



SMART UNIVERSITY ADMISSION PREDICTOR: A WEB-BASED CUTOFF ANALYSIS SYSTEM



PROJECT BASED LEARNING PROJECT REPORT

Submitted by

MANUTHRA M (720722104025)

PRAGATHEESH R A (720722104031)

ROHITH S (720722104034)

in partial fulfilment for the award of the degree of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

Hindusthan College of Engineering and Technology

Approved by AICTE, New Delhi, Accredited with 'A++' Grade by NAAC
(An Autonomous Institution, Affiliated to Anna University, Chennai) Valley
Campus, Pollachi Highway, Coimbatore – 641 032

MAY 2025



Hindusthan College of Engineering and Technology

Approved by AICTE, New Delhi, Accredited with 'A++' Grade by NAAC
(An Autonomous Institution, Affiliated to Anna University, Chennai)
Valley Campus, Pollachi Highway, Coimbatore – 641 032



BONAFIDE CERTIFICATE

Certified that this project report “**Smart University Admission Predictor: A Web-Based Cutoff Analysis System**” is the Bonafide work of “**MANUTHRA M (720722104025), PRAGATHEESH R A (720722104031), ROHITH S (720722104034)**” who carried out the project work under my supervision.

SIGNATURE

M. RAVIKUMAR, M.TECH., (Ph.D.)

SUPERVISOR

Assistant Professor

Computer Science and Engineering,
Hindusthan College of Engineering and
Technology, Coimbatore-32

SIGNATURE

Dr. S. SHANKAR, M.E., Ph.D.,

HEAD OF THE DEPARTMENT

Computer Science and Engineering,
Hindusthan College of Engineering
and Technology, Coimbatore-32

Submitted for the Autonomous Institution Mini Project Viva-Voce conducted on

INTERNAL EXAMINER

EXTERNAL EXAMINER

DECLARATION

We, hereby jointly declare that the project work entitled “ **Smart University Admission Predictor: A Web-Based Cutoff Analysis System**” , submitted to the Autonomous Institution Mini Project Viva voce - MAY 2025 in partial fulfilment for the award of the degree of “ **BACHELOR OF ENGINEERING IN COMPUTER SCIENCE AND ENGINEERING**”, is the report of the original project work done by us under the guidance of **M. RAVIKUMAR, M.TECH., (Ph.D.)**, Assistant Professor, Department of Computer Science and Engineering, Hindusthan College of Engineering and Technology, Coimbatore.

NAME

SIGNATURE

1. MANUTHRA M

2. PRAGATHEESH R A

3. ROHITH S

I certify that the declaration made by the above candidates are true.

Project Guide,

M. RAVIKUMAR, M.TECH., (Ph.D.)

Assistant Professor,

Department of CSE,

Hindusthan College of Engineering and

Technology, Coimbatore –32

ACKNOWLEDGEMENT

We take this opportunity to express our wholehearted thanks and our profound respect to all those who guided and inspired us in the completion of this project work.

We extend our sincere thanks to the Founder and Chairman of Hindusthan Educational and Charitable Trust **Shri. T.S.R. KHANNAIYANN** and the Managing Trustee **Smt. SARASUWATHI KHANNAIYANN** and Executive Trustee & Secretary **Mrs. PRIYA SATISH PRABHU** for providing essential infrastructure. We would like to reveal our profound thanks to our respected Principal, **Dr. J.JAYA, MTech, Ph.D.**, who happens to be striving force in all endeavors.

We would like to express our gratitude to the Head of the Department **Dr. S. SHANKAR, M.E., Ph.D.**, for bringing out the project successfully and for strengthening the ray of hope.

We would like to express our sincere thanks and deep sense of gratitude to our Class Advisor and Guide **M. RAVIKUMAR, M.Tech., (Ph.D)**, Assistant Professor, Department of Computer Science and Engineering, for his valuable guidance, suggestions and constant encouragement which paved way for the successful completion of the mini project work.

We express our immense pleasure and thankfulness to all our department faculty members, technical staffs and friends who helped us for the successful completion of this mini project .We express our gratitude to **Mr. Mohanan M** and **Ms. SINDHU S** parents of MANUTHRA M ,**Ms. ANURADHA** parents of PRAGATHEESH R A , **Mr. SUBRAMANI S** and **Ms. GEETHA P** parents of ROHITH S and our family members who encouraged us and strengthened us in perilous path, encountered during our task.

ABSTRACT

The Smart University Admission Predictor is a web-based platform designed to simplify the university admission process by providing an intuitive and efficient system for predicting eligibility based on cutoff analysis. The platform allows students to input their 12th marks or CGPA, select their degree type (UG/PG), and choose a department, after which the system provides a list of eligible universities within Coimbatore based on pre-defined cutoff criteria.

By leveraging a dynamic, data-driven system, the platform automatically compares user input with the cutoff data for each college, streamlining the decision-making process for prospective students.

The Smart University Admission Predictor offers real-time updates, district-wise college listings, and official website links for eligible colleges, empowering students with the information needed to make informed choices regarding their higher education.

With features such as secure login, college filtering based on degree type and department, and a clean, user-friendly interface, the system enhances the university admission experience.

The platform's efficiency not only helps students find the best fit for their academic profile but also contributes to an overall smoother admission process.

Keywords: Smart University Admission Predictor, Web-Based Platform, University Admissions, Cutoff Analysis, Eligibility Prediction, Degree Selection, District-Based College Listings, CGPA/Marks Calculator, Data-Driven System, College Filtering, Real-Time Updates, User-Friendly Interface, Secure Login, Educational Decision-Making.

