ASKDU: EMPOWERING STUDENTS WITH AI-DRIVEN CAREER NAVIGATION AND COLLEGE COUNSELLING ASSISTANCE

A MINI PROJECT REPORT

Submitted by

MASANIYAMMAL N	**61772121025**
MEGHA HARTHANA S	**61772121026**
SIVARANJANI C K	**61772121042**

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

GOVERNMENT COLLEGE OF ENGINEERING, SALEM – 636 011

(An Autonomous Institution Affiliated to Anna University, Chennai, NAAC Accredited)

ANNA UNIVERSITY: CHENNAI 600 025

JUNE 2024

GOVERNMENT COLLEGE OF ENGINEERING, SALEM-11

(An Autonomous Institution Affiliated to Anna University, Chennai, NAAC Accredited)

ANNA UNIVERSITY: CHENNAI-600 025

BONAFIDE CERTIFICATE

Certified that this mini project report "**ASKDU: EMPOWERING STUDENTS WITH AI-DRIVEN CAREER NAVIGATION AND COLLEGE COUNSELLING ASSISTANCE**" is the bonafide work of "**MASANIYAMMAL N (61772121025), MEGHA HARTHANA S (61772121026) AND SIVARANJANI C K (61772121042)**" who carried out the mini project under my supervision during the academic year 2023-2024.

SIGNATURE	**SIGNATURE**
Dr. A. M. KALPANA M.E., Ph.D. **PROFESSOR &** **HEAD OF THE DEPARTMENT**	**Dr. P. THARANI M.E., Ph.D.** **ASSISTANT PROFESSOR &** **SUPERVISOR**
Computer Science and Engineering, Government College of Engineering, Salem - 636 011	Computer Science and Engineering, Government College of Engineering, Salem - 636 011

Submitted for the mini project Viva-Voice examination held at the Government College of Engineering, Salem-11 on ____________

INTERNAL EXAMINER	**EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

I extend my deepest appreciation to Government college of Engineering, Salem for providing me with the opportunity to work on this project. Their trust, support, and resources have been invaluable in bringing this student management system to fruition.

We convey our heartfelt gratitude to the honourable and respected principal, **Dr. R. VIJAYAN M.E., Ph.D.** for his encouragement and support for the successful completion of the project

We would like to thank and express our gratitude to the Head of our Department, **Dr. A. M. KALPANA M.E., Ph.D.** who took keen interest till the completion of our project work by providing all the necessary information for developing a good system.

We would like to thank and express our gratitude to our project Guide and coordinator, **Dr. P. THARANI M.E., Ph.D.** for her valuable guidance and constant encouragement right from the beginning to accomplish the project successfully.

I express my heartfelt thanks to the teaching faculty members and non-teaching staff members who actively participated in the system's development and implementation process. Their valuable insights, feedback, and extensive domain knowledge have helped shape the system to cater to the unique needs and requirements.

We also acknowledge with a deep sense of reverence and gratitude towards our parents, family members and friends, who supported us for the successful completion of the project. Their involvement and partnership have been vital in ensuring a smooth transition and successful adoption of the system.

ABSTRACT

In today's fast-paced technological landscape, navigating the complexities of higher education has become increasingly challenging for students, parents, educators, and career advisors. Recognizing this need for innovation, this project introduces a groundbreaking chatbot system designed to revolutionize higher education guidance. At its core, the chatbot acts as a virtual advisor, harnessing the power of Artificial Intelligence (AI) and Natural Language Processing (NLP) to offer personalized support at every stage of a student's academic journey.

Unlike traditional advisory methods, our chatbot goes beyond one-size-fits-all approaches, analysing each student's unique profile, preferences, and aspirations to provide tailored guidance. It provides real-time assistance and accessibility, enabling students to access timely information and advice whenever and wherever they need it, while features such as exploring colleges beyond initial interests and streamlining profile creation cater to diverse user needs. It facilitates feedback mechanisms, allowing users to provide valuable input on their experiences, thus providing continuous improvement. Additionally, the chatbot offers tools like a cutoff calculator and detailed reports, empowering users with valuable insights to aid decision-making. By offering these features, this project aims to transform the higher education guidance landscape. The merits are

- Personalized guidance tailored to individual profiles enhances effectiveness.
- Real-time assistance ensures timely access to information.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

AI	-	Artificial Intelligence
API	-	Application Program Interface
CSS	-	Cascading Style Sheets
DFD	-	Data Flow Diagram
DOM	-	Document Object Model
HTML	-	Hyper Text Mark-up Language
JSON	-	JavaScript Object Notation
ML	-	Machine Learning
NLP	-	Natural Language Processing
UT	-	User Testing
UI	-	User Interface

CHAPTER 1
INTRODUCTION

1.1 OVERVIEW

This project is dedicated to providing comprehensive career guidance for students after completing their 12th grade education. In today's competitive world, making informed decisions about future career paths is crucial for students transitioning from secondary education to higher studies or employment opportunities. Recognizing this critical juncture in their lives, our project aims to offer a centralized platform for students to explore and plan their career paths effectively.

At the heart of our project is a chatbot system designed to serve as a virtual career advisor tailored specifically for post-12th grade students. The platform consolidates a wealth of information on various career options, educational pathways, entrance exams, and skill development opportunities into an easily accessible format. By providing personalized guidance and support, this chatbot assists students in making informed decisions about their future.

The primary objective of our project is to empower students with the knowledge and resources they need to navigate their landscape confidently. Through real-time assistance and user-friendly interfaces, this platform offers students the opportunity to explore diverse career options, understand admission requirements for different courses and colleges, and access guidance on skill development and career planning strategies.

Additionally, this platform provides several features like cutoff calculator, allowing students to assess their eligibility for various courses and colleges. A feedback mechanism enables users to provide valuable input on their experiences and preferences, offering continuous improvement. Seamless profile creation facilitates easy access to personalized features, while report download functionality offers valuable insights and analyses to aid decision-making.

In summary, this project aims to revolutionize career guidance for students by offering a centralized platform equipped with personalized support, resources, and essential tools. By assisting students in making informed decisions about their future career paths, this aims to empower them to pursue rewarding opportunities and achieve success in their chosen fields. AskDu chatbot has been depicted in Figure 1.1.

Figure 1.1 AskDu Chatbot

CHAPTER 2
SYSTEM ANALYSIS

System analysis is “the process of examining a system's components, interactions, and objectives in order to understand its structure, functionality, and behaviour”. It involves studying the system's requirements, constraints, and goals to identify potential improvements or solutions.

During system analysis, analysts gather information about the current system, such as its users, processes, data, and technology infrastructure. They analyse this information to identify strengths, weaknesses, opportunities, and threats (SWOT analysis) associated with the system.

2.1 EXISTING SYSTEM

Students face several challenges in their career planning journey. Traditional methods such as in-person counselling or workshops can be time-consuming and resource-intensive, posing difficulties for both students and institutions.

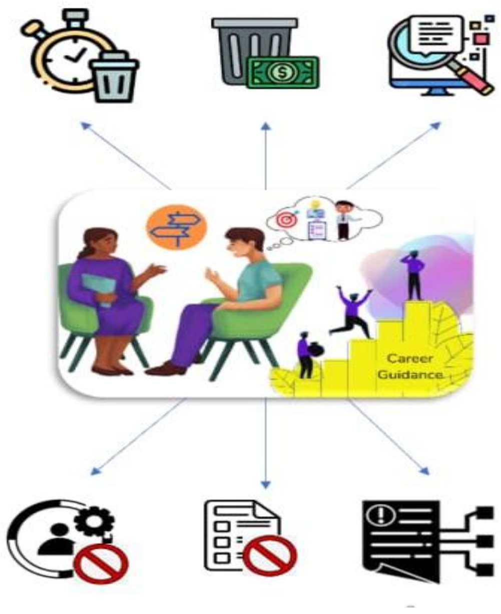


Figure 2.1 Existing System

Additionally, accessing information about colleges and competitive exams is often fragmented across scattered online resources, leading to inefficiencies in decision-making has been depicted in Figure 2.1.

The absence of a centralized platform exacerbates these issues, as existing resources lack personalization for each student's unique requirements. As a result, students may feel stressed and uncertain about their future, especially when faced with the daunting task of navigating the complex landscape of higher education and career opportunities.

Furthermore, the current systems often fail to provide individual student reports, hindering comprehensive assessment and planning. To address these challenges, there is a pressing need for the development of a centralized, personalized, and easily accessible platform that offers tailored advice, centralized information, and individualized reports, empowering students to make informed decisions and confidently pursue their career goals.

2.2 PROPOSED SYSTEM

In this project work, chatbot offers a personalized experience by providing tailored recommendations through categorized suggestions and customizable filters. Catering to multiple users, it adapts to individual interests, skills, and goals, ensuring relevance and engagement. Moreover, its immediate responses to student inquiries save valuable time, fostering efficiency and convenience in career planning.

Accessible anytime, anywhere, our chatbot eliminates barriers like geographical location or office hours, providing a seamless platform for students to seek guidance effortlessly. By integrating career guidance into a familiar and interactive format like a chatbot, it encourages active participation from students in their career development journey, empowering them with valuable insights and resources has been depicted in Figure 2.2.

Figure 2.2 Proposed System

Furthermore, our chatbot features an in-built cutoff calculator and analysis report, offering comprehensive information on colleges and storing chat history for reference. Additionally, it provides timely notifications about competitive exams and counselling sessions, ensuring students stay informed and prepared for upcoming opportunities.

2.3 FEASIBILITY STUDY

A feasibility study is an analysis conducted to assess the practicality and viability of a proposed project or initiative. It evaluates various aspects such as technical, economic, legal, operational, and scheduling feasibility to determine whether the project is feasible or achievable. The primary objective of a feasibility study is to identify potential risks, benefits, and challenges associated with the project and to provide recommendations for decision-making.

During a feasibility study, analysts gather and analyse relevant data to assess the project's feasibility. This may include conducting market research, evaluating technical requirements, estimating costs and resources, assessing legal and regulatory compliance.

Ultimately, the feasibility study serves as a critical tool for stakeholders to evaluate the viability of a project and to make informed decisions about whether to proceed with implementation. It helps minimize risks, optimize resources, and increase the likelihood of project success by identifying potential challenges early in the planning process.

Three key considerations are involved in the feasibility analysis,

- Economic feasibility
- Technical feasibility
- Operational feasibility
- Legal and Compliance feasibility
- Scheduled feasibility

2.3.1 Economical Feasibility

Economic feasibility is a kind of cost-benefit analysis of the examined project, which assesses whether it is possible to implement it. Our project is economically feasible as it doesn’t demand any subscriptions from the users. The application is completely free to use. All the user requires is a PC with Windows OS (Operating System) installed. However, a few APIs (Application Program Interface) used by the application require subscription after a certain number of free uses.

2.3.2 Technical Feasibility

Technical feasibility study is conducted to assess the practicality and viability of a product or service before launching it. Our project is technically feasible as the technical requirements of the application can be easily met by the users. For using this application, the user just requires basic computer knowledge. All that the user has to do is open the application and say the command or type it the voice assistant will take care of the rest.

2.3.3 Operational Feasibility

Operational feasibility is a measure of how well a proposed system solves the problems and takes advantage of the opportunities identified during scope definition and how it satisfies the requirements identified in the requirements analysis in the phase of system development. As our application solves most of the problems in the existing system and adds extra features and improvements on top of that, our project is operationally feasible.

2.3.4 Legal and Compliance Feasibility

This examines whether the proposed system complies with applicable laws, regulations, and industry standards. It assesses factors such as data privacy and security requirements, intellectual property rights, and any legal or regulatory constraints that need to be considered.

2.3.5 Scheduled Feasibility

This assesses whether the proposed project can be implemented within a reasonable timeframe and whether it aligns with the existing schedule constraints. It involves evaluating whether the development, deployment, and ongoing maintenance of our project can be completed within the desired timeframe and without disrupting other ongoing projects or commitments.

CHAPTER 3

SYSTEM SPECIFICATION

3.1 HARDWARE REQUIREMENTS

The following hardware specifications are recommended for the smooth functioning of this application.

- Processor requirements (recommended)
    - Brand: AMD
    - Name: Ryzen 5 Hexa Core
    - Variant: 4500U
    - Graphic Processor: AMD Radeon Vega 8
    - Number of Cores: 6
- RAM (recommended)
    - Capacity: 8 GB
    - Type: DDR4
    - Frequency: 2666 MHz
- Storage
    - Capacity: 256 GB
- Keyboard/Touchscreen
- Internet Connection

3.2 SOFTWARE REQUIREMENTS

The following software specifications are recommended for the smooth functioning of this application.

- Operating System
    - OS Windows 11 Home
    - OS Architecture: 64 Bit
    - System Architecture: 64
- Integrated Development Environment (IDE)
    - Visual Studio Code (Vs Code)
    - Jupyter Notebook and Google Collab
- Code Maintenance
    - Git and GitHub
- Framework
    - Flask
- Database
    - MongoDB

CHAPTER 4
SOFTWARE SPECIFICATION

4.1 FRONT END

4.1.1 Front End

Front-end refers to the part of a software application or website that users directly interact with. It includes the visual interface (UI) and user experience (UX) design, created using technologies like HTML, CSS, and JavaScript. Front-end development focuses on building a user-friendly interface that presents content, collects input, and provides an enjoyable experience for users.

4.1.2 HTML

HTML, or Hypertext Mark-up Language, is the standard mark-up language used to create the structure and content of web pages. It consists of a set of tags and elements that define the structure of a document. HTML elements represent different types of content such as headings, paragraphs, links, images, and forms. By using HTML tags, developers can create web pages that are displayed in web browsers, allowing users to access and interact with information on the internet.

4.1.3 CSS

CSS, or Cascading Style Sheets, is a stylesheet language used to control the presentation and layout of HTML elements on a web page. It allows developers to define styles such as colours, fonts, sizes, margins, padding, and positioning. CSS enables the customization and styling of the user interface to achieve a desired visual design, enhancing the appearance and usability of web pages.

4.1.4 JavaScript

JavaScript (JS) is a programming language used to create interactive and dynamic behaviour on web pages. It enables developers to add functionality such as event handling, animation, form validation, DOM manipulation, and asynchronous communication with servers. JavaScript is essential for creating responsive and interactive front-end applications, enhancing the user experience and functionality of websites.

4.1.5 Bootstrap

Bootstrap is a free and open-source collection of CSS and JavaScript/jQuery code used for creating dynamic websites layout and web applications. Bootstrap is one of the most popular front-end frameworks which has really a nice set of predefined CSS codes. Bootstrap uses different types of classes to make responsive websites.

4.2 BACK END

4.2.1 Backend definition

Backend refers to the part of the application that takes the things that are required to run the application, perform computations and send the results to the front end, in order to display them. Based on the client’s request, the server will accept it and perform actions like fetching data from database and generating views and sends it to the client.

4.2.2 Node JS Server

Node.js is a runtime environment that allows developers to run JavaScript code on the server-side. It uses an event-driven, non-blocking I/O model, making it lightweight and efficient for building scalable network applications. Node.js is commonly used for building web servers, APIs, and other server-side applications. It provides a rich ecosystem of libraries and frameworks, making it popular for both backend development and building full-stack applications.

4.2.3 MongoDB

MongoDB is a NoSQL database that uses a document-oriented data model. It stores data in flexible, JSON-like documents, making it easy to represent complex hierarchical relationships. MongoDB is known for its scalability, flexibility, and performance, making it suitable for a wide range of applications, from small projects to large-scale enterprise systems. It's commonly used in web development, mobile apps, and real-time analytics due to its ability to handle large volumes of data and support for high availability and horizontal scaling.

4.2.4 Flask

Flask is a lightweight and flexible web framework for Python. It allows developers to build web applications quickly and with minimal boilerplate code. Flask provides tools, libraries, and patterns to help developers create web applications, APIs, and microservices. It is known for its simplicity, extensibility, and ease of use, making it a popular choice for projects of all sizes. With Flask, developers have the freedom to choose their preferred libraries and tools, enabling them to create customized solutions tailored to their specific needs.

CHAPTER 5
SYSTEM DESIGN

5.1 DEFINITION

System Design is the process of defining the elements of a system such as architecture, modules and components, the different interfaces of those components and the data that goes through the system.

5.2 ARCHITECTURE

An Architecture diagram is a graphical representation of a set of concepts that are part of an architecture, including their principles, elements and components, which is depicted in Figure 5.1.

