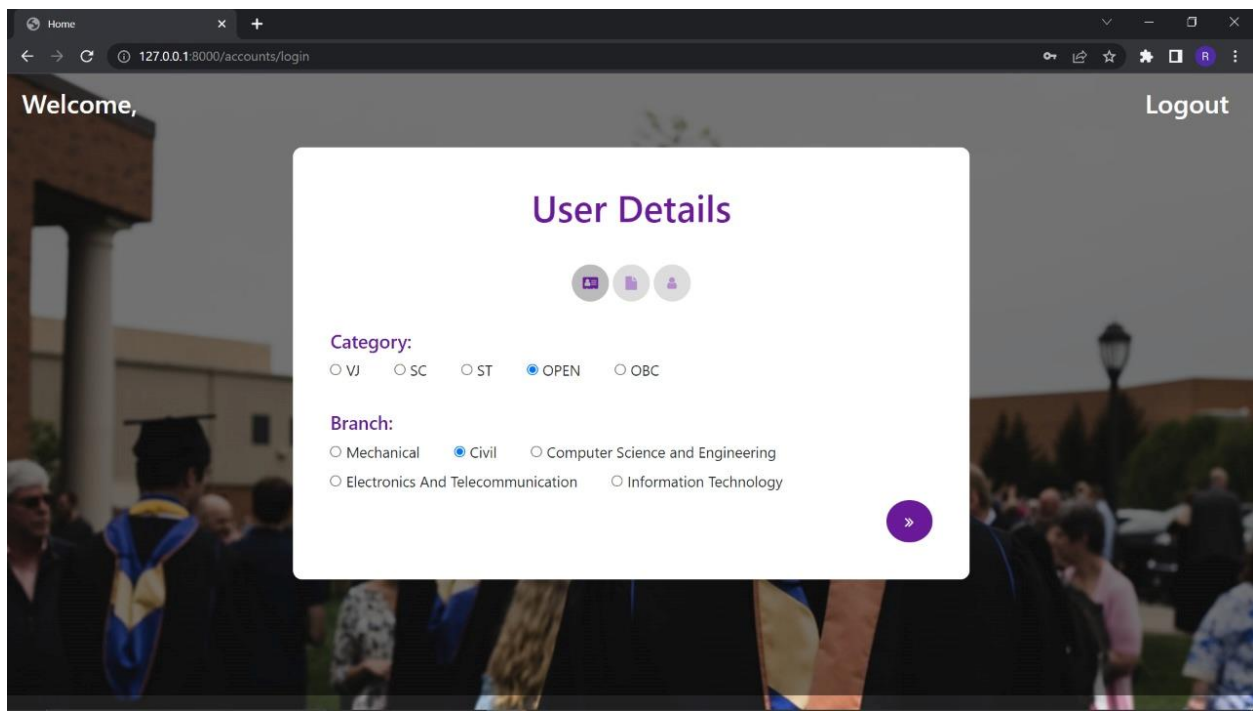


Fig 7.1 User Details

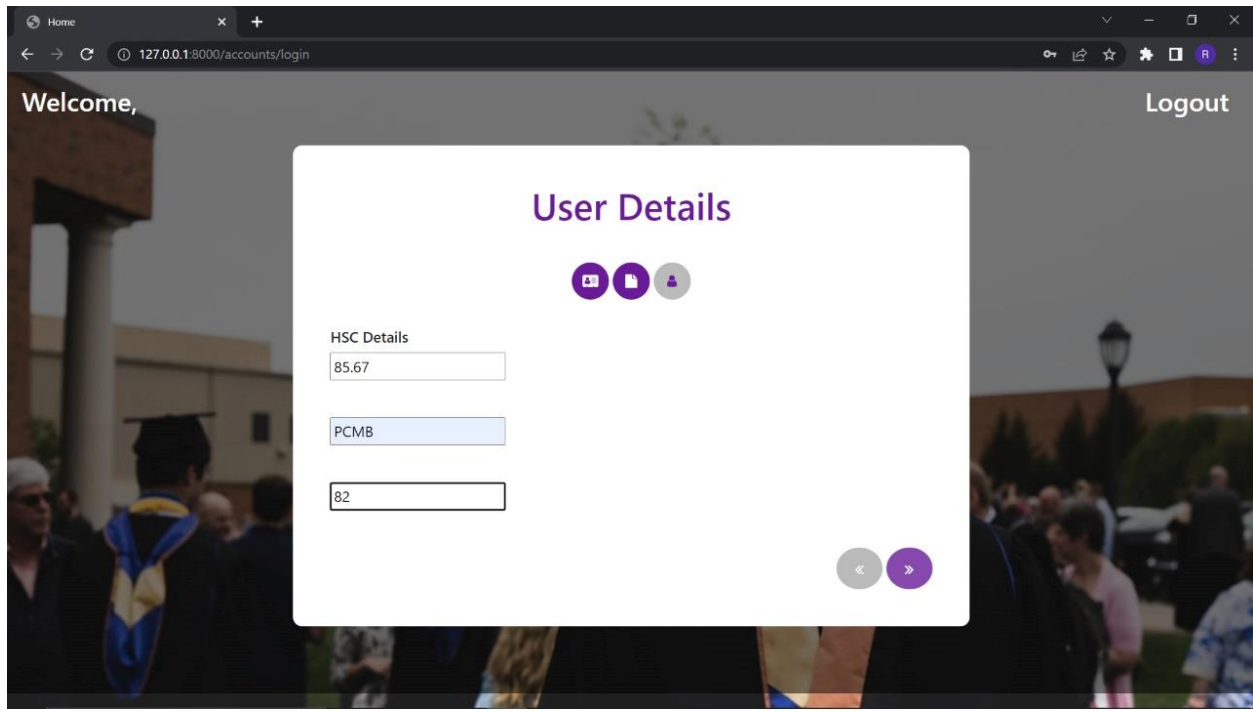


Fig 7.2 User Details

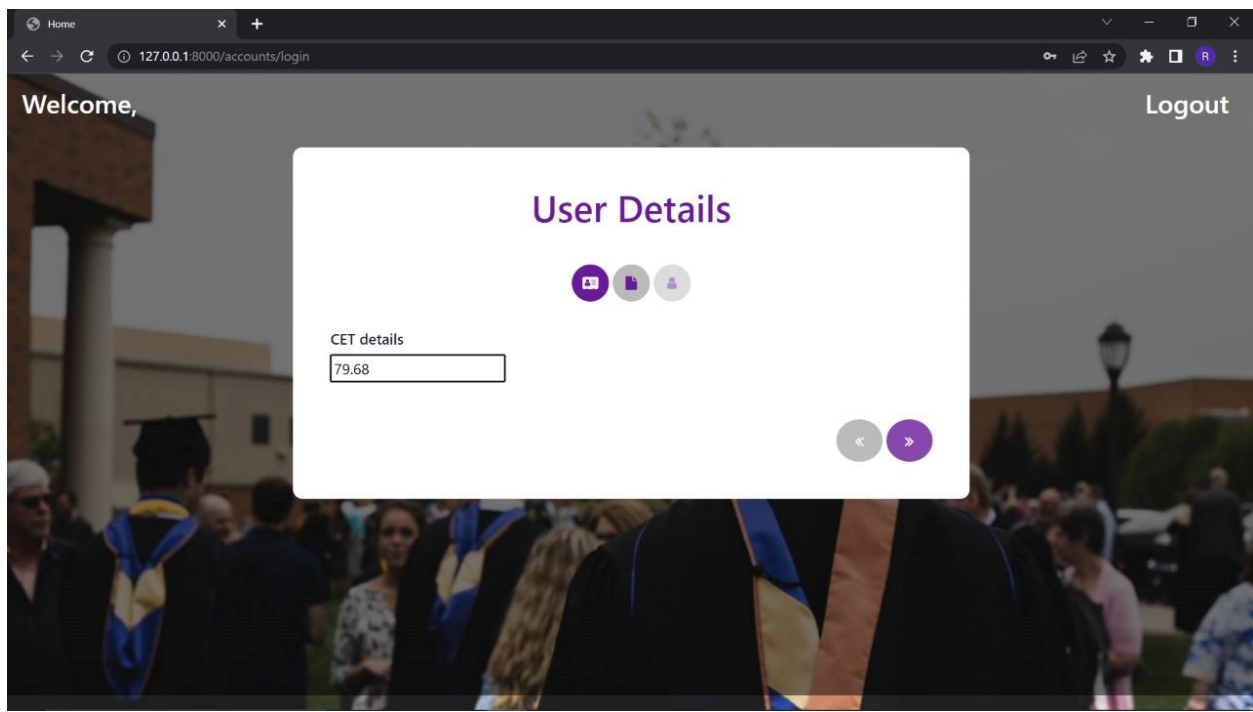


Fig 7.3 User Details

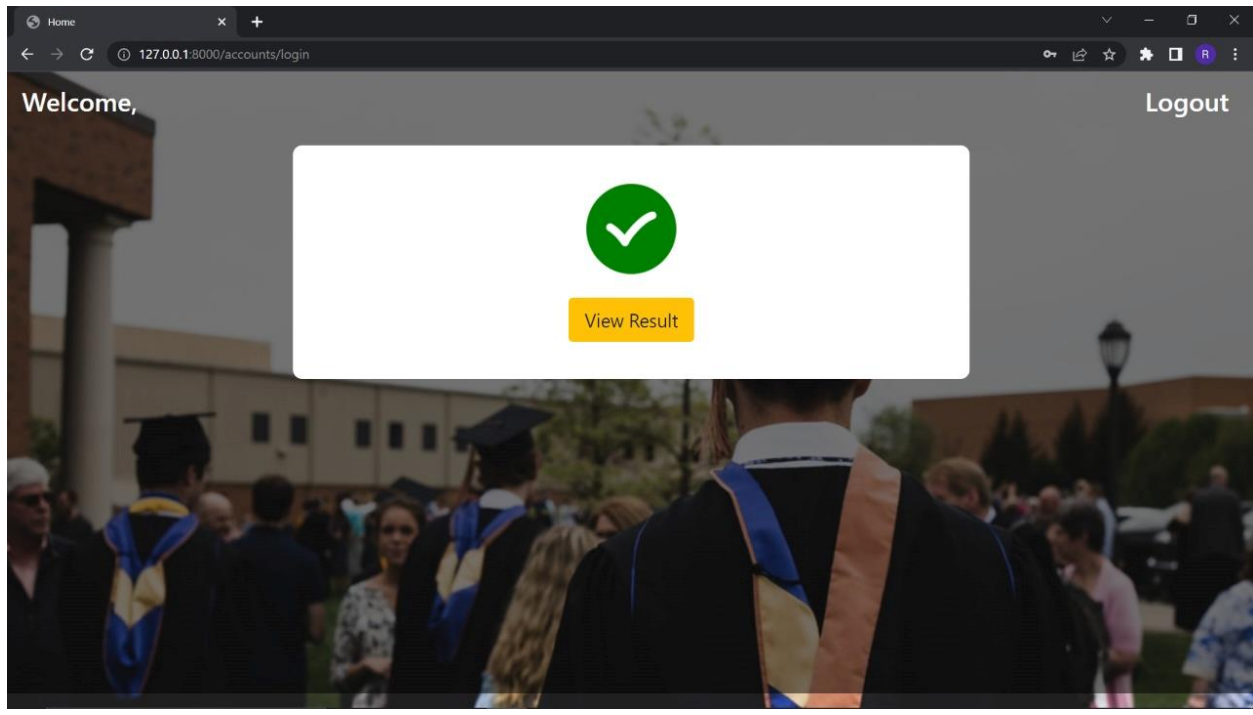


Fig 8.1 Prediction Result

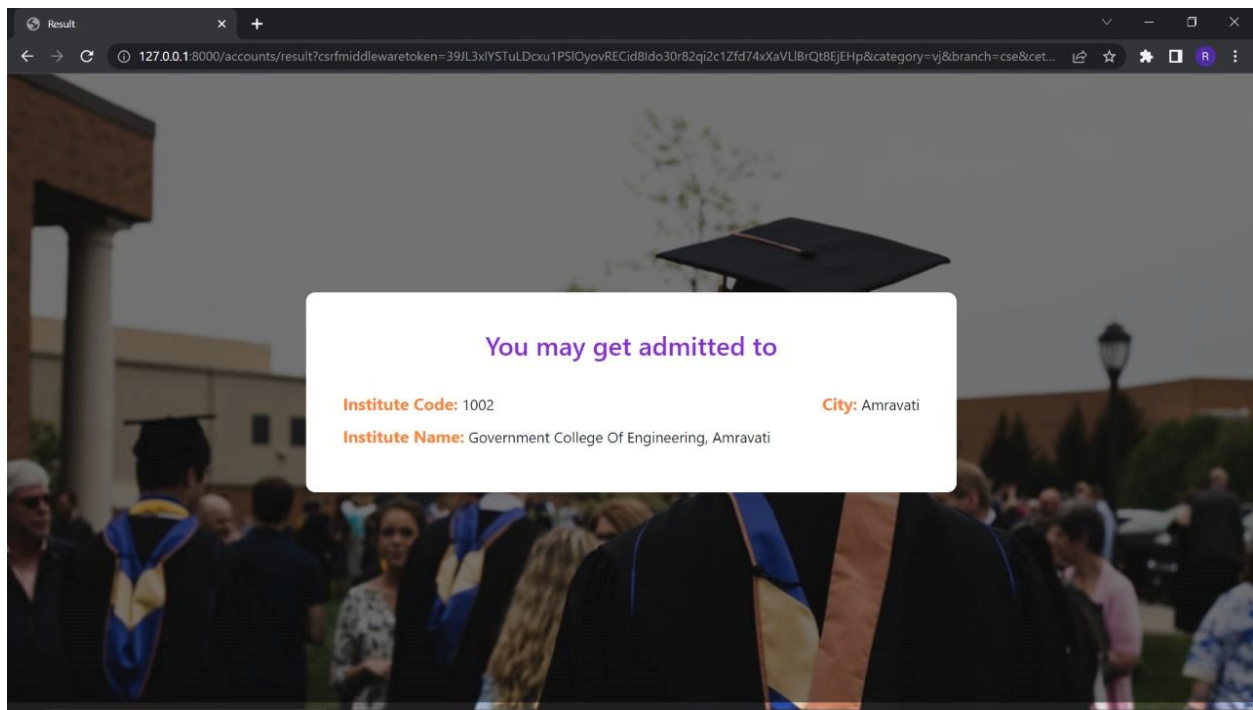


Fig 8.2 Prediction Result

Advantages and Disadvantages or Applications

❖ Advantages:

- It helps student for making decision for choosing a right college.
- This can be accessed anytime anywhere, since it is a web application provided only an internet connection.
- User-friendly system.
- Number of personnel required is considerably less.
- This gives an idea to students about their future college by calculating their cut off.