TABLE OF CONTENTS

CHAPTER	TITLE	PAGE NO
	ABSTRACT	i
1	INTRODUCTION	
	1.1 INTRODUCTION	1
2	SYSTEM ANALYSIS	2
	2.1 LITERATURE SURVEY	2
	2.1.1 PURPOSE	2
	2.1.2 EXISTING PROBLEM	2
	2.1.3 PROPOSED SOLUTION	3
	2.2 SYSTEM REQUIREMENTS	
	2.2.1 NON FUNCTIONAL REQUIREMENTS	4
	2.2.2 FUNCTIONAL REQUIREMENTS	5
	2.3 SOFTWARE AND HARDWARE REQUIREMENTS	
	2.3.1 SOFTWARE REQUIREMENT	5
	2.3.2 HARDWARE REQUIREMENT	6
	2.4 SMART UNIVERSITY ADMISSION PREDICTOR MODULES	6

3	SYSTEM DESIGN	
	3.1 DATA FLOW DIAGRAM	9
	3.2 TECHNICAL ARCHITECTURE	9
4	SYSTEM IMPLEMENTATION	
	4.1 USER INTERFACE : LOGIN/SIGNUP	10
	4.2 USER INTERFACE : DEPARTMENT SELECTION	11
	4.3 USER INTERFACE : ELIGIBILITY INTERFACE	12
5	SYSTEM TESTING	13
6	6.1 CONCLUSION	15
	6.2 FUTURE SCOPE	15
7	REFERENCES	16
8	APPENDICES	17

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

In the modern educational landscape, effective management of university admissions is critical for ensuring a smooth process for both prospective students and educational institutions. With the increasing complexity of admission criteria and varying cutoff thresholds, students often face challenges in determining their eligibility for different universities and courses. This project aims to develop a **Smart University Admission Predictor** that simplifies and automates the admission process, providing students with a reliable platform to predict their eligibility for university admissions based on cutoff analysis.

The system offers a user-friendly interface that integrates key modules, including degree type selection (UG/PG), department selection, and eligibility prediction based on 12th marks or CGPA. By automating the analysis of cutoff data, the system alleviates the burden on students. Additionally, the platform provides real-time access to eligibility results and college listings based on district, empowering students with the information they need to select the most suitable options for their academic profiles.

This **Smart University Admission Predictor** not only simplifies the university admission process but also offers valuable insights into the eligibility criteria of various universities. In response to the growing need for efficient tools in educational decision-making, this project serves as a reliable solution that enhances both the efficiency of the admission process and the ability of students to make well-informed choice.

CHAPTER 2

SYSTEM ANALYSIS

In this chapter, we will discuss the **Smart University Admission Predictor** and its features. In today's education system, managing university admissions effectively is essential for guiding students towards the right choices. Many students face challenges in determining their eligibility for different universities due to varying cutoff criteria. This system aims to simplify the process by allowing students to easily predict their eligibility based on their 12th marks or CGPA.

2.1 LITERATURE SURVEY

2.1.1 PURPOSE

The purpose of this project is to develop a comprehensive web-based **Smart University Admission Predictor** that simplifies the university admission process for prospective students. With the complexity of varying cutoff criteria and admission requirements across universities, students often face difficulties in determining their eligibility. This system aims to streamline the process by allowing students to easily predict their eligibility based on 12th marks or CGPA

2.1.2 EXISTING PROBLEM

University admissions can be complex, with varying cutoff criteria across institutions. Students often face difficulties in determining their eligibility, leading to confusion and delays. This system addresses these challenges by automating eligibility prediction, streamlining the process for students to make informed decisions efficiently.

2.1.3 PROPOSED SOLUTION

S.no	Parameter	Description
1.	Problem to be solved	Students face challenges in determining their eligibility for various universities due to varying cutoff criteria.
2.	Idea/solution description	The project provides a web-based platform that automates eligibility prediction based on 12th marks or CGPA, helping students make informed decisions.
3.	Novelty/Uniqueness	The system offers real-time eligibility updates, a user-friendly interface, and district-wise college filtering, ensuring a seamless user experience
4.	Social impact/Customer Satisfaction	The system simplifies the admission process, improving decision-making efficiency and reducing confusion for students..
5.	Business Model (Revenue Model)	The platform could adopt a freemium model, offering basic services for free and advanced features (like detailed reports) on a subscription basis.
6.	Scalability of the Solution	The system can scale to include more districts, additional universities, and various courses as the user base grows.

2.2 SYSTEM REQUIREMENTS

2.2.1 NON-FUNCTIONAL REQUIREMENTS

SN No.	Non-Functional Requirement	Description
1	Usability	The system should be user-friendly, allowing students to easily navigate through the admission process.
2	Security	Student data must be securely stored and transmitted, ensuring privacy and compliance with data protection regulations.
3	Reliability	The system must operate consistently with minimal downtime, ensuring smooth access to eligibility results.
4	Performance	The system should process and update student eligibility data quickly, with minimal delay in displaying results.
5	Availability	The website should be available 24/7 for users to check their eligibility and make decisions anytime.
6	Scalability	The platform must handle increased user traffic and allow the addition of more features, such as filtering by additional criteria (e.g., specializations).

2.2.2 FUNCTIONAL REQUIREMENTS

SN No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
1	Collecting user details	Students should be able to register, update, and view their personal details, including marks and CGPA.
2	Launch the website	The system should be accessible via a web browser for students to check their eligibility for university admissions.
3	View the result	Users should be able to input their marks or CGPA and view a list of eligible universities based on their selected criteria.
4	Taking decision	The system should provide accurate eligibility predictions and recommendations for universities, helping students make informed decisions.
5	Compare College	Users should be able to compare multiple eligible colleges based on course availability, fees, and location.
6	Save preferred Choices	Students should be able to save a shortlist of preferred universities for future reference and decision-making.

2.3 SOFTWARE AND HARDWARE REQUIREMENTS:

2.3.1 SOFTWARE REQUIREMENTS

- 1. OPERATING SYSTEM:** Ubuntu, Windows, or macOS.
- 2. FRONTEND:** HTML, CSS, JavaScript.
- 3. BACKEND:** Python with Flask
- 4. DATABASE:** SQLite(for managing user data)
- 5. IDE:** Visual Studio Code or any preferred code editor.
- 6. Deployment:** GitHub Pages or Netlify for hosting the website.