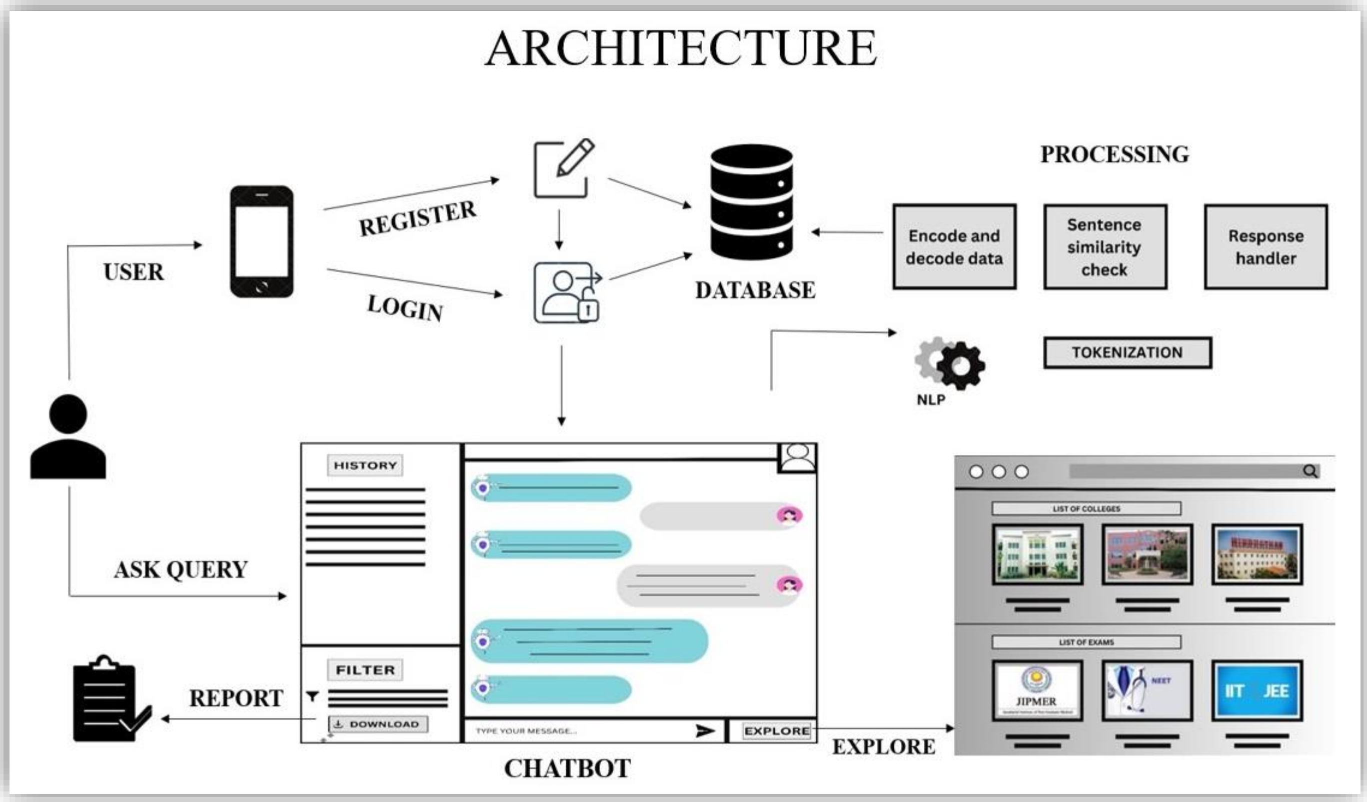
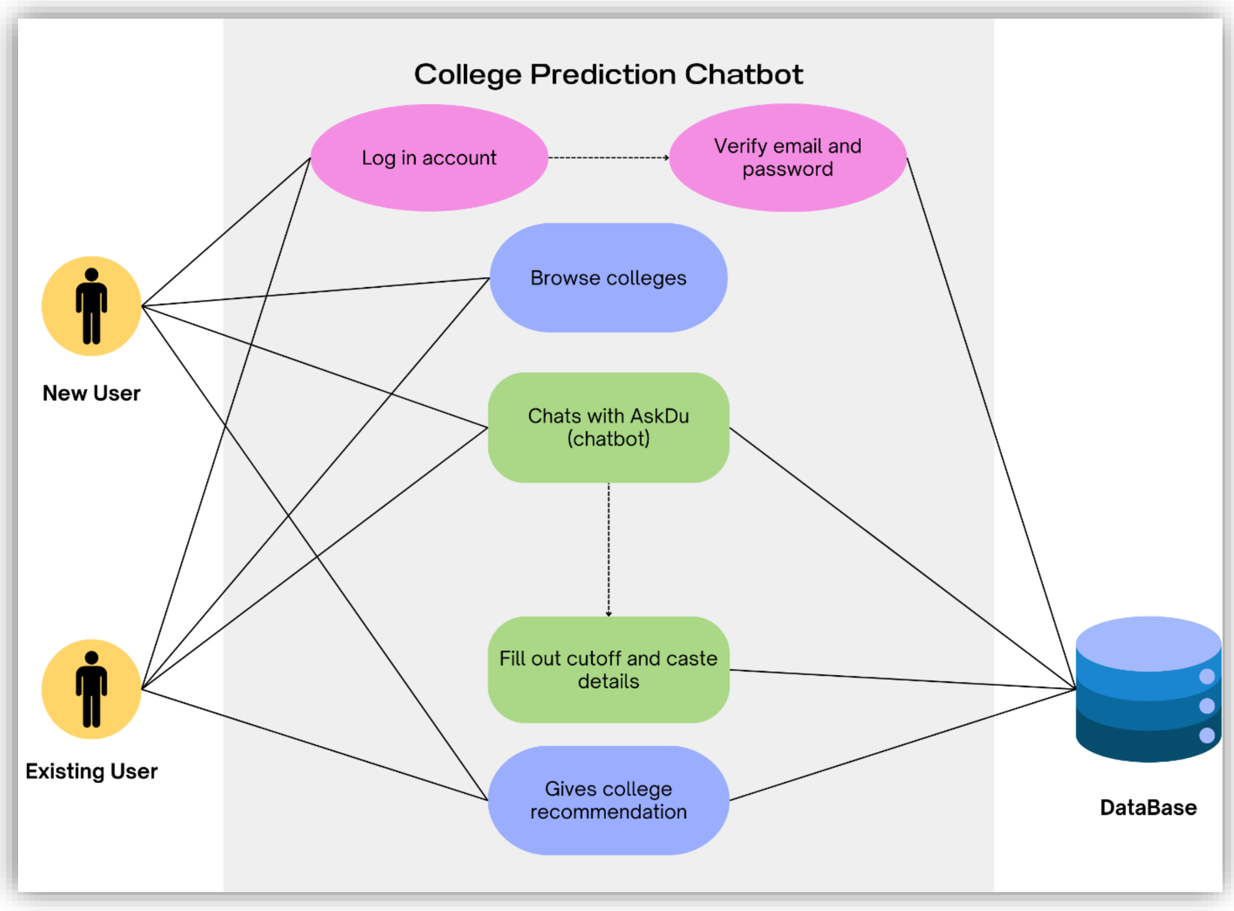


Figure 5.1 Overall Architecture

5.3 USE CASE DIAGRAMS

Use Case Diagram is a group of actors. It is a methodology used in system analysis to identify, clarify and organise system requirements. It is made up of a set of possible sequences of interaction between system and users in a particular environment and related to a particular goal. It is depicted in Figure 5.2.



College Prediction Chatbot
Log in account
Verify email and password
Browse colleges
Chats with AskDu (chatbot)
Fill out cutoff and caste details
Gives college recommendation
New User
Existing User
DataBase

Figure 5.2 Use Case Diagram

5.4 DATA FLOW DIAGRAMS

A DFD (Data-Flow Diagram) is a way of representing the flow of data of a process or a system. The DFD also provides information about the outputs and inputs of each entity and the process itself. Individuals seeking to draft a data flow diagram must identify external input and output, determine how the inputs and outputs relate to each other, and explain with graphics how these connections relate and what they result in. This type of diagram helps business development and design teams visualise how data is processed and identify or improve certain aspects. It is depicted in Figure 5.3.

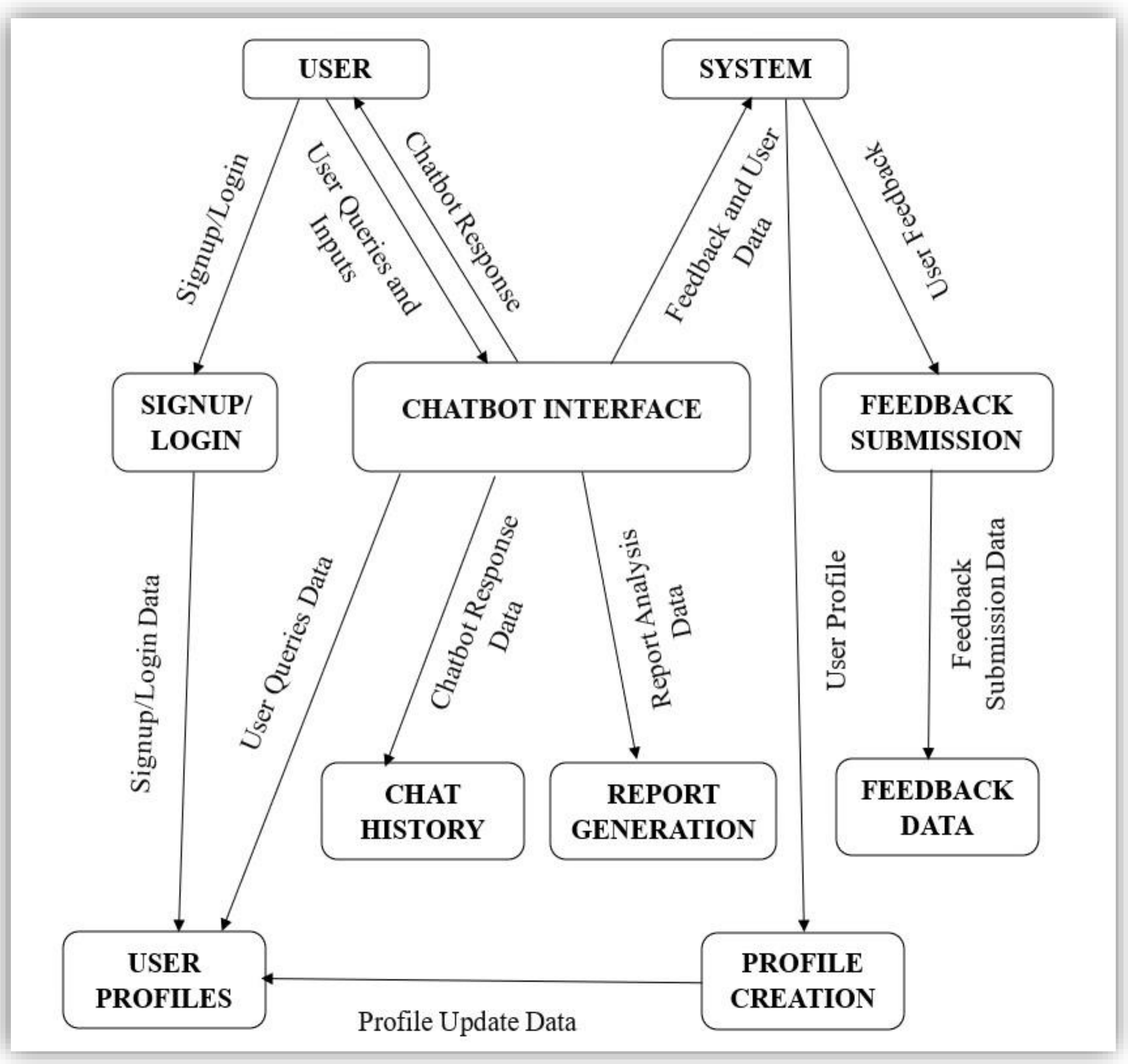


Figure 5.3. Data Flow Diagram

CHAPTER 6
MODULES

6.1 SIGNUP AND LOGIN

The Signup and Login module serves as a pivotal component within systems necessitating user authentication and access control. In this system, new users are required to sign up to create an account, providing necessary details such as username, email, and password. Once signed up, they can subsequently log in using their credentials. However, existing users can bypass the signup process and directly log in using their previously registered credentials. This setup ensures that new users go through the registration process while providing a streamlined login experience for returning users. The database of signup module and login module is depicted in Table 6.1 and Table 6.2 respectively.

Table 6.1 Database of Signup

Column Name	**Data Type**	**Description**
first_name	VARCHAR	First name of the user
last_name	VARCHAR	Last name of the user
date_of_birth	DATE	Date of birth of the user
gender	VARCHAR	Gender of the user

Column Name	Data Type	Description
email	VARCHAR	Email address of the user
contact_number	VARCHAR	Contact number of the user
new_password	VARCHAR	New password of the user
confirm_password	VARCHAR	Confirmation of the new password of the user

Table 6.2 Database of Login

Column Name	Data Type	Description
id	INT	Unique identifier
email	VARCHAR	Email address of the user
password	VARCHAR	Password of the user

6.2 MODEL TRAINING AND CHATBOT INTERFACE

This module is responsible for training recommendation models and providing an interface for the chatbot. It involves preprocessing data, selecting appropriate features, training recommendation algorithms and optimizing hyperparameters. Additionally, it allows users to interact with the recommendation system through natural language queries. The database of chat module is depicted in Table 6.3.

Table 6.3 Database of Chat

column Name	**Data Type**	**Description**
id	INT	Unique identifier
student_name	VARCHAR	Name of the user
caste	VARCHAR	Caste entered by the user
cut_off	FLOAT	Cut-off marks entered by the user
Recommended_College	VARCHAR	Recommended college based on cutoff and caste
Timestamp	Timestamp	Timestamp of the chat

6.3 CUTOFF CALCULATOR

The Cut-off Calculator module is designed to streamline the process of calculating cut-off scores for admissions or other purposes. By entering the marks obtained in physics, chemistry, and mathematics, users can quickly get their total cut-off. The module performs the necessary calculations and provides the cut-off score, helping users assess their eligibility efficiently. The database of cutoff module is depicted in Table 6.4.

Table 6.4 Database of Cutoff

Column Name	Data Type	Description
id	INT	Unique identifier
physics	FLOAT	Mark in physics
chemistry	FLOAT	Mark in chemistry
mathematics	FLOAT	Mark in mathematics
result_cutoff	FLOAT	Calculated cut-off
timestamp	TIMESTAMP	Timestamp of the entry

6.4 REPORT DOWNLOAD

The Report Download Page offers users the ability to download a comprehensive summary of their current chat session. Users can choose from various options including a full transcript of the conversation, a condensed report highlighting key insights discussed during the session, or a list of action items identified. To download the summary, users simply select their preferred option and click the download button to save the file to their device. The database of report module is depicted in Table 6.5.

Table 6.5 Database of Report

Column Name	Data Type	Description
id	INT	Unique identifier for each chat session
user_id	INT	Unique identifier for the user
bot_id	INT	Unique identifier for the chatbot
session_start	TIMESTAMP	Timestamp indicating start of session
session_end	TIMESTAMP	Timestamp indicating end of session

Column Name	Data Type	Description
message_count	INT	Number of messages exchanged during chat session
report	BLOB	Field for storing report

6.5 EXPLORE

The Explore module serves as a valuable tool for users seeking information about various colleges. Within this module, users can access comprehensive details about different colleges, including their academic programs, facilities, faculty, and campus life. Users can navigate through a user-friendly interface to search for colleges based on criteria such as location, program offerings, accreditation status, and admission requirements.

CHAPTER 7
RANDOM FOREST CLASSIFIER MODEL

7.1 RANDOM FOREST MODEL

Random Forest Classifier stands out as an ensemble learning algorithm renowned for its effectiveness in predictive modelling. By combining multiple decision trees, each trained on a random subset of the data, it harnesses the collective wisdom of these trees to deliver accurate predictions. This approach not only enhances prediction accuracy but also mitigates the risk of overfitting, a common pitfall in machine learning has been depicted in Figure 7.1.

One of the most appealing aspects of Random Forest is its versatility. Whether tasked with classification or regression, it excels in diverse scenarios, making it a go-to choose for a wide range of predictive tasks. Moreover, its innate ability to handle various types of data, including categorical and numerical, further solidifies its position as a preferred algorithm in the Machine Learning.