- When students come from rural places, they find it hard to go along with the formal procedures. So, this application helps them a lot and eases out their fear.

❖ Disadvantages:

- Required active internet connection.
- System will provide inaccurate results if data entered incorrectly.

❖ Applications:

- Can be used by various universities for allotting multiple students in various colleges.

Future scope

The future scope of this project are:

- Model is designed by considering only Maharashtra state so further it can modify for all other states.
- Further can be altered for predicting likely universities for other streams like Medical, Post graduation, Pharmacy etc.
- Can be used for predicting abroad universities based on different criteria and entrance exam score by changing the dataset.

Overall Project Cost

No cost

Conclusion

We have efficiently processed and implement the college predictor for the prediction of college which helps to predict your future college which is also move towards the right decision. The prediction of college is based on the MHT-CET percentile. The prediction is formed using ML algorithm. We applied different algorithm and choose best algorithm on the basis of its accuracy. Also, we collect the data of student for the further information which helps them.

We faced some problem when arranging data that sort and solved the issues and finalize Dataset, now we work with small amount of data and further we will increase it. The College admission prediction, compare cross-validation techniques for accuracy as performance and finally concluded the prediction of College for career after 12th in the engineering and technology.

References

- <https://www.bibliomed.org/mnsfulltext/218/218-1631562599.pdf?1651247842>
- <https://ieeexplore.ieee.org/document/6234369>
- <https://www.ijrte.org/wp-content/uploads/papers/v8i6/F9043038620.pdf>
- <https://towardsdatascience.com/creating-a-machine-learning-based-web-application-using-django-5444e0053a09>
- <https://www.geeksforgeeks.org/>
- <https://engineering.careers360.com/articles/mht-cet-cutoff>
- <https://towardsdatascience.com/top-machine-learning-algorithms-for-classification-2197870ff501>