2.3.2 HARDWARE REQUIREMENTS

- 1. Processor:** Intel Core i3 or equivalent AMD Ryzen processors (for development and server hosting).
- 2. RAM:** Minimum 8 GB (recommended 16 GB for handling larger datasets and multiple processes).
- 3. Storage:** Solid-State Drive (SSD) with at least 512 GB of storage (for faster data access and system performance).

2.3 SMART UNIVERSITY ADMISSION PREDICTOR MODULES:

- 1. User Registration and Login Module:** This module allows students to create an account, log in, and securely manage their personal details, including marks and CGPA. It ensures a seamless user experience with secure authentication.

2. **Eligibility Calculation Module:** This module processes the student's input (marks or CGPA) and calculates eligibility based on predefined criteria for various universities and departments, offering instant results for students to assess their admission prospects.
3. **University and College Listing Module:** This module displays a list of eligible universities and their details, based on the student's eligibility. It includes information such as university name, location, courses offered, and official websites..
4. **District Selection and College Filter Module:** This module allows students to filter colleges by district and department, providing more localized results. Students can select their preferred district to find relevant universities in their area.
5. **Results Display Module:** This module presents the final results to the student, showing a list of eligible universities and colleges based on their input data. It includes links to the official websites for easy access.
6. **Admin Dashboard Module:** This module provides administrators with a centralized interface to manage student registrations, monitor user activity, and update university and college data. It includes features for managing system settings and monitoring usage statistics.
7. **Database Management Module:** This module handles all the backend operations related to storing and retrieving student data, university details, and eligibility criteria. It ensures that the database is updated and optimized for fast queries
8. **Analytics and Reporting Module:** This module generates insights on student admissions trends, such as popular courses, preferred districts, and eligibility patterns. It helps the admin to monitor the system's performance and improve the service.
9. **Student Profile Management Module:** This module allows students to update and manage their profiles, including their personal information, academic records, and

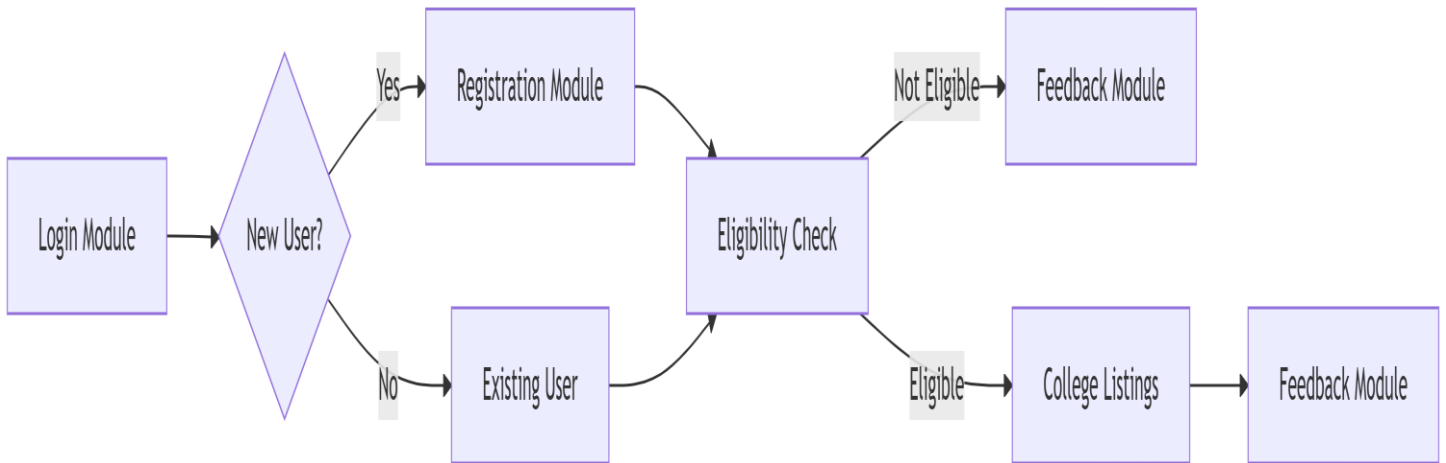
preferred course or university options. It ensures that all the necessary data is accurate and up-to-date for eligibility calculations.

10. **Eligibility Criteria Update Module:** This module allows administrators to update and modify eligibility criteria for different courses and universities. It ensures that the system reflects any changes in the admission process or university policies.
11. **Help and Support Module:** This module provides students with FAQs, tutorials, and contact information for support. It ensures that students can navigate the platform effectively and resolve any issues they may face.
12. **Feedback and Rating Module:** This module allows students to rate and provide feedback on the universities and courses they are considering. It helps future students make informed decisions based on the experiences of their peers.
13. **Notification and Alert System:** This module sends notifications to students and administrators regarding system updates, eligibility results, important deadlines, and any changes in university criteria. It keeps all users informed and engaged with timely alerts.