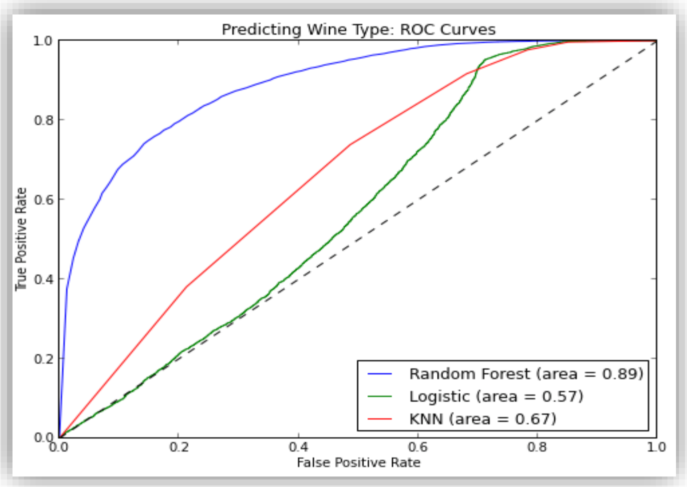


Figure 7.1 Random Forest Model

In essence, Random Forest Classifier represents a robust and reliable tool in the arsenal of data scientists and machine learning practitioners, consistently delivering high-quality predictions across different domains and applications.

7.2 STEPS INVLOVED IN TRAINING THE MODEL

7.2.1 Load the Dataset

The first step in building a Random Forest model is to load the dataset 'model_dataset.csv' into a pandas Data Frame. This dataset serves as the foundation for training the Random Forest model, containing essential information about cut-off scores, college details, and courses admitted.

7.2.2 Data Pre-Processing

Once the dataset is loaded, it's crucial to pre-process the data to ensure its quality and compatibility with the Random Forest algorithm. This involves checking for missing values and handling them appropriately to maintain data integrity. Additionally, categorical variables such as college name, college code, and course admitted need to be converted into numerical format. Techniques like one-hot encoding or label encoding are commonly employed for this purpose.

7.2.3 Split the Dataset

With pre-processing complete, the dataset is divided into features (X) and the target variable (y). Features comprise input variables like cut-off scores and college information, while the target variable encompasses the course admitted. Utilizing the train_test_split function from the sklearn library, the data is further partitioned into training and testing sets, facilitating model evaluation.

7.2.4 Train the Model

To train the Random Forest model, the RandomForestClassifier class from the sklearn.ensemble module is imported. An instance of the RandomForestClassifier object is then instantiated with desired hyperparameters, such as the number of trees and maximum depth, tailored to the specific dataset and problem at hand. Subsequently, the model is fitted to the training data using the fit method, enabling it to learn patterns and relationships within the dataset.

7.2.5 Model Evaluation

With the model trained, it's essential to evaluate its performance to assess its effectiveness in making accurate predictions. The trained model is employed to predict the classes for the test set using the predict method. The model's performance is then evaluated using classification metrics such as accuracy, precision, recall, and F1-score, providing insights into its predictive capabilities. Additionally, visualizing pertinent metrics like the confusion matrix can offer a deeper understanding of the model's strengths and areas for improvement. Random forest model architecture has been depicted in Figure 7.2.

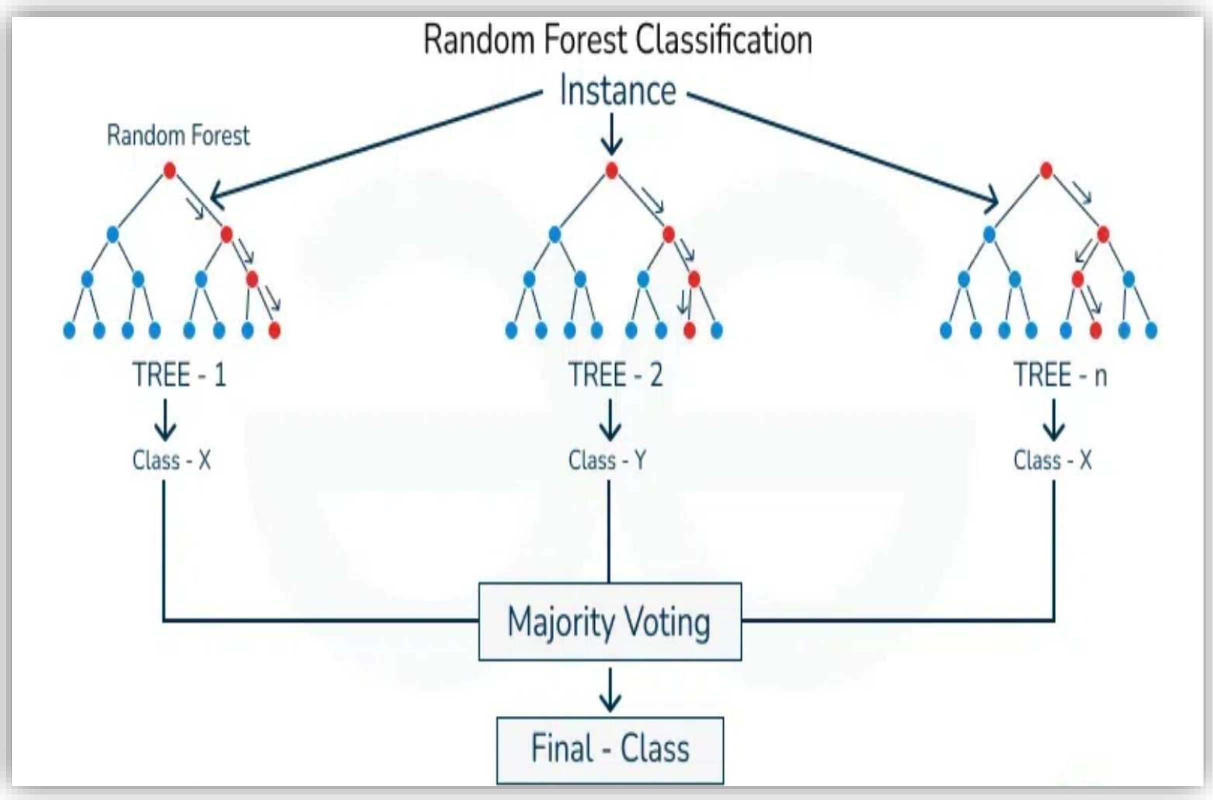


Figure 7.2 Random Forest Model Architecture

CHAPTER 8
SYSTEM TESTING

8.1 INTRODUCTION TO TESTING

Testing is a process of creating a program with the explicit intention of finding errors making the program fail. Successful test is the one that reports discovered errors. As an additional benefit, testing demonstrates that the software function appears to be working to the specification. The testing has several purposes. They are:

● To affirm the quality of the project.

● To find and eliminate any error in the program.

● To validate the software and to eliminate the operational reliability of system

The development process involves various types of testing. Each test type addresses a specific testing requirement. The most common types of testing involved in the development process are described below.

8.2 TYPES OF TESTING

8.2.1 Unit Testing

The first test in the development process is the unit test. The source code is normally divided into modules, which in turn are divided into smaller units called units. These units have specific behaviour. The test done on these units of code is called unit test.

An example of a unit test for the report download module could involve simulating a scenario where a user attempts to download a report. The test would verify that the system correctly saves the downloaded PDF file and displays a success message. This ensures that users receive confirmation when a download is successful. Unit testing has been depicted in Table 8.1.

Table 8.1 Database of Unit Testing

Test Case ID	Test Scenario	Test Steps	Expected Result	Actual Result	Status
UT001	Create user profile	Enter profile details	New user profile created successfully	New user profile created successfully	Pass
UT002	Enter signup details	New account created	Signup successful	Signup successful	Pass
UT003	Enter login details	Login existing account	Login successful	Login successful	Pass
UT004	Calculate cut-off	Enter marks in phy, chem and maths	Result cut-off displayed	Result cut-off displayed	Pass
UT005	Download report	After chat session click download report.	Report downloaded successfully	Report downloaded successfully	Pass

8.2.2 User Acceptance Testing:

User Acceptance Testing (UAT), or application testing, is the final stage of any software development or change request lifecycle before go-live. UAT meaning the final stage of any development process to determine that the software does what it was designed to do in real-world situations. User acceptance testing is a level of the software testing process where a system is tested for acceptability. The purpose of this test is to evaluate the system's compliance with the business requirements and assess whether it is acceptable for delivery.

The primary function of this chatbot is to engage in conversations with students, gather information regarding their caste and cut-off marks, and provide personalized college recommendations based on this data. Additionally, the chatbot facilitates the download of reports summarizing each chat session.

By conversing with students, our chatbot collects essential details such as caste category and cut-off marks, enabling it to offer tailored college recommendations that align with individual preferences and academic achievements. After each chat session, the chatbot generates a comprehensive report documenting the interaction, which can be downloaded for further reference or analysis.

This focused functionality ensures that students receive valuable guidance in their college selection process while also providing administrators with insightful reports to monitor and enhance the chatbot's performance over time. User acceptance testing has been depicted in Table 8.2.

Table 8.2 Database of User-Acceptance Testing

Test Case ID	**Test Scenario**	**Test Steps**	**Expected Result**	**Actual Result**	**Status**
UAT001	User Login	1. Enter valid username and password	User is successfully logged in	User logged in successfully	Pass

		2. Enter invalid username and password	User receives an error message indicating invalid login	User received an error message for invalid login	Pass
UAT002	View Explore page	Select explore button	Page is displayed	Page is displayed	Pass
UAT003	Chats with chatbot	1. Starts chat	Chatbot replies	Chatbot relies	Pass
		2. Enter cut-off and caste	Recommended colleges shown	Recommended colleges shown	Pass
UAT004	Download report	1. Click ‘save as PDF’	PDF saved successfully	PDF saved successfully	Pass
UAT005	Enter feedback	1. Go to feedback page	Feedback page displayed	Feedback page displayed	Pass
		2. Enter feedback	Feedback sent successfully	Feedback sent successfully	Pass

UAT006	Edit profile details	1. Enter profile page	Profile page displayed	Profile page displayed	Pass
		2. Edit details	Profile details edited	Profile details edited	Pass
		3. Update profile	Profile updated successfully	Profile updated successfully	Pass

CHAPTER 9
CONCLUSION AND FUTURE ENHANCEMENTS

9.1 CONCLUSION

In conclusion, AskDu is a cutting-edge tool designed to empower students in navigating the complexities of higher education choices. By leveraging AI-driven career navigation and college counseling assistance, it offers personalized recommendations and cutoff score calculations tailored to each student's preferences. Rigorous testing has been conducted to ensure reliability and accuracy in its recommendations. Moreover, the commitment to continuous improvement provides a valuable resource for students as they embark on their educational journey. With its innovative approach and user-friendly interface, it stands to revolutionize the college selection process and support students in making informed decisions about their future.

9.2 FUTURE ENHANCEMENTS

In future work, the recommendation system can be further advanced by exploring enhanced recommendation algorithms, leveraging advanced machine learning techniques and deep learning models to improve the accuracy and relevance of recommendations. Integration with real-time data sources such as college admission portals will provide users with up-to-date information and insights, enriching their decision-making process. Collaboration with educational institutions can facilitate tailored recommendations for student enrolment and career placement programs, ensuring alignment with the evolving needs of students and educational institutions. Additionally, the system can aim for global expansion, encompassing various streams, colleges, and universities across India and beyond, to cater to a diverse audience and provide relevant recommendations on a global scale.

APPENDICES

Appendix-1 (Source Code)

FLASK:

```
from flask import Flask, render_template, request, jsonify, redirect, url_for, flash
, session
from pymongo import MongoClient
from werkzeug.security import generate_password_hash, check_password_hash
import pandas as pd

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

# Connect to MongoDB
try:
    client = MongoClient('mongodb://localhost:27017/')
    db = client['chatbot']
    users_collection = db['users']
    feedback_collection = db['feedback']
    profile_collection = db['profile']
    # marks_collection = db['marks']
    print("Connected to MongoDB successfully")
except Exception as e:
    print("Error connecting to MongoDB:", e)

class User:
    def _init_(self, email, password):
        self.email = email
        self.password = password

# Login route
@app.route('/login', methods=['POST'])
def login():
    if request.method == 'POST':
        email = request.form['email']
        password = request.form['password']
        print(email,password)
        user = users_collection.find_one({'email': email})
        if user and check_password_hash(user['password'], password):
            flash('Login successful', 'success')
            return redirect(url_for('chatbot'))
        else:
            flash('Invalid email or password', 'error')
            return redirect(url_for('home'))
    else:
        return 'Method Not Allowed', 405

# Signup route
@app.route('/signup', methods=['POST'])
def signup():
    if request.method == 'POST':
        email = request.form.get('email')
        existing_user = users_collection.find_one({'email': email})
        if existing_user:
            flash('User with this email already exists. Please log in.', 'error')
            return redirect(url_for('home'))
        else:
            password = request.form.get('password')
            hashed_password = generate_password_hash(password)
            new_user = {'email': email, 'password': hashed_password}
            users_collection.insert_one(new_user)
            flash('Signup successful. Please log in.', 'success')
            return redirect(url_for('home'))
    else:
        return 'Method Not Allowed', 405

 # Existing feedback endpoint function
@app.route('/feedback-page', methods=['GET'])
def feedback_page():
    return render_template('feedback.html')

# New feedback endpoint function
@app.route('/feedback', methods=['POST'])
def submit_feedback():
    if request.method == 'POST':
        email = request.form.get('email')
        password = request.form.get('password')
        message = request.form.get('feedmessage')

        # Check if the email and password match a user in the database
        user = users_collection.find_one({'email': email})
        if user and check_password_hash(user['password'], password):
            # Store feedback in the database
            new_feedback = {'email': email, 'password': password, 'message': message}
            feedback_collection.insert_one(new_feedback)
            flash('Feedback submitted successfully.', 'success')
        else:
            flash('Invalid email or password.', 'error')

        return redirect(url_for('feedback_page'))
    else:
        return redirect(url_for('feedback_page'))

# Profile route
from flask import request

@app.route('/profile', methods=['POST'])
def create_profile():
    if request.method == 'POST':
        email = request.form.get('email')
        existing_profile = profile_collection.find_one({'email': email})
        if existing_profile:
            flash('Profile with this email already exists.', 'error')
            return redirect(url_for('home'))
        else:
            # Get profile data from the form
            first_name = request.form.get('firstName')
            last_name = request.form.get('lastName')
            date_of_birth = request.form.get('dateOfBirth')
            age = request.form.get('age')
            location = request.form.get('location')
            gender = request.form.get('gender')
            phone = request.form.get('phone')  # Get phone number data
            # Perform validation on phone number (you can add your validation logic here)
            if not phone.isdigit() or len(phone) != 10:
                flash('Please enter a valid 10-digit phone number.', 'error')
                return redirect(url_for('home'))

            # Store profile data in the profile collection
            new_profile = {
                'email': email,
                'first_name': first_name,
                'last_name': last_name,
                'date_of_birth': date_of_birth,
                'age': age,
                'location': location,
                'gender': gender,
                'phone': phone  # Include phone number in the new profile document
            }
            profile_collection.insert_one(new_profile)
            flash('Profile created successfully.', 'success')
            return redirect(url_for('home'))
    else:
        return 'Method Not Allowed', 405

# Load the dataset
df = pd.read_excel("C:/Users/sivar/OneDrive/Desktop/Mini Project Chatbot/model_deployment/dataset_chatbot.xlsx")


# Preprocess the DataFrame
```

```
col_to_drop = ['S NO', 'APPLN NO', 'NAME OF THE CANDIDATE', 'DOB', 'RANK', 'ALLOTTED\nCATEGORY']
df.drop(columns=col_to_drop, inplace=True)
df['COMMUNI\nTY'] = df['COMMUNI\nTY'].str.replace('COMMUNI\nTY', '')
df.replace('', pd.NA, inplace=True)
df.dropna(inplace=True)

# Group by relevant columns and aggregate min/max marks
co_range = df.groupby(['COLLEGE\nCODE', 'COMMUNI\nTY', 'BRANCH\nCODE']).agg({'AGGR\nMARK': ['min', 'max']})

def predict_colleges_and_branches(agg_marks, community):
    # Filter the DataFrame based on the provided community
    filtered_df = co_range.loc[(slice(None), community), :]

    # Initialize a list to store matching colleges and branches
    matches = []

    # Iterate over the filtered DataFrame
    for index, row in filtered_df.iterrows():
        college_code = index[0]
        branch_code = index[2]
        min_cutoff = float(row[('AGGR\nMARK', 'min')])  # Convert to float
        max_cutoff = float(row[('AGGR\nMARK', 'max')])  # Convert to float

        # Check if the aggregate marks fall within the range
        if min_cutoff <= float(agg_marks) <= max_cutoff:
            matches.append((college_code, branch_code))
    # Check if any matches were found
    if matches:
        # Convert the matches to a list of strings
        matches_str = [f"College code: {match[0]}, Branch: {match[1]}" for match
in matches]
        return matches_str
    else:
        return ["No colleges found for the given aggregate marks and community."]

# Store conversation messages
conversation_messages = []

agg_marks = ""
community = ""
conversation_stage = 0  # Initialize conversation stage counter



def is_valid_community(community):
    valid_communities = ['BC', 'BCM', 'MBC', 'OC', 'SC', 'SCA', 'ST']
    return community.upper() in valid_communities

# @app.route('/')
# def home():
#     return render_template('chatbot.html', messages=conversation_messages)

@app.route('/')
def home():
    return render_template('index.html')
```

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

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

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

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

@app.route('/chatbot')
def chatbot():
    return render_template('chatbot.html', messages=conversation_messages)

@app.route('/send-message', methods=['POST'])
def send_message():
    global agg_marks, community, conversation_stage

    user_message = request.form['message']
    print("User message:", user_message)
    conversation_messages.append(('user', user_message))

    # Bot response logic
    bot_response = ""

    # Handle different stages of the conversation
    if conversation_stage == 0:
        agg_marks = user_message
        print(agg_marks)
        bot_response = f"You entered aggregate marks: {agg_marks}. Please enter your community."
        conversation_stage = 1
    elif conversation_stage == 1:
        if is_valid_community(user_message):
            community = user_message.upper()
            print('comm',community)
            bot_response = f"You entered community: {community}. Please wait while I fetch the results...\nIt will take some time...."
            conversation_stage = 2
        else:
            bot_response = "Invalid community. Please enter a valid community from BC, BCM, MBC, OC, SC, SCA, ST."
    elif conversation_stage == 2:
        print("Agg marks:", agg_marks)
        print("Community:", community)
        result = predict_colleges_and_branches(agg_marks, community)
        print(result)
        if isinstance(result, list):
           bot_response = "\n".join(result)
           print(bot_response)
        else:
            bot_response = f"College code: {result[0]}, Branch: {result[1]}"
    # Send response back to the client
    jsonify_response = jsonify({'status': 'success', 'bot_messages': [bot_response]})

    # Append bot's response to conversation_messages
    conversation_messages.append(('bot', bot_response))
    print(conversation_messages)

    return jsonify_response

if _name_ == '_main_':
    app.run(debug=True)
```

HTML:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>chatbot Page</title>
    <!-- Google Fonts Link For Icons -->
    <link rel="stylesheet" href="https://fonts.googleapis.com/css2?family=Material+Symbols+Rounded:opsz,wght,FILL,GRAD@48,400,0,0">
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.2/css/all.min.css" integrity="sha512-z3gLpd7yknf1YoNbCzqRKc4qyor8gaKU1qmn+CShxbuBusANI9QpRohGBreCFkKxLhei6S9CQXFEbbKuqLg0DA==" crossorigin="anonymous" referrerpolicy="no-referrer" />
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css"
```

```
rel="stylesheet"                                         integrity="sha384-
T3c6CoIi6uLrA9TneNEoa7RxnatzjcDSCmG1MXxSR1GAsXEV/Dwwykc2M
PK8M2HN" crossorigin="anonymous">
  <link href="{{ url_for('static', filename='style.css') }}" rel="stylesheet">
  <script src="{{ url_for('static', filename='chatbotscript.js') }}" defer></script>
  <!-- pdf -->
  <script
src="https://cdnjs.cloudflare.com/ajax/libs/html2pdf.js/0.9.2/html2pdf.bundle.m
in.js"></script>
   <!-- Import jQuery -->
   <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
   <style>
     body{
       background: url("./static/chatbot\ background.jpg") center/cover no-repeat;
       border-bottom: solid 2px white;
     }
     header{
       border-bottom: solid 2px white;
       background: local;
     }
     /* history */
     .history {
     text-align: left;
     border: solid 2px white;
     color: white;
     height: 245px;
     margin: 5px;
     border-radius: 5px;
     margin-right: 7px;
     font-size: 21px;
```

```
    font-weight: 500;
    padding: 20px;
    padding-top: 23px;
    overflow-y: auto;
}

.conversation-heading {
    cursor: pointer; /* Change cursor to pointer on hover */
    padding: 10px; /* Add padding to the headings */
    margin: 5px 0; /* Add margin to separate headings */
    position: relative; /* Relative positioning for absolute icons */
    font-size: 15px;
}

.conversation-heading:hover {
    background-color: #ffffff;
    color: #cf2393;
    border-radius: 5px;
}

/* Styles for action icons */
.actions {
    position: absolute;
    top: 50%;
    right: 10px;
    transform: translateY(-50%);
    opacity: 0; /* Hide icons by default */
    transition: opacity 0.3s ease-in-out;
}
```

```
.conversation-heading:hover .actions {
   opacity: 1; /* Show icons on hover */
}

/* Styles for edit and delete icons */
.edit-icon,
.delete-icon {
   margin-left: 10px;
   color: #cf2393;
   font-size: 16px;
}

.history i{
   margin-left: 10px;
}
.history .newchat{
   text-align: center;
}
.history button{
border: none;
outline: none;
background: #fff;
color: #d12193;
font-size: 1rem;
font-weight: 600;
padding: 10px 18px;
border-radius: 3px;
cursor: pointer;
```

```
    transition: 0.15s ease;
    width: 50%;
    margin-bottom: 5px;
    }
    .history button:hover{
        background: #ddd;
}

   .filter i {
    margin-right: 10px; /* Adjust spacing between icon and text */
}
    </style>
</head>
<body>
    <div class="row">
    <header class="col-12">
        <nav class="navbar">
            <span class="hamburger-btn material-symbols-rounded">menu</span>
            <button id="goback" onclick="goBack()"><i class="fas fa-arrow-
left"></i></button>
            <a href="#" class="logo">
                <img src="./static/Screenshot (377).png" alt="logo">
                <h2>AskDu</h2>
            </a>
            <ul class="links">
                <span class="close-btn material-symbols-rounded">close</span>
                <li><a href="{{ url_for('home') }}">Home</a></li>
                <li><a href="{{ url_for('explore') }}">Explore</a></li>
                <li><a href="{{ url_for('feedback') }}">Feedback</a></li>
                <li><a href="{{ url_for('aboutus') }}">About us</a></li>
                <li><a href="{{ url_for('profile') }}">Profile</a></li>
            </ul>
        </nav>
    </header>
    </div>
    <div class="sidebar">
        <div class="history">
            <p> <i class="fas fa-history"></i> History</p>
            <!-- <div class="newchat">
                <button type="submit">New Chat</button>
            </div> -->
            <!-- <div class="conversation-popup"></div> -->
            <div id="conversation-list">
                <div class="conversation-heading">
                    Exploring Top Universities
                    <span class="actions">
                        <i class="fas fa-pen edit-icon" title="Edit"></i>
                        <i class="fas fa-trash delete-icon" title="Delete"></i>
                    </span>
                </div>
                <div class="conversation-heading">
                    Journey Through Higher Education
                    <span class="actions">
                        <i class="fas fa-pen edit-icon" title="Edit"></i>
                        <i class="fas fa-trash delete-icon" title="Delete"></i>
                    </span>
                </div>
                <div class="conversation-heading">
                    Unveiling Best Academic Pathways
                    <span class="actions">
                        <i class="fas fa-pen edit-icon" title="Edit"></i>
                        <i class="fas fa-trash delete-icon" title="Delete"></i>
                    </span>
                </div>
                <div class="conversation-heading">
                    Discovering Educational Opportunities
                    <span class="actions">
                        <i class="fas fa-pen edit-icon" title="Edit"></i>
                        <i class="fas fa-trash delete-icon" title="Delete"></i>
                    </span>
                </div>
            </div>
        </div>
        <div class="calmodule">
        <button class="cutcal">Cutoff Calculator</button>
        </div>
        <div class="filter">
            <p>Filter <i class="fas fa-filter"></i> </p>
            <div>
                <input type="radio" id="counselling" name="filterType"
    value="counselling">
                <label for="counselling">Counselling</label>
            </div>
            <div class="counselling-options" style="display: none;">
                <div>
                    <input type="checkbox" id="cutoff" name="cutoff">
                    <label for="cutoff">Cutoff</label>
                </div>
                <div>
                    <input type="checkbox" id="community" name="community">
                    <label for="community">Community</label>
                </div>
            </div>
            <div>
                <input type="radio" id="exam" name="filterType" value="exam">
                <label for="exam">Exam</label>
            </div>
        </div>

        <div class="downmodule">
        <button class="download" id="download">Download</button>
        </div>
    </div>
    <div class="chatbox">
        <div class="conversation" id="conversation">
            <div class="received-message">
                <div class="receiver-info">
                    <img src="./static/Screenshot (377).png"></img>
                    <span>AskDu</span>
                </div>
                <div class="message received">
                <div class="message-text">Welcome to AskDu! How can I assist you
today?</div>
                </div>
            </div>
            <div class="sent-message">
                <div class="sender-info">
                    <span>User</span>
                    <i class="fas fa-user"></i>
                </div>
            <div class="message sent">
                <div class="message-text">Hi, I have a question about admissions.</div>
            </div>
            </div>
            <div class="received-message">
                <div class="receiver-info">
                    <img src="./static/Screenshot (377).png"></img>
                    <span>AskDu</span>
                </div>
                <div class="message received">
                <div class="message-text">Great! Please enter your Cutoff, if you're not sure, you can use our cutoff calculator.</div>
                </div>
            </div>
            {% for message in messages %}
                {% if message[0] == 'user' %}
                    <div class="sent-message">
                        <div class="sender-info">
                            <span>User</span>
                            <i class="fas fa-user"></i>
                        </div>
                        <div class="message sent">
                            <div class="message-text">{{ message[1] }}</div>
                        </div>
                    </div>
                {% elif message[0] == 'bot' %}
                    <div class="received-message">
                        <div class="receiver-info">
                            <img src="./static/Screenshot (377).png"></img>
                            <span>AskDu</span>
                        </div>
                        <div class="message received">
                            <div class="message-text">{{ message[1] }}</div>
                        </div>
                    </div>
                {% endif %}
            {% endfor %}
          </div>
          <div id="message-form">
            <form action="" id="user-message-form">
            <div class="message-wrapper">
                <textarea id="message" rows="1" placeholder="Send a
    message...."></textarea>
                <div class="sendbtn">
                <button class="send-button" onclick="sendMessage()">Send <i
    class="fa fa-paper-plane"></i></button>
                </div>
            <div>
            </form>
          </div>
        </div>
        <!-- calculator popup -->
        <div class="cal-blur-bg-overlay"></div>
```

```
  <div class="calculator-popup">
    <span class="close-btn material-symbols-rounded">close</span>
    <div class="calcontent">
      <h2>CUTOFF CALCULATOR</h2>
      <h4>Enter Marks:</h4>
      <form action="/calculate-cutoff" id="marks-form" method="POST">
        <div class="input-field">
          <input type="number" step="0.01" id="physics-mark" name="physicsMark" required>
          <label for="physics-mark">Physics</label>
        </div>
        <div class="input-field">
          <input type="number" step="0.01" id="chemistry-mark" name="chemistryMark" required>
          <label for="chemistry-mark">Chemistry</label>
        </div>
        <div class="input-field">
          <input type="number" step="0.01" id="maths-mark" name="mathsMark" required>
          <label for="maths-mark">Maths</label>
        </div>
        <button type="submit">Calculate</button>
      </form>
      <div id="result" name="cutoffScore" ></div>
    </div>
  </div>
<!-- download popup -->
<div class="down-blur-bg-overlay"></div>
<div class="download-popup" id="download-popup">
  <span class="close-btn material-symbols-rounded">close</span>
```

```
    <div class="downcontent" id="pdf-content">
        <h2>Report</h2>
        <!-- Add your content here -->
        <div class="received-message">
            your cutoff mark
        </div>
        {% for message in messages %}
            {% if message[0] == 'user' %}
                <div class="sent-message">
                    <div class="message-text">{{ message[1] }}</div>
                </div>
            {% elif message[0] == 'bot' %}
                <div class="received-message">
                    <div class="message-text">{{ message[1] }}</div>
                </div>
            {% endif %}
        {% endfor %}
    </div>
    <button id="generate-pdf">Generate PDF</button>
</div>
{% with messages = get_flashed_messages(with_categories=true) %}
{% if messages %}
    {% for category, message in messages %}
        <script>
            alert('{{ message }}')
        </script>
    {% endfor %}
{% endif %}
{% endwith %}
```

```
<!-- JavaScript code to handle printing as PDF -->
<script>
  document.getElementById('generate-pdf').addEventListener('click', function()
{
    // Select the element containing the content you want to convert to PDF
    const element = document.getElementById('pdf-content');

    // Options for PDF generation (optional)
    const options = {
      margin: 0.5,
      filename: 'chatbot_analysis_report.pdf',
      image: { type: 'jpeg', quality: 0.98 },
      html2canvas: { scale: 2 },
      jsPDF: { unit: 'in', format: 'letter', orientation: 'portrait' }
    };

    // Generate PDF from the selected element
    html2pdf().from(element).set(options).save();
  });

// filter
document.addEventListener('DOMContentLoaded', function() {
  var counsellingRadio = document.getElementById('counselling');
  var counsellingOptions = document.querySelector('.counselling-options');
  var examRadio = document.getElementById('exam');

  counsellingRadio.addEventListener('change', function() {
    if (counsellingRadio.checked) {
      counsellingOptions.style.display = 'block';
```

```
        } else {
            counsellingOptions.style.display = 'none';
        }
    });

    examRadio.addEventListener('change', function() {
        if (examRadio.checked) {
            counsellingOptions.style.display = 'none';
        }
    });
});

// // history
// // JavaScript code for edit and delete functionality
document.addEventListener('DOMContentLoaded', function() {
    // Function to handle edit icon click
    function handleEditClick(event) {
        // Get the conversation heading element
        var conversationHeading = event.target.closest('.conversation-heading');

        // Get the ID of the conversation
        var conversationId = conversationHeading.dataset.id;

        // Show an alert message for editing
        var newMessage = prompt('Enter the new message:');
        if (newMessage !== null) {
            conversationHeading.textContent = newMessage;
        }
    }
```

```
// Function to handle delete icon click
function handleDeleteClick(event) {
    // Get the conversation heading element
    var conversationHeading = event.target.closest('.conversation-heading');

    // Get the message of the conversation
    var message = conversationHeading.textContent;

    // Confirm before deleting
    var confirmDelete = confirm(Are you sure you want to delete the
conversation ?);
    if (confirmDelete) {
        // Remove the conversation heading
        conversationHeading.remove();
    }
}

// Attach event listeners to edit icons
var editIcons = document.querySelectorAll('.edit-icon');
editIcons.forEach(function(icon) {
    icon.addEventListener('click', handleEditClick);
});

// Attach event listeners to delete icons
var deleteIcons = document.querySelectorAll('.delete-icon');
deleteIcons.forEach(function(icon) {
    icon.addEventListener('click', handleDeleteClick);
});
```

```
});
</script>
</body>
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.j
s" integrity="sha384-
C6RzsynM9kWDrMNeT87bh95OGNyZPhcTNXj1NW7RuBCsyN/o0jlpcV8Q
yq46cDfL" crossorigin="anonymous"></script>
</html>
```