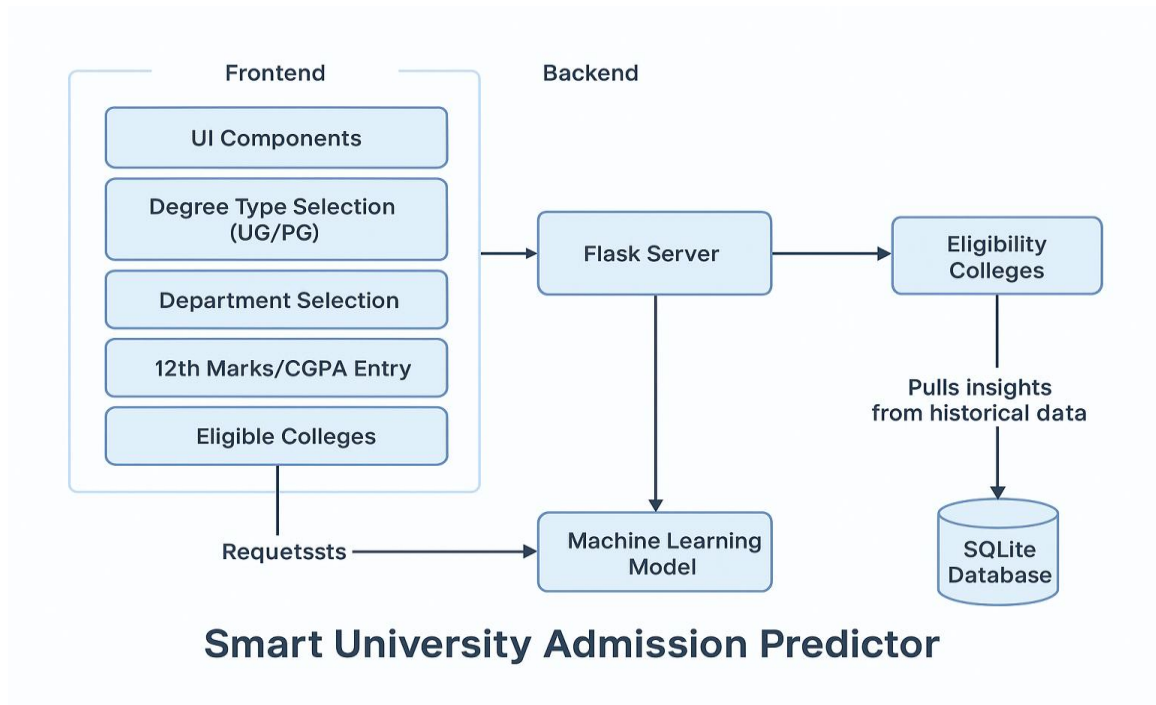
CHAPTER 3

SYSTEM DESIGN

3.1 DATA FLOW DIAGRAMS



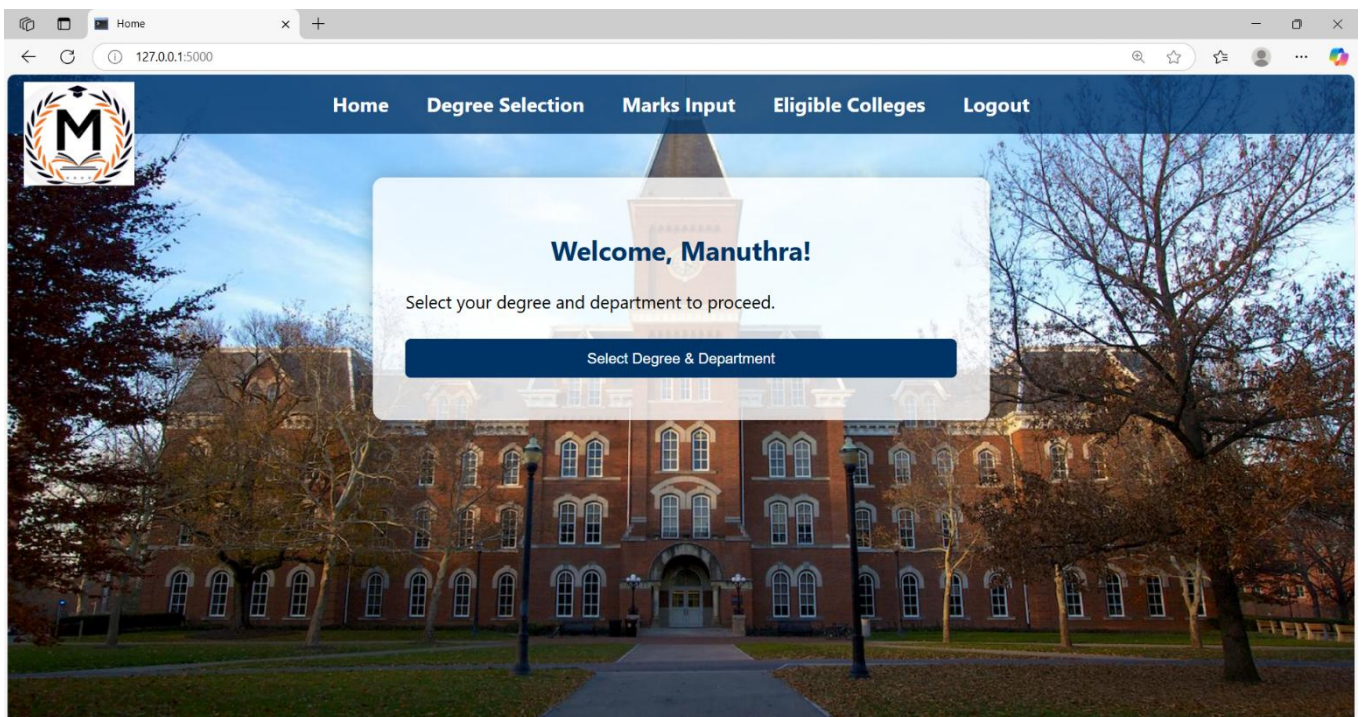
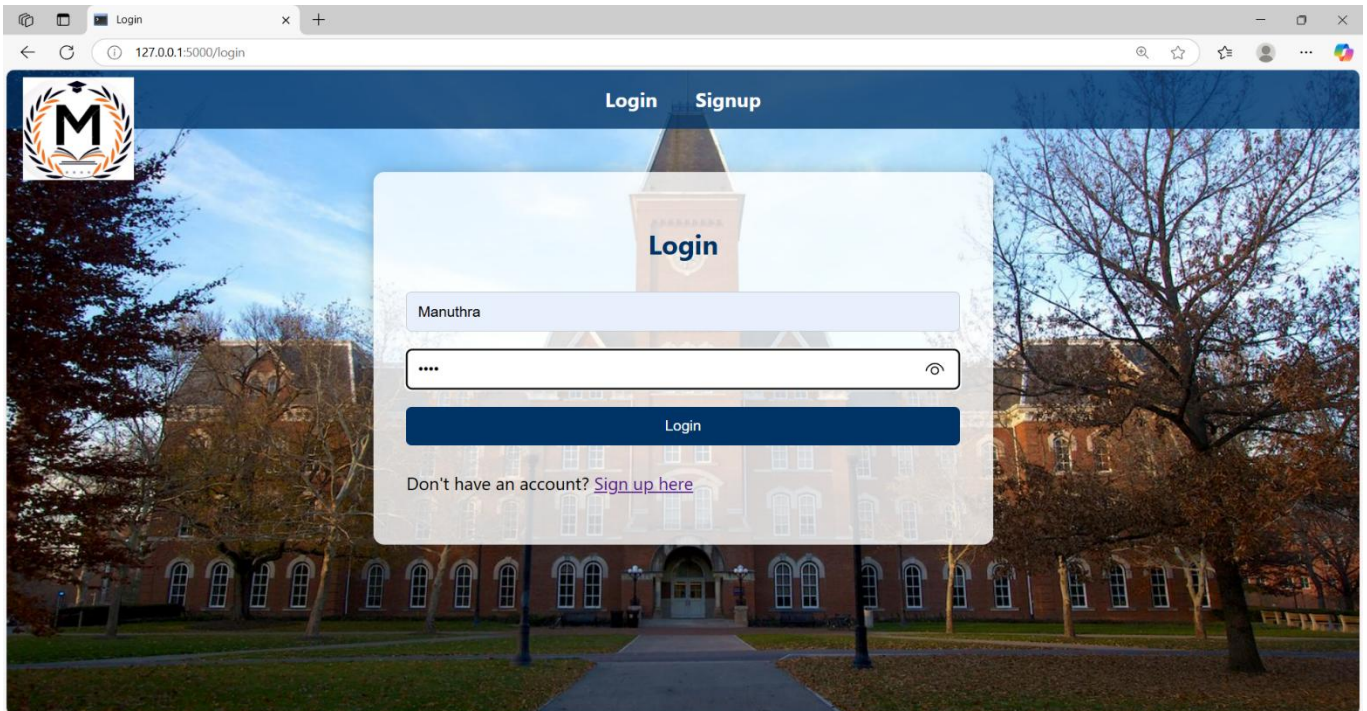
3.2 TECHNICAL ARCHITECTURE