CSS:

```
@import
url("https://fonts.googleapis.com/css2?family=Open+Sans:wght@400;500;600;
700&display=swap");
* {
   margin: 0;
   padding: 0;
   box-sizing: border-box;
   font-family: "Open Sans", sans-serif;
}
body {
   height: 100vh;
   width: 100%;
   /* background: url("chatbot\ background.jpg") center/cover no-repeat; */
}
/* The side navigation menu */
.sidebar {
   margin: 0;
   padding: 0;
   width: 330px;
```

```
  left: 5px;
  border-right: solid 2px white;
  position: fixed;
  height: 100%;
  overflow: auto;
  top: 100px;
  border-radius: 5px;
}
.sidebar .cutcal {
  border: none;
  outline: none;
  background: #fff;
  color: #d12193;
  font-size: 1rem;
  font-weight: 600;
  padding: 8px 15px;
  border-radius: 3px;
  cursor: pointer;
  transition: 0.15s ease;
}
.sidebar .cutcal:hover {
  background: #ddd;
}
.calmodule{
  text-align: center;
  margin: 2px;
}
.sidebar .download{
  border: none;
  outline: none;
  background: #fff;
  color: #d12193;
  font-size: 1rem;
  font-weight: 600;
  padding: 8px 15px;
  border-radius: 3px;
  cursor: pointer;
  transition: 0.15s ease;
 }
 .sidebar .download:hover{
  background: #ddd;
 }
 .downmodule{
  text-align: center;
  margin: 2px;
 }
/* chatbox */
.chatbox {
  margin: 0;
  padding: 0;
  right: 5px;
  width: 1190px;
  border-right: solid 2px white;
  position: fixed;
  height: 100%;
  overflow: auto;
  top: 100px;
  border-radius: 5px;
```

```
    background-color: white;
    bottom: 15px;
  }

.conversation{
    height: 520px;
    border: solid 2px #d12193a3;
    margin: 10px;
    border-radius: 5px;
    overflow-y: auto;
    padding: 10px;
  }

.message {
    margin-bottom: 10px;
    padding: 18px;
    border-radius: 30px;
    max-width: 50%; /* Adjust as needed */
    clear: both;
}
.received-message, .sent-message{
    margin: 20px;
    /* animation: slideIn 0.5s ease forwards; */
}
.received {
    background-color: #f0f0f0;
    border-top-left-radius: 0px;
    align-self: flex-start;
    float: left;
    animation: slideIn 0.5s ease forwards;
}
.sent {
    background-color: #e95fb9eb;
    color: white;
    border-bottom-right-radius: 0px;
    align-self: flex-end;
    float: right;
    animation: slideIn 1s ease forwards;
}
@keyframes slideIn {
    0% {
        opacity: 0;
        transform: translateY(20px);
    }
    100% {
        opacity: 1;
        transform: translateY(0);
    }
}
.receiver-info{
    display: flex;
    align-items: center;
    margin-bottom: 8px;
    float: left;
}

.receiver-info img{
    width: 50px;
    padding: 8px;
}
.sender-info {
    display: flex;
    align-items: center;
    float: right;
    margin-bottom: 5px;
}
.sender-info i {
    font-size: 21px;
    margin: 10px;
    color: #646161;
}
.conversation span {
    font-weight: bold;
    font-size: 20px;
}

.message-text {
    word-wrap: break-word; /* Wrap long words */
}

#message-form {
    margin: 0 auto;
    width: 100%;
    box-sizing: border-box;
    max-width: 800px;
    text-align: center;
    padding: 0px 25px 0 25px;
    box-shadow: grey;
 }
.message-wrapper {
    position: relative;
}

#message::placeholder {
    color: rgb(44, 44, 44);
}
#message {
    background: white;
    border-radius: 50px;
    width: 100%;
    box-sizing: border-box;
    resize: none;
    padding: 10px 65px 14px 15px;
    font-family: inherit;
    font-size: 1em;
    box-shadow: rgba(0,0,0,0.2) 0 0 45px;
    outline: none;
    border: solid 1px rgb(169, 169, 169);
}
.send-button {
    position: absolute;
    right: 4px;
    top: 50%;
    transform: translateY(-50%);
    background: #e139a6eb;
    border-radius: 5px;
    display: inline-block;
    font-size: 1em;
    padding: 12px 50px;
    color: white;
    border: none;
    margin: -3px;
    border-radius: 50px;
    line-height: 1.5;
    text-align: center;
    width: 180px;
}
button.send-button:hover {
    border: solid 2px #d12193;
    background: white;
    color: #d12193;
}
```

JAVASCRIPT:

```
const navbarMenu = document.querySelector(".navbar .links");
const hamburgerBtn = document.querySelector(".hamburger-btn");
const hideMenuBtn = navbarMenu.querySelector(".close-btn");
// Show mobile menu
hamburgerBtn.addEventListener("click", () => {
    navbarMenu.classList.toggle("show-menu");
});
// Hide mobile menu
hideMenuBtn.addEventListener("click", () =>  hamburgerBtn.click());
// calculator popup
```

```
const showCalPop = document.querySelector(".cutcal");

const calculatorPopup = document.querySelector(".calculator-popup");

showCalPop.addEventListener("click", () => {
   document.body.classList.toggle("show-calpopup");
});

const hidecalPop = document.querySelector(".calculator-popup .close-btn");
hidecalPop.addEventListener("click", () => showCalPop.click());

// cutoff calculator result
// Select form and result element
const marksForm = document.getElementById('marks-form');
const resultElement = document.getElementById('result');

// Add event listener to form submit
marksForm.addEventListener('submit', function(event) {
   // Prevent default form submission
   event.preventDefault();

   // Get marks entered by the user
   const physicsMark = parseFloat(document.getElementById('physics-mark').value);
   const chemistryMark = parseFloat(document.getElementById('chemistry-mark').value);
   const mathsMark = parseFloat(document.getElementById('maths-mark').value);

   // Calculate cutoff score
```

```
    if (isValidMark(physicsMark) && isValidMark(chemistryMark) && isValidMark(mathsMark)) {
        // Calculate cutoff score
        const cutoffScore = (physicsMark / 2) + (chemistryMark / 2) + mathsMark;

        // Display result on the screen
        resultElement.textContent = Cutoff Score: ${cutoffScore.toFixed(2)};
    } else {
        // Display error message if marks are not within the range
        resultElement.textContent = "Please enter valid marks between 0 and 100.";
    }
});
// Function to validate marks
function isValidMark(mark) {
    return !isNaN(mark) && mark >= 0 && mark <= 100;
}
// download popup
const showdownPop = document.querySelector(".download");

const downloadPopup = document.querySelector(".download-popup");

showdownPop.addEventListener("click", () => {
    document.body.classList.toggle("show-downpopup");
});

const hidedownPop = document.querySelector(".download-popup .close-btn");
hidedownPop.addEventListener("click", () => showdownPop.click());
```

```
 // clearing input fields
const closeBtn = document.querySelector(".close-btn");

// Add event listener to close button
closeBtn.addEventListener("click", () => {
    // Reload the website
    location.reload(true);
});

// go back
function goBack() {
    window.history.back();
}

function sendMessage() {
    var userInput = document.getElementById("message").value.trim();
    console.log("User input:", userInput);
    if (userInput !== "") {
        addUserMessage(userInput);
        sendUserMessage(userInput);
        document.getElementById("message").value = "";
    }
}

function addUserMessage(message) {
    var conversation = document.getElementById("conversation");
    var sentMessageDiv = document.createElement("div");
    sentMessageDiv.className = "sent-message";
    sentMessageDiv.innerHTML = `
```

```
        <div class="sender-info">
            <span>User</span>
            <i class="fas fa-user"></i>
        </div>
        <div class="message sent">
            <div class="message-text">${message}</div>
        </div>`;
    conversation.appendChild(sentMessageDiv);
}

function addBotMessage(message) {
    var conversation = document.getElementById("conversation");
    var receivedMessageDiv = document.createElement("div");
    receivedMessageDiv.className = "received-message";
    receivedMessageDiv.innerHTML = `
        <div class="receiver-info">
            <img src="./static/Screenshot (377).png" alt="Bot Image">
            <span>AskDu</span>
        </div>
        <div class="message received">
            <div class="message-text">${message}</div>
        </div>`;
    conversation.appendChild(receivedMessageDiv);
}

function sendUserMessage(message) {
    var formData = new FormData();
    formData.append("message", message);
    fetch("/send-message", {
            method: "POST",
            body: formData
        })
        .then(response => response.json())
        .then(data => {
            if (data.status === "success") {
                var botMessages = data.bot_messages;
                botMessages.forEach(message => addBotMessage(message));
            } else {
                console.error("Error sending user message:", data.error);
            }
        })
        .catch(error => {
            console.error("Error sending user message:", error);
        });
}
// to add download content
function addMessageToDownloadContent(message, messageType) {
    var downloadContent = document.getElementById("pdf-content");
    var messageDiv = document.createElement("div");

    if (messageType === 'user') {
        messageDiv.className = "sent-message";
    } else if (messageType === 'bot') {
        messageDiv.className = "received-message";
    }

    messageDiv.innerHTML = `
```

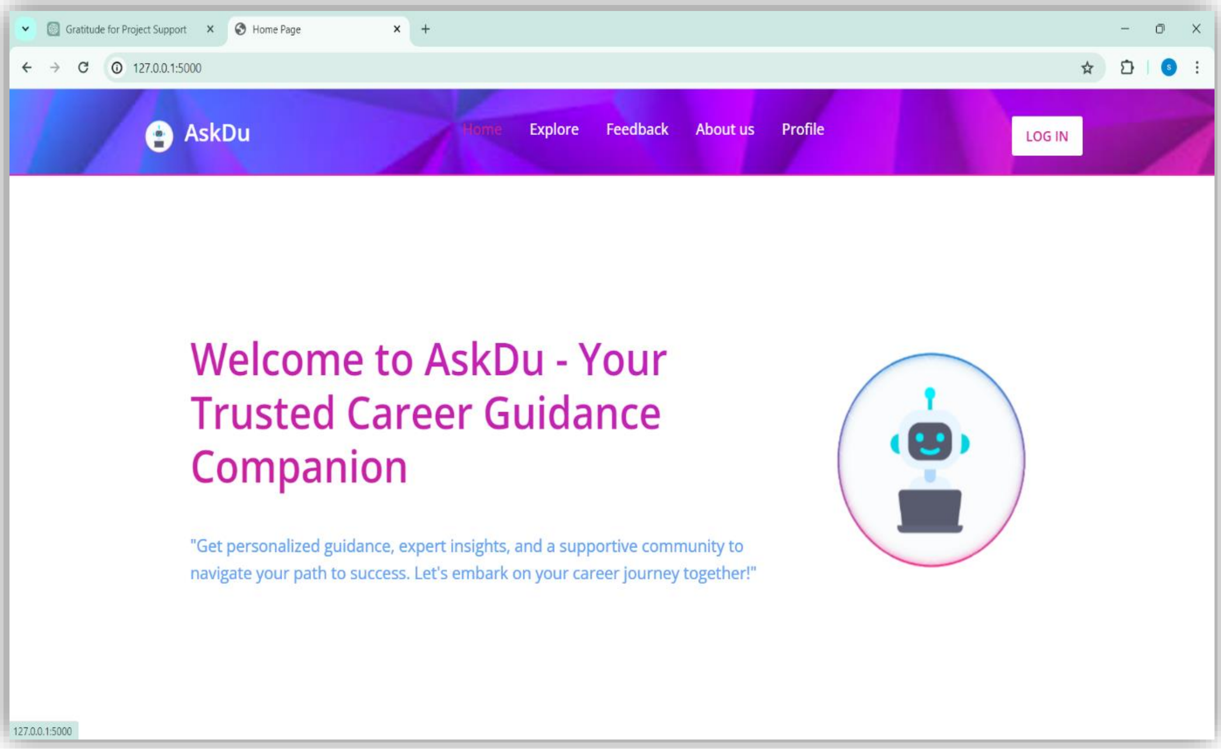
```
        <div class="message-text">${message}</div>`;

    downloadContent.appendChild(messageDiv);
}
function sendUserMessage(message) {
    var formData = new FormData();
    formData.append("message", message);

    fetch("/send-message", {
            method: "POST",
            body: formData
        })
        .then(response => response.json())
        .then(data => {
            if (data.status === "success") {
                var botMessages = data.bot_messages;
                botMessages.forEach(message => {
                    addBotMessage(message);
                    addMessageToDownloadContent(message, 'bot');
                });
            } else {
                console.error("Error sending user message:", data.error);
            }
        })
        .catch(error => {
            console.error("Error sending user message:", error);
        });
}
```

Appendix-2 (Screenshots)

SINGUP AND LOGIN MODULE

- This module handles user authentication and authorization processes.
- Features include user signup, login, logout, password reset, and account management functionalities.
- It ensures secure access to the system by authenticating users based on their credentials.

Home Page

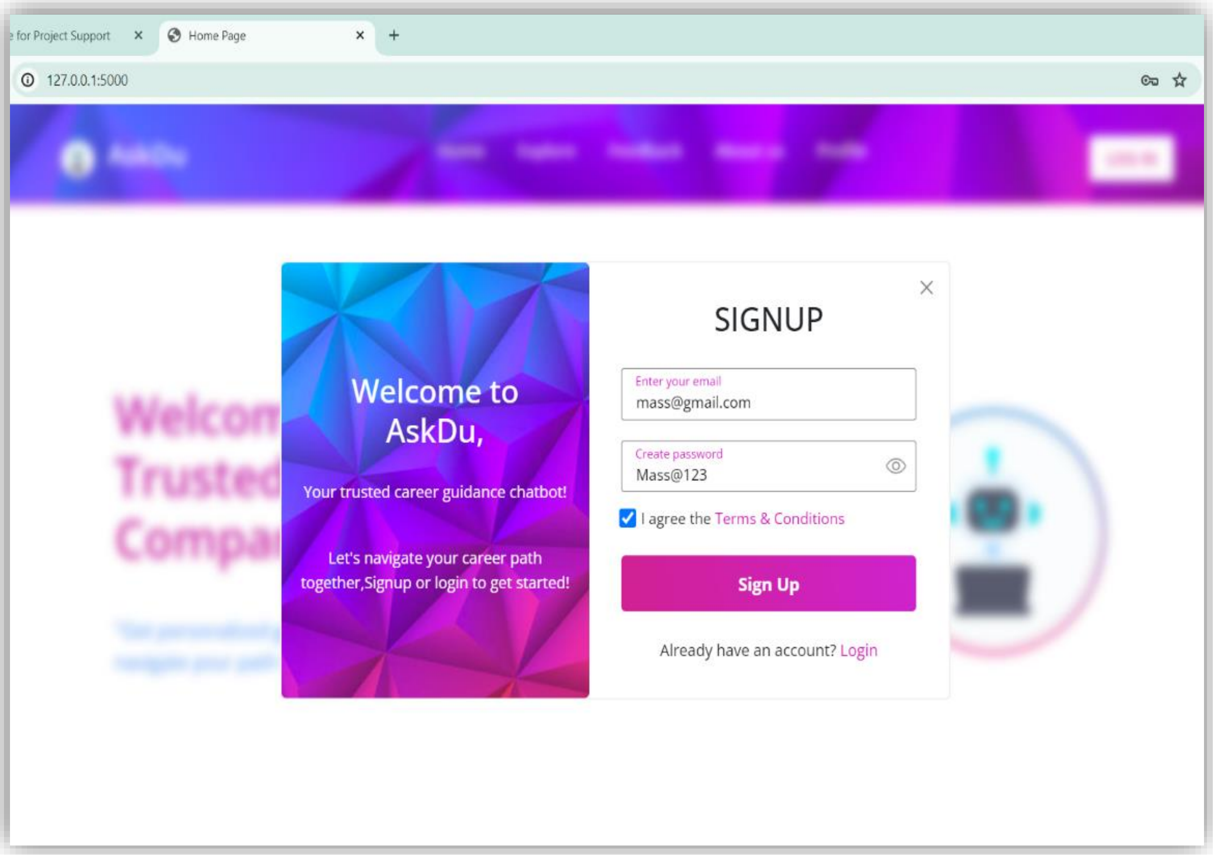


Welcome to AskDu - Your Trusted Career Guidance Companion

"Get personalized guidance, expert insights, and a supportive community to navigate your path to success. Let's embark on your career journey together!"

Signup Page

Here, new users can easily create an account to access personalized career guidance and college planning resources, empowering them in their academic and professional journeys



Welcome to AskDu,

Your trusted career guidance chatbot!

Let's navigate your career path together,Signup or login to get started!

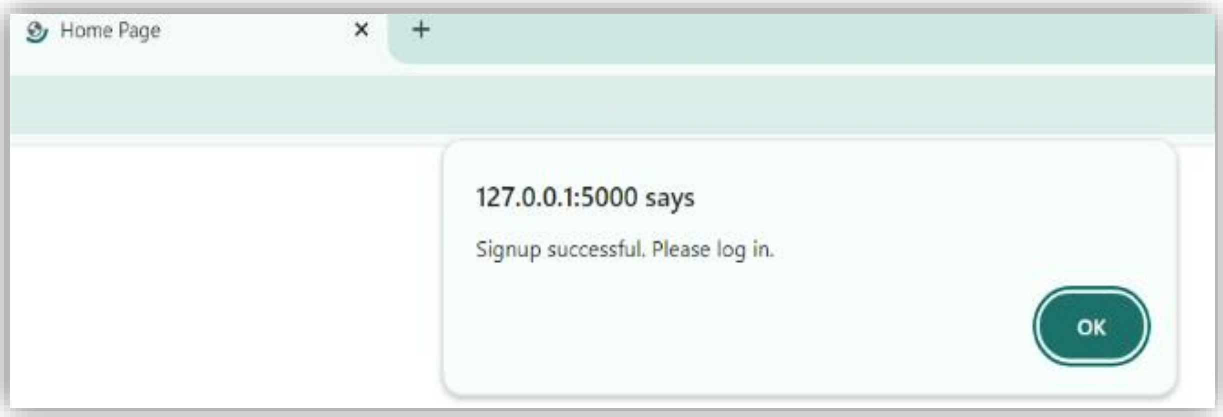
Enter your email: mass@gmail.com

Create password: Mass@123

[✓] I agree the Terms & Conditions

Sign Up

Already have an account? Login

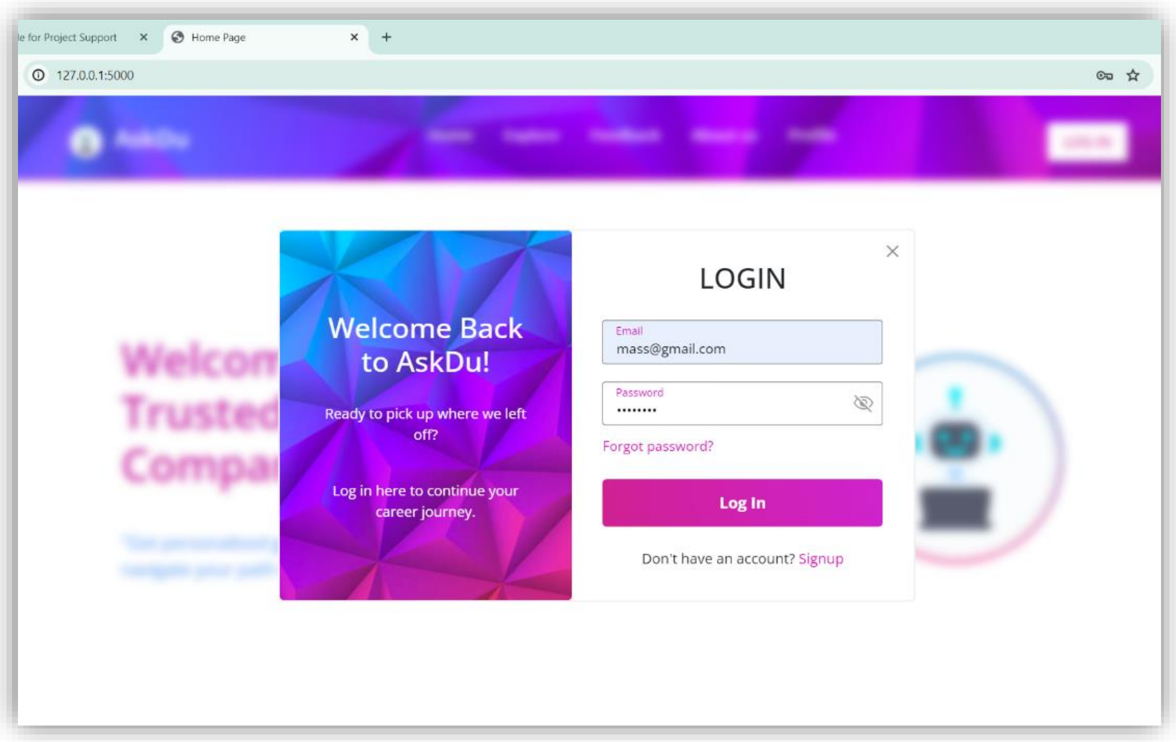


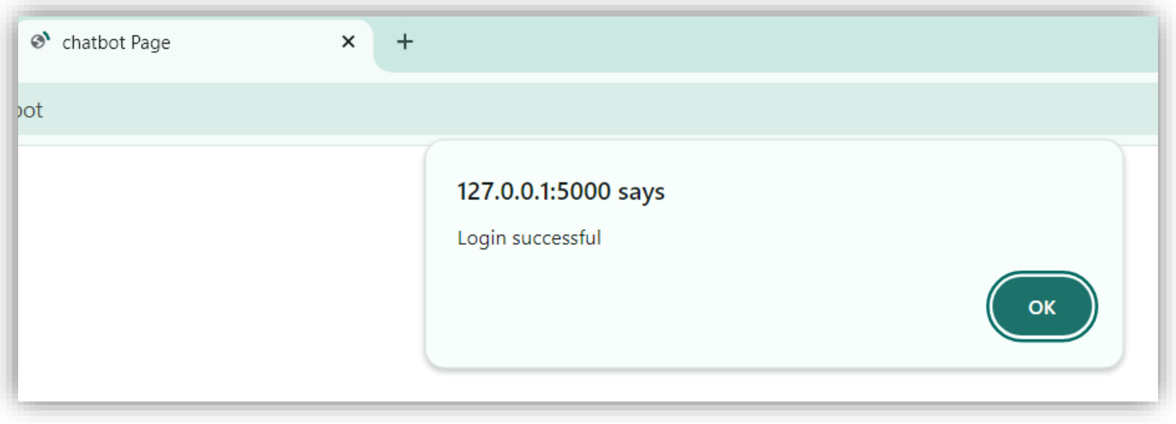
127.0.0.1:5000 says

Signup successful. Please log in.

Login Page

Existing users can quickly log in to our platform to resume their journey of accessing tailored career guidance and college planning support, ensuring a seamless experience in achieving their academic and professional goals.





127.0.0.1:5000 says

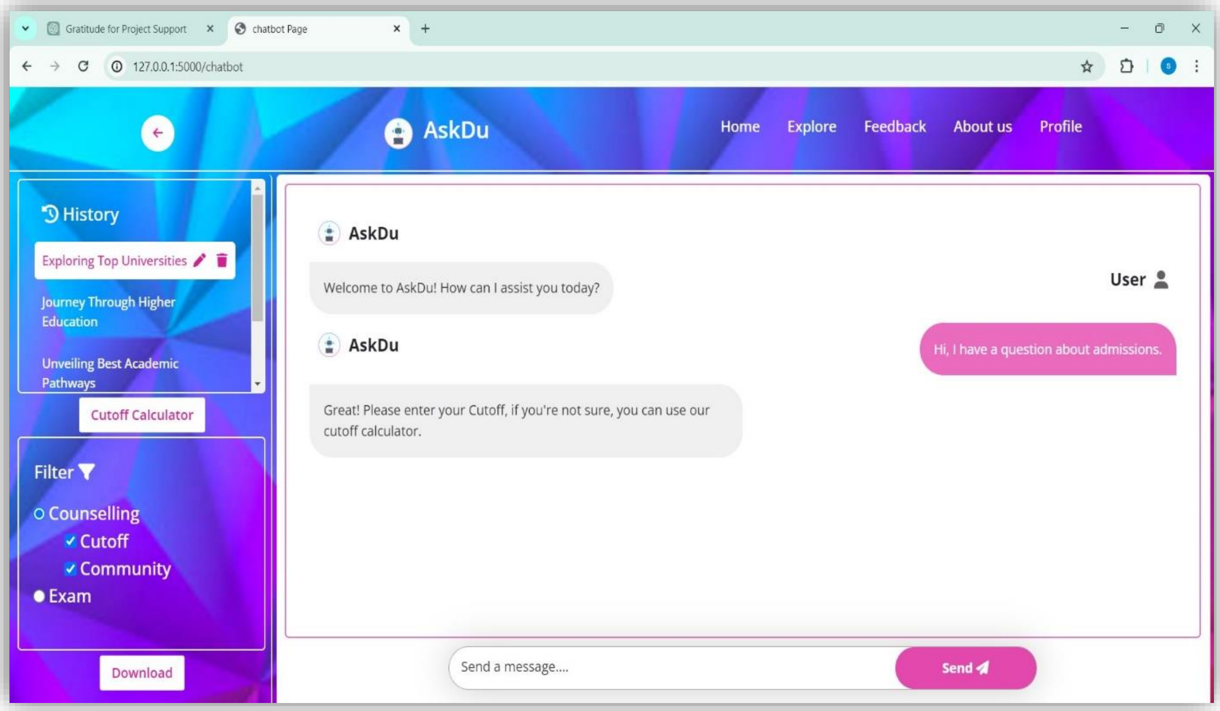
Login successful

CHATBOT MODULE

- This module is responsible for training recommendation models and providing an interface for the chatbot.
- It involves preprocessing data, selecting appropriate features, training recommendation algorithms (e.g., content-based, collaborative filtering), and optimizing hyperparameters.
- Additionally, it allows users to interact with the recommendation system through natural language queries.

Chatbot page

It’s features including customizable filters and chat history functionalities, allowing you to tailor your experience to suit your specific needs. Edit and modify titles as needed, and easily delete features when they are no longer necessary.



AskDu

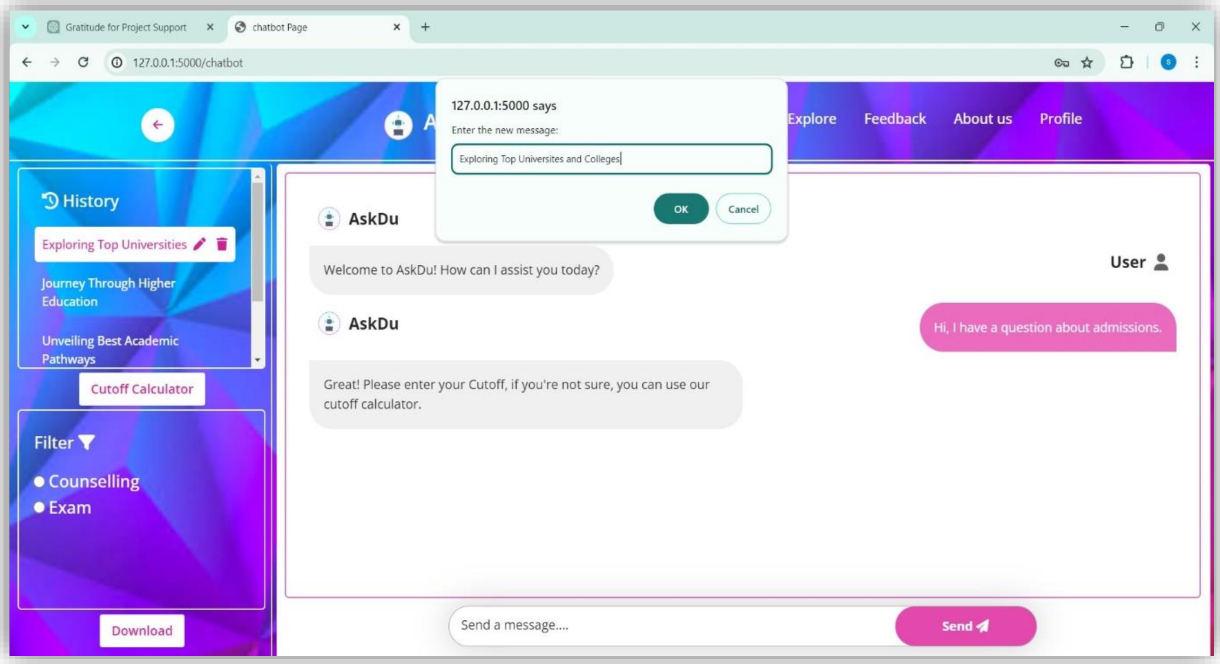
Welcome to AskDu! How can I assist you today?

User

Hi, I have a question about admissions.

AskDu

Great! Please enter your Cutoff, if you're not sure, you can use our cutoff calculator.



127.0.0.1:5000 says

Enter the new message:

Exploring Top Universities and Colleges

AskDu

Welcome to AskDu! How can I assist you today?

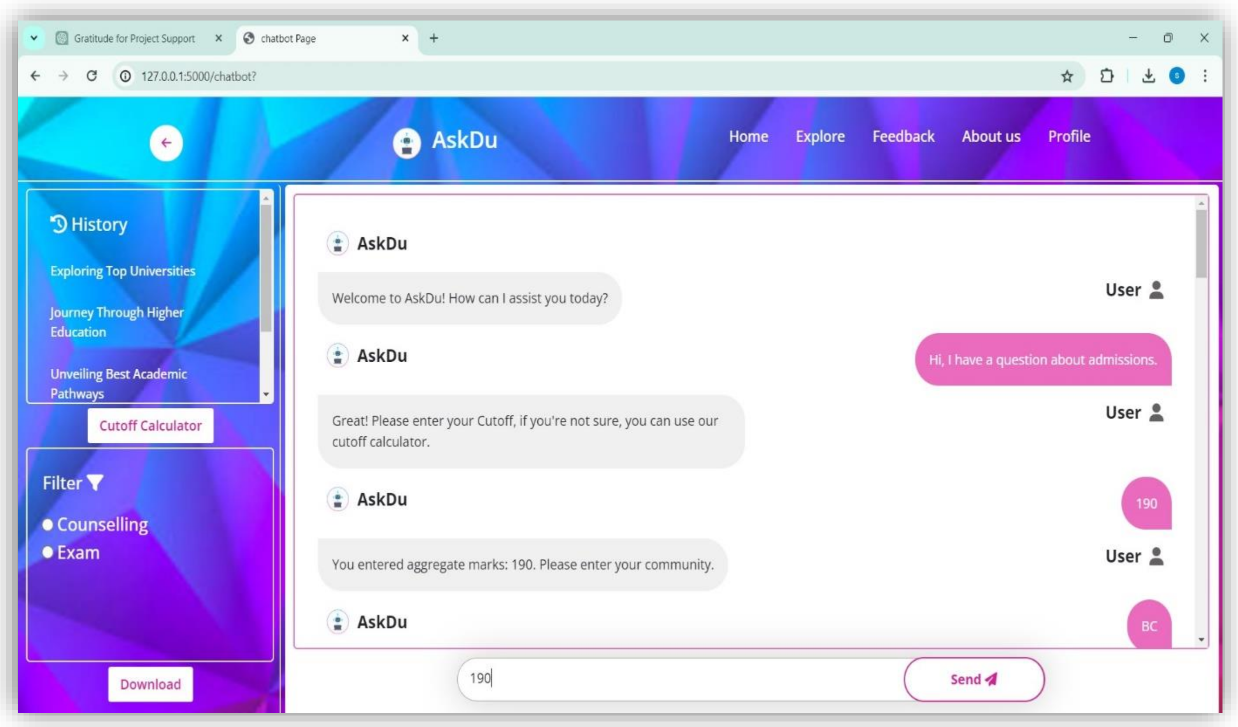
User

Hi, I have a question about admissions.

AskDu

Great! Please enter your Cutoff, if you're not sure, you can use our cutoff calculator.

Engage with our virtual career advisor to receive tailored advice based on your unique profile, simplifying the complexity of college planning and career exploration and get instant responses to your inquiries and access real-time assistance, empowering you to make informed decisions about your academic and professional future.



AskDu

Welcome to AskDu! How can I assist you today?

User

Hi, I have a question about admissions.

AskDu

Great! Please enter your Cutoff, if you're not sure, you can use our cutoff calculator.