CHAPTER 4

4. SYSTEM IMPLEMENTATION

4.1 USER INTERFACE : LOGIN / SIGNUP



4.2 USER INTERFACE: DEPARTMENT SELECTION

The screenshot shows a web browser window with the address bar displaying '127.0.0.1:5000/degree-selection'. The page features a dark blue navigation bar with the following links: Home, Degree Selection, Marks Input, Eligible Colleges, and Logout. A logo with a stylized 'M' inside a laurel wreath is positioned on the left. The main content area has a background image of a large, historic brick building with many windows. Overlaid on this is a white modal box titled 'Select Degree and Department'. Inside the modal, there are two dropdown menus: 'Degree Type' set to 'Undergraduate (UG)' and 'Department' set to 'Computer Science Engineering (CSE)'. Below these is a 'Next' button.

Select Degree

Home Degree Selection Marks Input Eligible Colleges Logout

Select Degree and Department

Degree Type: Undergraduate (UG) ▼

Department: Computer Science Engineering (CSE) ▼

Next

The screenshot shows a web browser window with the address bar displaying '127.0.0.1:5000/marks-input'. The page features the same dark blue navigation bar and logo as the previous screenshot. The main content area has the same background image of the historic building. Overlaid on this is a white modal box titled 'Enter Your 12th Marks'. Inside the modal, there is a 'Select District' dropdown menu set to 'Coimbatore'. Below this are three input fields for marks: 'Physics' with the value '95', 'Chemistry' with the value '94', and 'Mathematics' with the value '96'. At the bottom of the modal is a blue 'Submit' button.

Enter 12th Marks

Home Degree Selection Marks Input Eligible Colleges Logout

Select District: Coimbatore ▼

Physics: 95

Chemistry: 94

Mathematics: 96

Submit

Enter CGPA for Postgraduate (PG) Degree

Select District: Coimbatore

CGPA:

Submit

4.3 USER INTERFACE: ELIGIBILITY RESULT

Eligible Colleges

Cutoff: 190.5

District: Coimbatore

College Name	Department	Cutoff	District	Website
PSG Tech	CSE	190	Coimbatore	Visit
Kumaraguru College	CSE	180	Coimbatore	Visit
Karpagam College	CSE	150	Coimbatore	Visit
SNS College of Technology	CSE	155	Coimbatore	Visit
P.A. College of Engineering	CSE	140	Coimbatore	Visit

CHAPTER 5

SYSTEM TESTING

For the SMART University Admission Predictor, system testing focuses on validating the functionality, integration, performance, and security to ensure the platform accurately predicts eligible colleges based on user input. The testing process ensures that users experience smooth and reliable access to admission predictions:

1. Functionality Testing:

Functional testing ensures that the system performs all its intended functions as per the specifications. Core features like user registration, login, degree selection (UG/PG), department selection, entering 12th marks or CGPA, district selection, and cutoff-based prediction of colleges are tested. The prediction process is verified to ensure that users get accurate results without any errors. Additionally, it checks for proper form validations and smooth navigation across all pages to confirm that the system operates seamlessly.

2. Integration Testing:

Integration testing focuses on verifying smooth communication between the frontend (HTML, CSS, JavaScript) and backend (Python with Flask). It ensures that user inputs from the frontend are correctly processed by the backend logic, and the data is accurately retrieved and stored in the SQLite database. All internal connections between modules like login, prediction engine, and eligibility display pages are tested to ensure a seamless user experience.

3. Performance Testing:

Performance testing evaluates the system's speed, responsiveness, and stability under various workloads. Key performance metrics such as page load time (expected under three seconds) are tested to ensure fast results. The system is also tested for multiple user

scenarios to verify that it handles a moderate number of simultaneous users efficiently without slowing down or crashing, ensuring consistent performance during heavy traffic.

4. Security Testing:

Security testing ensures that user data and application functionalities are protected. Tests are conducted to verify secure login authentication, safe handling of input data to prevent SQL injection, and protection of stored information in the database. If deployed online, SSL/TLS protocols are checked to ensure encrypted data transmission. This testing strengthens the system's defenses against unauthorized access and data breaches.

5. Usability Testing:

Usability testing assesses how easy and intuitive the system is for users. The interface is tested to ensure clear instructions, simple navigation between pages, and user-friendly forms for entering details. The prediction results are made easily understandable, and the overall design is evaluated for mobile responsiveness and accessibility for users with different devices and skill levels.

6. Regression Testing:

Regression testing ensures that after updates, bug fixes, or feature additions, all core functionalities continue to work correctly. Key processes such as login, entering marks, selecting preferences, and generating predictions are retested after any code changes. This guarantees that improvements do not unintentionally break existing features, maintaining the system's overall reliability and performance.

CHAPTER 6

CONCLUSION

Managing college admission predictions based on cutoff scores can be challenging and confusing for students. Our SMART University Admission Predictor simplifies this process with a user-friendly platform that streamlines degree selection, department choice, marks or CGPA input, and eligibility prediction. It helps students make informed decisions, reduces the confusion around admission processes, and provides clear, accurate results. Ultimately, it offers a comprehensive solution for a smooth and stress-free college admission experience.

FUTURE SCOPE

- Enhance system design for a more intuitive and user-friendly admission prediction experience.
- Implement AI and machine learning models for more accurate and personalized college predictions.
- Strengthen data encryption and secure API integration to protect user information and cutoff data.
- Expand the database to include more universities, courses, and updated cutoff trends.
- Introduce a counseling chatbot for providing instant admission guidance to students.
- Promote a fully digital, paperless admission support system to reduce environmental impact.

CHAPTER 7

REFERENCES

1. Kumar, R., & Gupta, S. (2023). Building Efficient University Admission Prediction Systems: A Web-Based Approach. *Journal of Education Technology and Management*, 18(3), 45-60.
2. Van Busum, K., & Fang, S. (2023). Analysis of AI Models for Student Admissions: A Case Study. *ACM/SIGAPP Symposium on Applied Computing*.
3. Raghavendran, C. V., et al. (2021). Predicting Student Admissions Rate into University Using Machine Learning Models. *Machine Intelligence and Soft Computing*.
4. Ramya, S., et al. (2023). An Analysis of University Admission Prediction using Machine Learning. *Advances in Intelligent Systems and Computing*.
5. Lee, H., Kizilcec, R. F., & Joachims, T. (2023). Evaluating a Learned Admission-Prediction Model as a Replacement for Standardized Tests in College Admissions. *Proceedings of the 2023 ACM Conference on Learning at Scale*

CHAPTER 8

APPENDICES

App.py:

```
from flask import Flask, render_template, request, redirect, url_for, session
import sqlite3

app = Flask(__name__)
app.secret_key = 'your_secret_key'

def init_db():
    with sqlite3.connect('users.db') as conn:
        cursor = conn.cursor()
        cursor.execute("CREATE TABLE IF NOT EXISTS users
                        (username TEXT PRIMARY KEY, password TEXT)")
        conn.commit()

def get_user_from_db(username):
    with sqlite3.connect('users.db') as conn:
        cursor = conn.cursor()
        cursor.execute('SELECT * FROM users WHERE username = ?', (username,))
        user = cursor.fetchone()
    return user

def add_user_to_db(username, password):
    with sqlite3.connect('users.db') as conn:
        cursor = conn.cursor()
        cursor.execute('INSERT INTO users (username, password) VALUES (?, ?)',
                        (username, password))
        conn.commit()

@app.route('/')
def home():
    if 'username' in session:
        return render_template('home.html', username=session['username'])
    return redirect(url_for('login'))

@app.route('/signup', methods=['GET', 'POST'])
def signup():
    message = "
```



```

if request.method == 'POST':

    username = request.form['username']
    password = request.form['password']

    user = get_user_from_db(username)
    if user:
        message = 'Username already exists!'
    else:
        add_user_to_db(username, password)
        return redirect(url_for('login'))
    return render_template('signup.html', message=message)

@app.route('/login', methods=['GET', 'POST'])
def login():
    message = ""
    if request.method == 'POST':
        username = request.form['username']
        password = request.form['password']

        user = get_user_from_db(username)
        if user and user[1] == password:
            session['username'] = username
            return redirect(url_for('degree_selection'))
        else:
            message = 'Invalid credentials!'
    return render_template('login.html', message=message)

@app.route('/degree-selection', methods=['GET', 'POST'])
def degree_selection():
    if request.method == 'POST':
        degree_type = request.form['degree_type']
        department = request.form['department']

        session['degree_type'] = degree_type
        session['department'] = department

        return redirect(url_for('marks_input'))

    return render_template('degree_selection.html')

@app.route('/marks-input', methods=['GET', 'POST'])
def marks_input():
    if request.method == 'POST':
        degree_type = session.get('degree_type')

```

```

district = request.form['district']

session['district'] = district

if degree_type == 'UG':
    physics = float(request.form['physics'])
    chemistry = float(request.form['chemistry'])
    maths = float(request.form['maths'])

    cutoff = (physics / 2) + (chemistry / 2) + maths
    session['cutoff'] = cutoff

elif degree_type == 'PG':
    cgpa = float(request.form['cgpa'])
    cutoff = cgpa * 10
    session['cutoff'] = cutoff

return redirect(url_for('eligible_colleges'))

degree_type = session.get('degree_type')
if degree_type == 'UG':
    return render_template('marks_input_ug.html')
elif degree_type == 'PG':
    return render_template('marks_input_pg.html')
else:
    return redirect(url_for('degree_selection'))

@app.route('/eligible-colleges')
def eligible_colleges():
    cutoff = session.get('cutoff')
    department = session.get('department')
    district = session.get('district')
    degree_type = session.get('degree_type')

    # College list updated for UG and PG separately
    colleges = [

        # Coimbatore - UG
        {'name': 'PSG Tech', 'department': 'CSE', 'cutoff': 190, 'district': 'Coimbatore', 'degree':
'UG', 'website': 'https://www.psgtech.edu/'},
        {'name': 'Kumaraguru College', 'department': 'CSE', 'cutoff': 180, 'district':
'Coimbatore', 'degree': 'UG', 'website': 'https://www.kct.ac.in/'},

        {'name': 'Sri Krishna College', 'department': 'IT', 'cutoff': 170, 'district': 'Coimbatore',