User

190

AskDu

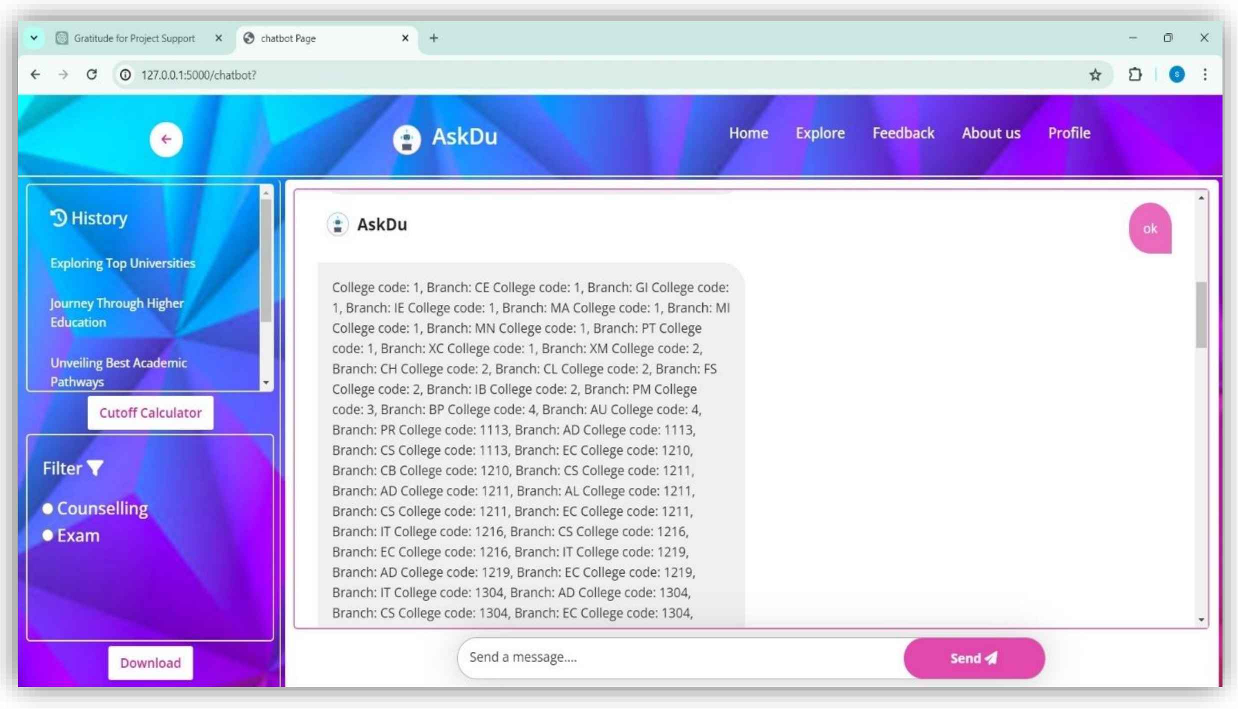
You entered aggregate marks: 190. Please enter your community.

User

BC

AskDu

190



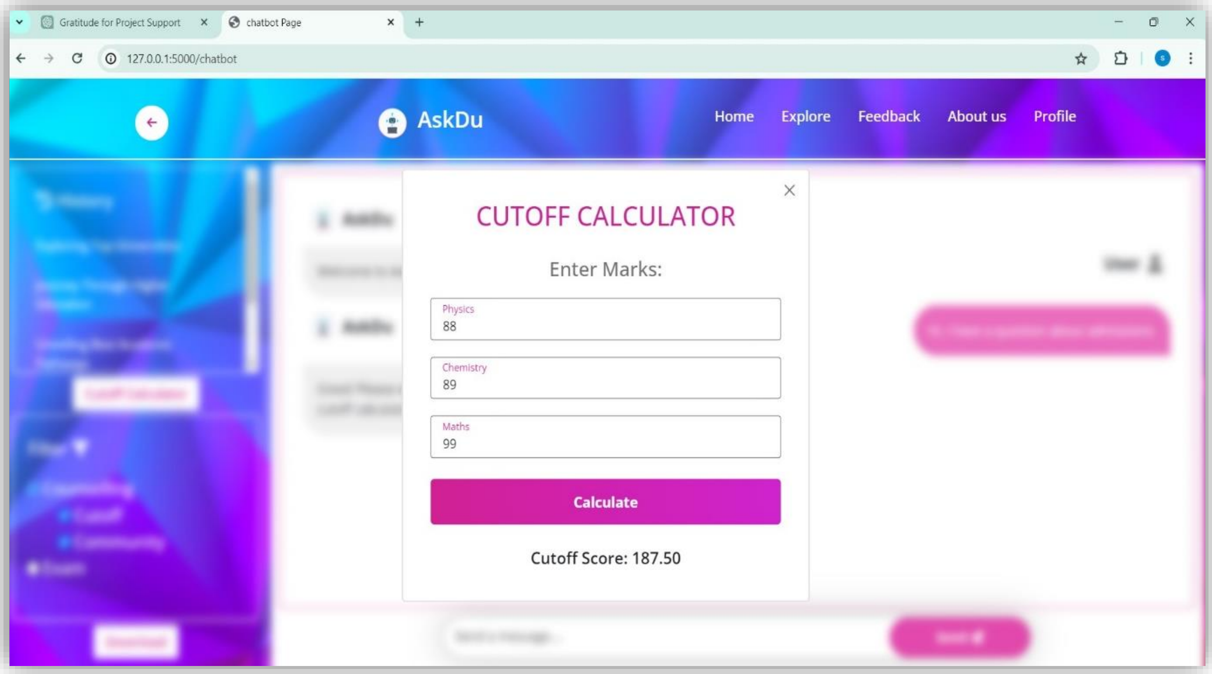
AskDu

ok

College code: 1, Branch: CE College code: 1, Branch: GI College code: 1, Branch: IE College code: 1, Branch: MA College code: 1, Branch: MI College code: 1, Branch: MN College code: 1, Branch: PT College code: 1, Branch: XC College code: 1, Branch: XM College code: 2, Branch: CH College code: 2, Branch: CL College code: 2, Branch: FS College code: 2, Branch: IB College code: 2, Branch: PM College code: 3, Branch: BP College code: 4, Branch: AU College code: 4, Branch: PR College code: 1113, Branch: AD College code: 1113, Branch: CS College code: 1113, Branch: EC College code: 1210, Branch: CB College code: 1210, Branch: CS College code: 1211, Branch: AD College code: 1211, Branch: AL College code: 1211, Branch: CS College code: 1211, Branch: EC College code: 1211, Branch: IT College code: 1216, Branch: CS College code: 1216, Branch: EC College code: 1216, Branch: IT College code: 1219, Branch: AD College code: 1219, Branch: EC College code: 1219, Branch: IT College code: 1304, Branch: AD College code: 1304, Branch: CS College code: 1304, Branch: EC College code: 1304,

CUTOFF CALCULATOR MODULE

- Users can input their academic performance and preferences to estimate their eligibility for admission to colleges.
- The module dynamically computes cutoff scores based on user-provided marks.
- Recommendations are provided to users based on calculated cutoff scores, assisting them in identifying colleges where they are likely to meet admission criteria.



Physics 88

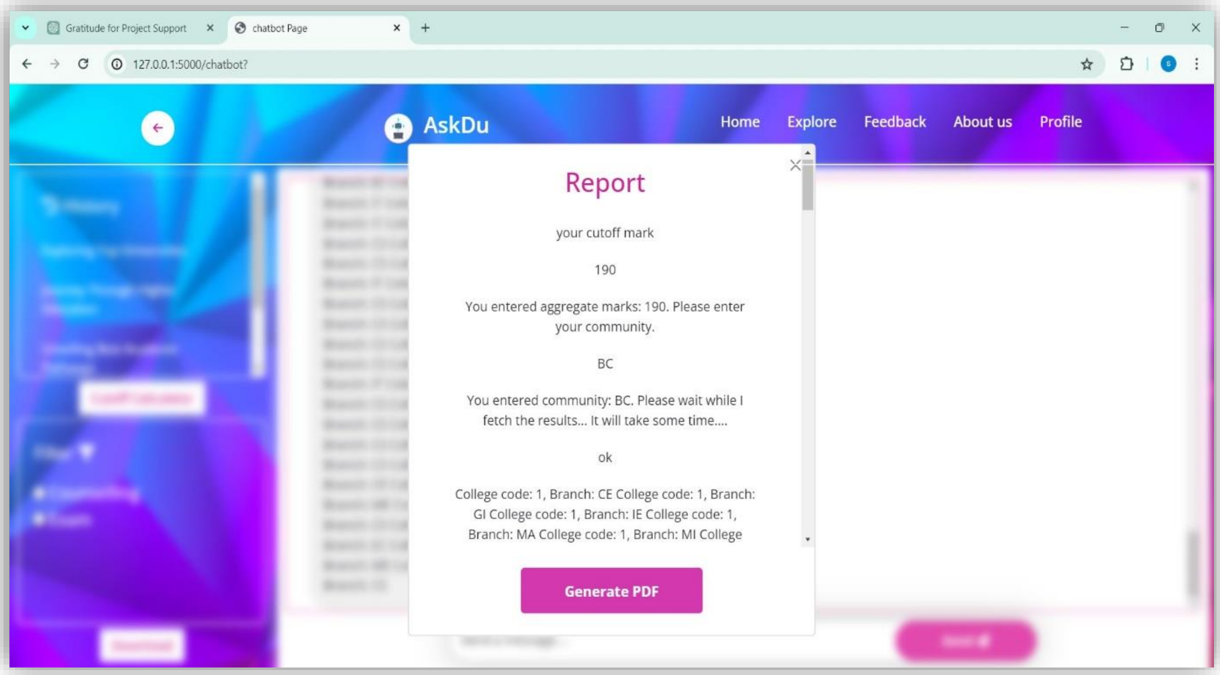
Chemistry 89

Maths 99

Cutoff Score: 187.50

REPORT DOWNLOAD MODULE

- This module allows users to download reports or summaries of their interactions with the system.
- Users can generate and download reports containing information such as recommended colleges, user preferences, chatbot conversations, etc.
- It provides users with a way to access and review their activity within the system or share it with others as needed.



Report

your cutoff mark

190

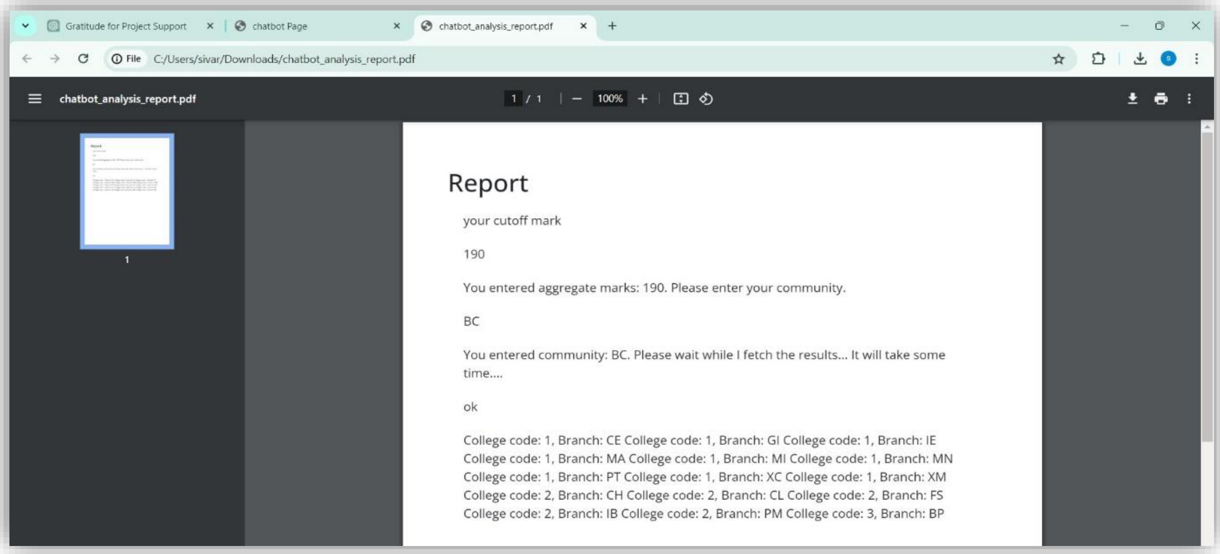
You entered aggregate marks: 190. Please enter your community.

BC

You entered community: BC. Please wait while I fetch the results.... It will take some time....

ok

College code: 1, Branch: CE College code: 1, Branch: GI College code: 1, Branch: IE College code: 1, Branch: MA College code: 1, Branch: MI College



Gratitude for Project Support
chatbot Page
chatbot_analysis_report.pdf
C:/Users/sivar/Downloads/chatbot_analysis_report.pdf
1 / 1
100%

Report

your cutoff mark

190

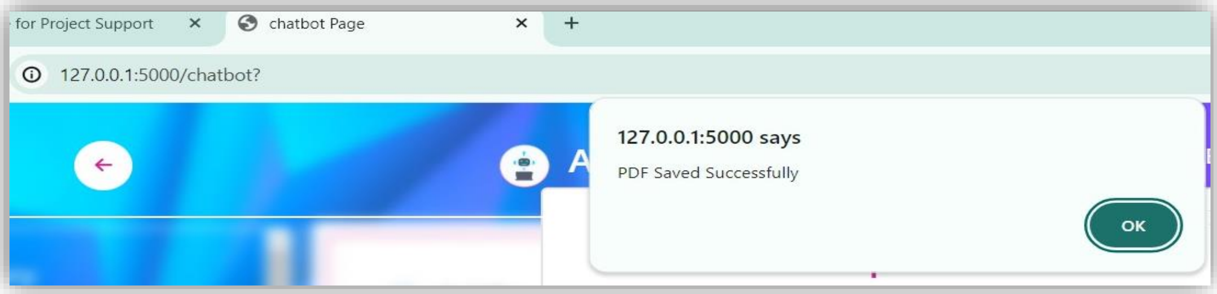
You entered aggregate marks: 190. Please enter your community.

BC

You entered community: BC. Please wait while I fetch the results.... It will take some time....

ok

College code: 1, Branch: CE College code: 1, Branch: GI College code: 1, Branch: IE College code: 1, Branch: MA College code: 1, Branch: MI College code: 1, Branch: MN College code: 1, Branch: PT College code: 1, Branch: XC College code: 1, Branch: XM College code: 2, Branch: CH College code: 2, Branch: CL College code: 2, Branch: FS College code: 2, Branch: IB College code: 2, Branch: PM College code: 3, Branch: BP