```

'degree': 'UG'},

{'name': 'Hindusthan College', 'department': 'EEE', 'cutoff': 160, 'district': 'Coimbatore', 'degree': 'UG'},

{'name': 'Karpagam College', 'department': 'CSE', 'cutoff': 150, 'district': 'Coimbatore', 'degree': 'UG'},

{'name': 'SNS College of Technology', 'department': 'CSE', 'cutoff': 155, 'district': 'Coimbatore', 'degree': 'UG'},

{'name': 'Sri Ramakrishna Engineering College', 'department': 'IT', 'cutoff': 165, 'district': 'Coimbatore', 'degree': 'UG'},

{'name': 'Dr. N.G.P Institute of Technology', 'department': 'EEE', 'cutoff': 158, 'district': 'Coimbatore', 'degree': 'UG'},

{'name': 'P.A. College of Engineering and Technology', 'department': 'CSE', 'cutoff': 140, 'district': 'Coimbatore', 'degree': 'UG'},

Coimbatore - PG

{'name': 'PSG Tech', 'department': 'CSE', 'cutoff': 75, 'district': 'Coimbatore', 'degree': 'PG'},

{'name': 'Kumaraguru College', 'department': 'CSE', 'cutoff': 70, 'district': 'Coimbatore', 'degree': 'PG'},

{'name': 'Sri Krishna College', 'department': 'IT', 'cutoff': 68, 'district': 'Coimbatore', 'degree': 'PG'},

{'name': 'Hindusthan College', 'department': 'EEE', 'cutoff': 65, 'district': 'Coimbatore', 'degree': 'PG'},

{'name': 'Karpagam College', 'department': 'CSE', 'cutoff': 60, 'district': 'Coimbatore', 'degree': 'PG'},

{'name': 'SNS College of Technology', 'department': 'CSE', 'cutoff': 62, 'district': 'Coimbatore', 'degree': 'PG'},

{'name': 'Sri Ramakrishna Engineering College', 'department': 'IT', 'cutoff': 67, 'district': 'Coimbatore', 'degree': 'PG'},

{'name': 'Dr. N.G.P Institute of Technology', 'department': 'EEE', 'cutoff': 64, 'district': 'Coimbatore', 'degree': 'PG'},

{'name': 'P.A. College of Engineering and Technology', 'department': 'CSE', 'cutoff': 58, 'district': 'Coimbatore', 'degree': 'PG'},

Chennai - UG

{'name': 'College of Engineering, Guindy', 'department': 'CSE', 'cutoff': 190, 'district': 'Chennai', 'degree': 'UG'},

{'name': 'Anna University', 'department': 'CSE', 'cutoff': 195, 'district': 'Chennai', 'degree': 'UG'},

{'name': 'SSN College of Engineering', 'department': 'IT', 'cutoff': 185, 'district': 'Chennai', 'degree': 'UG'},

{'name': 'MIT Campus', 'department': 'EEE', 'cutoff': 180, 'district': 'Chennai', 'degree': 'UG'},

{'name': 'SRM Institute of Science and Technology', 'department': 'CSE', 'cutoff': 170,

'district': 'Chennai', 'degree': 'UG'},

{'name': 'Saveetha Engineering College', 'department': 'CSE', 'cutoff': 165, 'district': 'Chennai', 'degree': 'UG'},

{'name': 'VIT Chennai', 'department': 'IT', 'cutoff': 175, 'district': 'Chennai', 'degree': 'UG'},

Chennai - PG

{'name': 'College of Engineering, Guindy', 'department': 'CSE', 'cutoff': 78, 'district': 'Chennai', 'degree': 'PG'},

{'name': 'Anna University', 'department': 'CSE', 'cutoff': 80, 'district': 'Chennai', 'degree': 'PG'},

{'name': 'SSN College of Engineering', 'department': 'IT', 'cutoff': 76, 'district': 'Chennai', 'degree': 'PG'},

{'name': 'MIT Campus', 'department': 'EEE', 'cutoff': 75, 'district': 'Chennai', 'degree': 'PG'},

{'name': 'SRM Institute of Science and Technology', 'department': 'CSE', 'cutoff': 70, 'district': 'Chennai', 'degree': 'PG'},

{'name': 'Saveetha Engineering College', 'department': 'CSE', 'cutoff': 68, 'district': 'Chennai', 'degree': 'PG'},

{'name': 'VIT Chennai', 'department': 'IT', 'cutoff': 74, 'district': 'Chennai', 'degree': 'PG'},

Trichy - UG

{'name': 'NIT Trichy', 'department': 'CSE', 'cutoff': 195, 'district': 'Trichy', 'degree': 'UG'},

{'name': 'Saranathan College of Engineering', 'department': 'IT', 'cutoff': 170, 'district': 'Trichy', 'degree': 'UG'},

{'name': 'M.A.M College of Engineering', 'department': 'EEE', 'cutoff': 160, 'district': 'Trichy', 'degree': 'UG'},

{'name': 'Oxford Engineering College', 'department': 'CSE', 'cutoff': 150, 'district': 'Trichy', 'degree': 'UG'},

{'name': 'Jayaram College of Engineering', 'department': 'CSE', 'cutoff': 140, 'district': 'Trichy', 'degree': 'UG'},

Trichy - PG

{'name': 'NIT Trichy', 'department': 'CSE', 'cutoff': 80, 'district': 'Trichy', 'degree': 'PG'},

{'name': 'Saranathan College of Engineering', 'department': 'IT', 'cutoff': 70, 'district': 'Trichy', 'degree': 'PG'},

{'name': 'M.A.M College of Engineering', 'department': 'EEE', 'cutoff': 65, 'district': 'Trichy', 'degree': 'PG'},

{'name': 'Oxford Engineering College', 'department': 'CSE', 'cutoff': 60, 'district': 'Trichy', 'degree': 'PG'},

{'name': 'Jayaram College of Engineering', 'department': 'CSE', 'cutoff': 58, 'district': 'Trichy', 'degree': 'PG'},

```

# Madurai - UG
{'name': 'Thiagarajar College of Engineering', 'department': 'CSE', 'cutoff': 180,
'district': 'Madurai', 'degree': 'UG'},
{'name': 'Madurai Institute of Engineering and Technology', 'department': 'IT', 'cutoff':
160, 'district': 'Madurai', 'degree': 'UG'},
{'name': 'Velammal College of Engineering', 'department': 'EEE', 'cutoff': 165, 'district':
'Madurai', 'degree': 'UG'},
{'name': 'K.L.N. College of Engineering', 'department': 'CSE', 'cutoff': 150, 'district':
'Madurai', 'degree': 'UG'},
{'name': 'Raja College of Engineering and Technology', 'department': 'CSE', 'cutoff':
140, 'district': 'Madurai', 'degree': 'UG'},

```

```

# Madurai - PG
{'name': 'Thiagarajar College of Engineering', 'department': 'CSE', 'cutoff': 77, 'district':
'Madurai', 'degree': 'PG'},
{'name': 'Velammal College of Engineering', 'department': 'EEE', 'cutoff': 70, 'district':
'Madurai', 'degree': 'PG'},
{'name': 'K.L.N. College of Engineering', 'department': 'CSE', 'cutoff': 68, 'district':
'Madurai', 'degree': 'PG'},
{'name': 'Raja College of Engineering and Technology', 'department': 'CSE', 'cutoff':
60, 'district': 'Madurai', 'degree': 'PG'},

```

```

]
```

```

# Filter based on degree type
eligible_colleges = [
    college for college in colleges
    if college['department'] == department and
    college['district'] == district and
    college['degree'] == degree_type and
    college['cutoff'] <= cutoff
]
```

```

return render_template('eligible_colleges.html', colleges=eligible_colleges,
cutoff=cutoff)

```

```

@app.route('/logout')
def logout():
    session.pop('username', None)
    session.pop('degree_type', None)
    session.pop('department', None)
    session.pop('cutoff', None)

```

```

session.pop('district', None)

return redirect(url_for('login'))

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

```

LOGIN Page:

```

{% extends "base.html" %}

{% block title %}Login{% endblock %}

{% block content %}
<h2>Login</h2>
<form method="POST">
    <input type="text" name="username" placeholder="Username" required><br>
    <input type="password" name="password" placeholder="Password" required><br>
    <input type="submit" value="Login">
</form>
<p class="message">{{ message }}</p>
<p>Don't have an account? <a href="{{ url_for('signup') }}">Sign up here</a></p>
{% endblock %}

```

Signup.html:

```

{% extends "base.html" %}

{% block title %}Signup{% endblock %}

{% block content %}
<h2>Signup</h2>
<form method="POST">
    <input type="text" name="username"
placeholder="Username" required><br>
    <input type="password" name="password"
placeholder="Password" required><br>
    <input type="submit" value="Signup">
</form>
<p class="message">{{ message }}</p>
<p>Already have an account? <a href="{{
url_for('login') }}">Login here</a></p>
{% endblock %}

```

EligibleCollege.html:

```
{% extends "base.html" %}

{% block title %}Enter Marks for UG{%
endblock %}

{% block content %}
<h2>Eligible Colleges</h2>
<p>Cutoff: {{ cutoff }}</p>
<p>District: {{ session['district'] }}</p>

<table>
  <tr>
    <th>College Name</th>
    <th>Department</th>
    <th>Cutoff</th>
    <th>District</th>
    <th>Website</th> <!-- NEW COLUMN -->
  </tr>
  {% for college in colleges %}
    <tr>
      <td>{{ college.name }}</td>
      <td>{{ college.department }}</td>
      <td>{{ college.cutoff }}</td>
      <td>{{ college.district }}</td>
      <td><a href="{{ college.website }}"
target="_blank">Visit</a></td> <!-- Website
link -->
    </tr>
  {% endfor %}
</table>
{% endblock %}
```

Style.css:

```
/* Background Image */
body {
  background-image:
url('https://phoscreative.com/wp-
```

content/uploads/2017/10/How-to-Build-a-Powerful-College-Website-1-1-2.jpg');

```
background-size: cover;
background-position: center;
font-family: Arial, sans-serif;
margin: 0;
padding: 0;
}

/* Logo Styling */
.logo {
width: 150px; /* Adjust size of the logo */
height: auto;
margin: 20px;
display: block;
margin-left: auto;
margin-right: auto;
}

/* Navigation Bar Styling */
nav {
background-color: #003366;
padding: 15px;
text-align: center;
}

nav a {
color: white;
text-decoration: none;
margin: 0 15px;
font-weight: bold;
font-size: 18px;
}

nav a:hover {
text-decoration: underline;
}

/* Container Styling */
.container {
margin: 40px auto;
background: rgba(255, 255, 255, 0.8); /*
```



```

Semi-transparent background */
padding: 30px;

max-width: 500px;
box-shadow: 0px 0px 15px rgba(0,0,0,0.2);
border-radius: 10px;
}

/* Button and Input Styling */
input[type="text"], input[type="password"],
input[type="number"], input[type="submit"] {
    width: 100%;
    padding: 10px;
    margin: 8px 0;
    box-sizing: border-box;
    border: 1px solid #ccc;
    border-radius: 5px;
}

input[type="submit"] {
    background-color: #003366;
    color: white;
    border: none;
    cursor: pointer;
}

input[type="submit"]:hover {
    background-color: #00509e;
}

/* Table Styling */
table {
    width: 100%;
    border-collapse: collapse;
    margin-top: 20px;
}

th, td {
    padding: 10px;
    text-align: center;
    border: 1px solid #ccc;
}

```

```
th {  
  background-color: #003366;  
  
  color: white;  
}  
  
/* Message Styling */  
.message {  
  text-align: center;  
  color: red;  
}
```