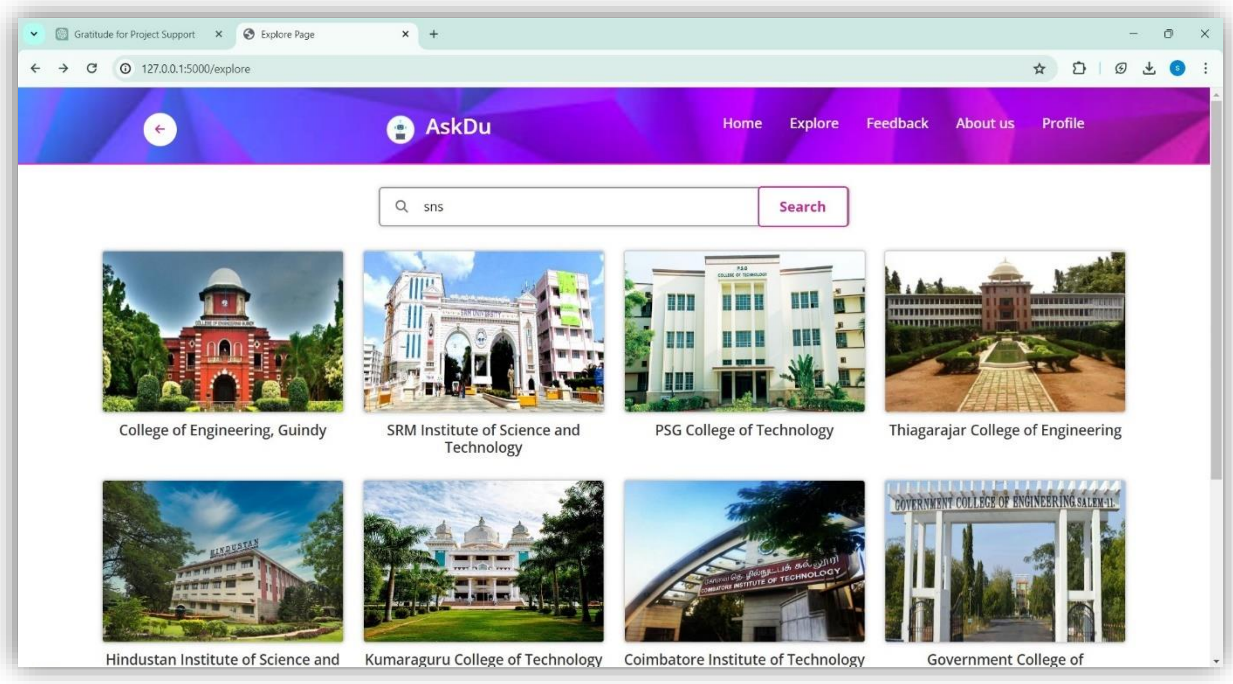


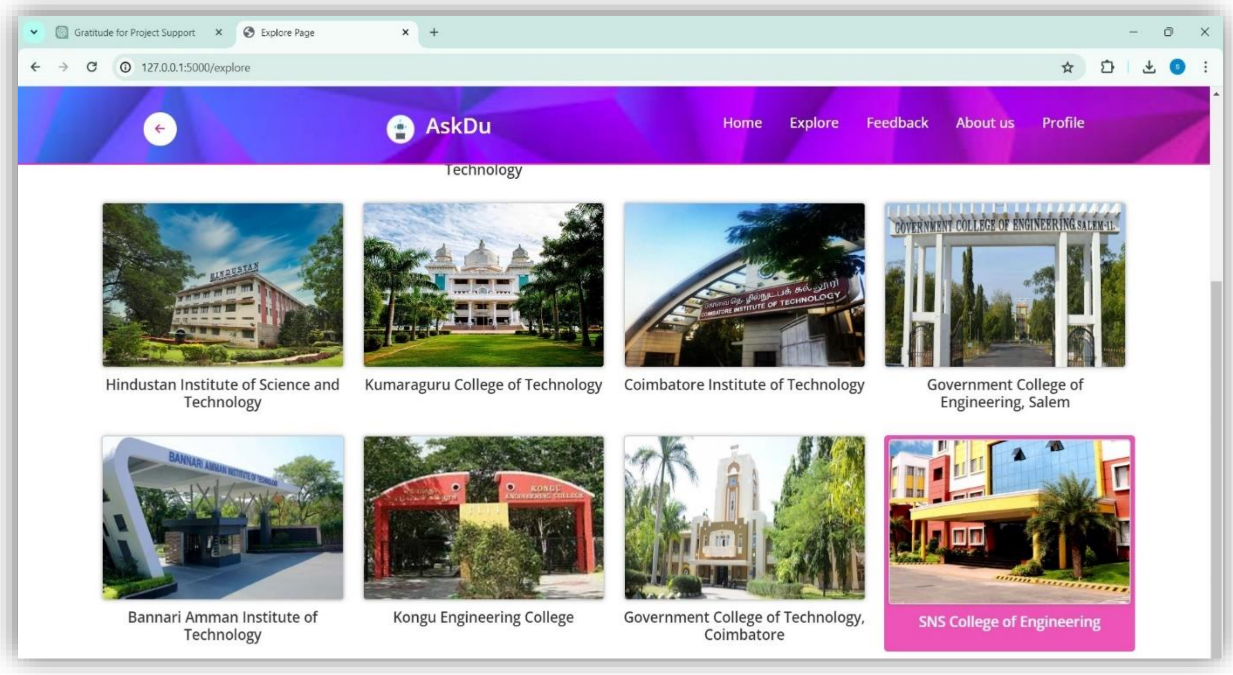
127.0.0.1:5000 says

PDF Saved Successfully

EXPLORE MODULE

- This module enables users to explore information about colleges, courses, and admission criteria.
- It provides features such as browsing college profiles, viewing available courses, checking admission requirements, and accessing additional resources.
- Users can also explore about various competitive exams.


Gratitude for Project Support
Explore Page
127.0.0.1:5000/explore
AskDu
Home
Explore
Feedback
About us
Profile
sns
Search
College of Engineering, Guindy
SRM Institute of Science and Technology
PSG College of Technology
Thiagarajar College of Engineering
Hindustan Institute of Science and
Kumaraguru College of Technology
Coimbatore Institute of Technology
Government College of

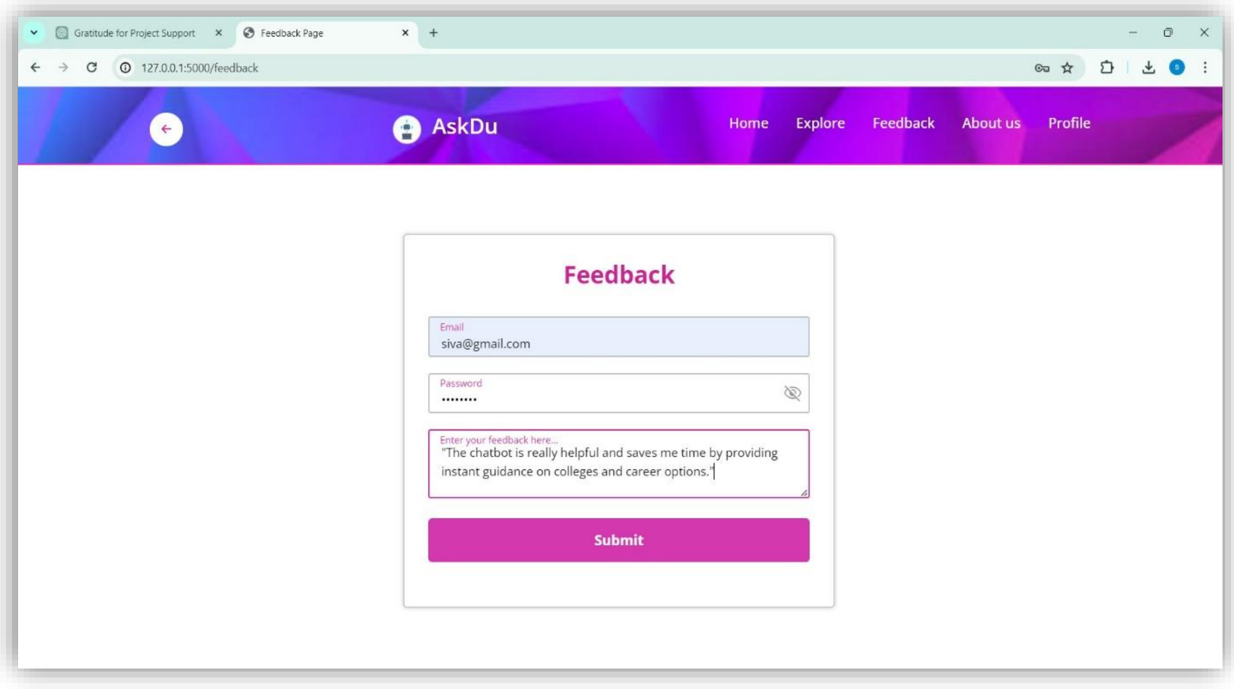

Gratitude for Project Support
Explore Page
127.0.0.1:5000/explore
AskDu
Home
Explore
Feedback
About us
Profile
Technology
Hindustan Institute of Science and Technology
Kumaraguru College of Technology
Coimbatore Institute of Technology
Government College of Engineering, Salem
Bannari Amman Institute of Technology
Kongu Engineering College
Government College of Technology, Coimbatore
SNS College of Engineering

Gratitude for Project Support
Best Engineering College in Coi
snsce.ac.in
SNS College of Engineering
An Autonomous Institution
Counselling Code: 2734
News & Events: • Top Fresher's{terv}Company (Placement Drive on 19.01.24) - 07 Studen
COE EDC IQAC ARIIA AQAR NIRF AICTE Email Validation GRC CAMPUS GMAIL STATUTORY BODIES
HOME ABOUT US SPOTLIGHTS NEWS ROOM DIFFERENTIATORS COURSES BLOGS PLACEMENT LIFE @SNS ADMISSION 2024 CONTACT
Redesigning Common Mind & Business Towards Excellence
Dream High With SNS
Create a Lifestyle YOU have always dreamed of, with our world class facility and state-of-the-art Infrastructure.
WATCH ON YOUTUBE

ADDITIONAL MODULES

Feedback Page

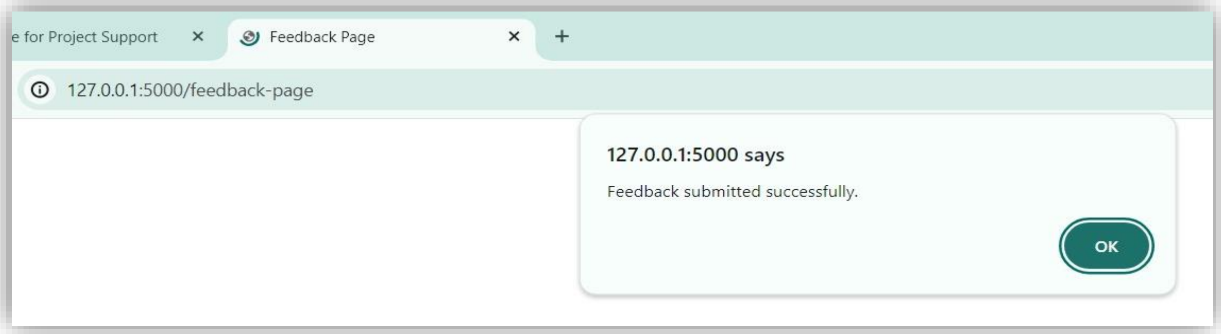
This Module allows users to share their thoughts and suggestions, enabling us to continuously improve our platform to better serve your needs.



siva@gmail.com

••••••••

"The chatbot is really helpful and saves me time by providing instant guidance on colleges and career options."



e for Project Support
Feedback Page
127.0.0.1:5000/feedback-page

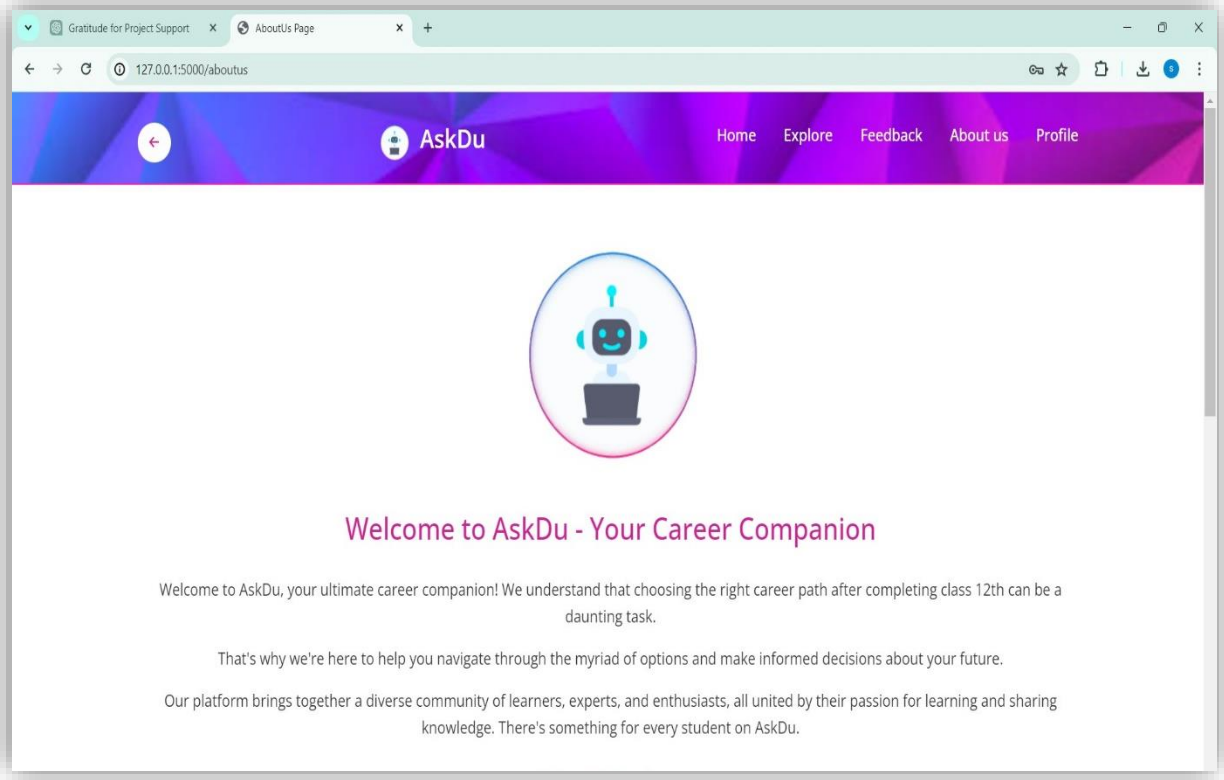
127.0.0.1:5000 says

Feedback submitted successfully.

OK

About us Page

Dive into our About Us Module to discover the vision, mission, and values that drive our platform, and learn about the team behind its creation and can also contact us via mail.



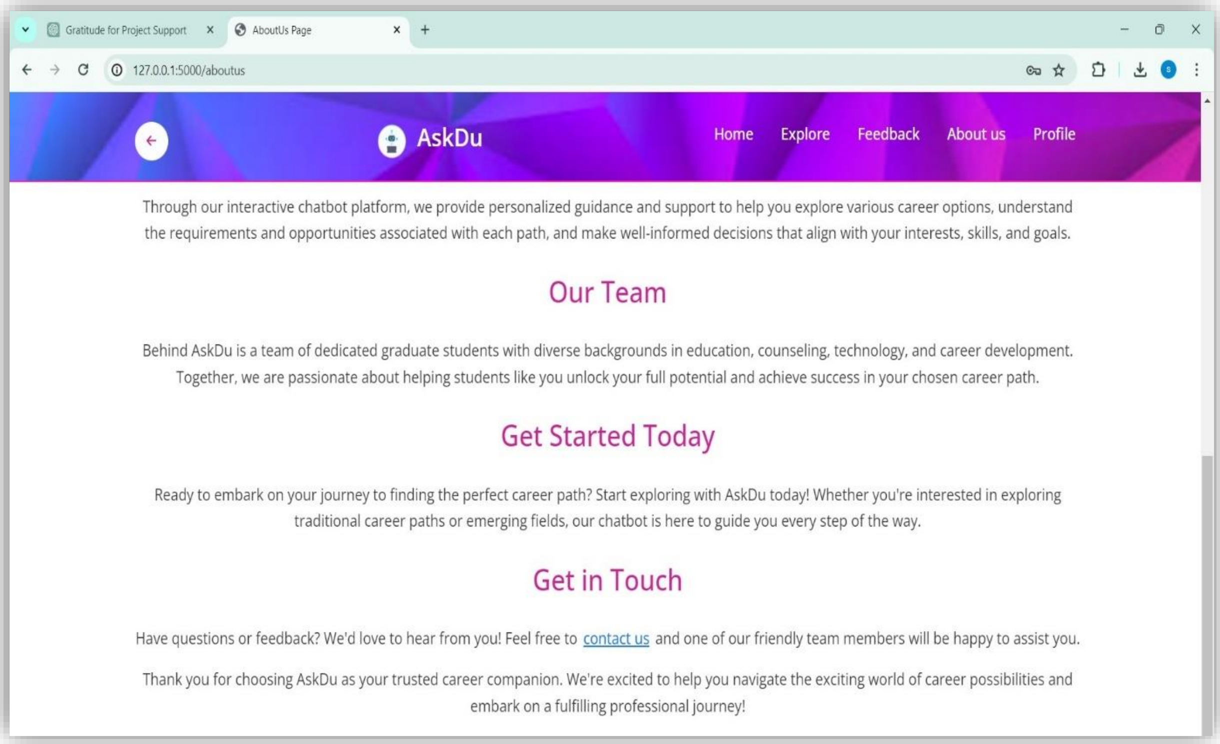
Gratitude for Project Support
AboutUs Page
127.0.0.1:5000/aboutus

Welcome to AskDu - Your Career Companion

Welcome to AskDu, your ultimate career companion! We understand that choosing the right career path after completing class 12th can be a daunting task.

That's why we're here to help you navigate through the myriad of options and make informed decisions about your future.

Our platform brings together a diverse community of learners, experts, and enthusiasts, all united by their passion for learning and sharing knowledge. There's something for every student on AskDu.



Through our interactive chatbot platform, we provide personalized guidance and support to help you explore various career options, understand the requirements and opportunities associated with each path, and make well-informed decisions that align with your interests, skills, and goals.

Our Team

Behind AskDu is a team of dedicated graduate students with diverse backgrounds in education, counseling, technology, and career development. Together, we are passionate about helping students like you unlock your full potential and achieve success in your chosen career path.

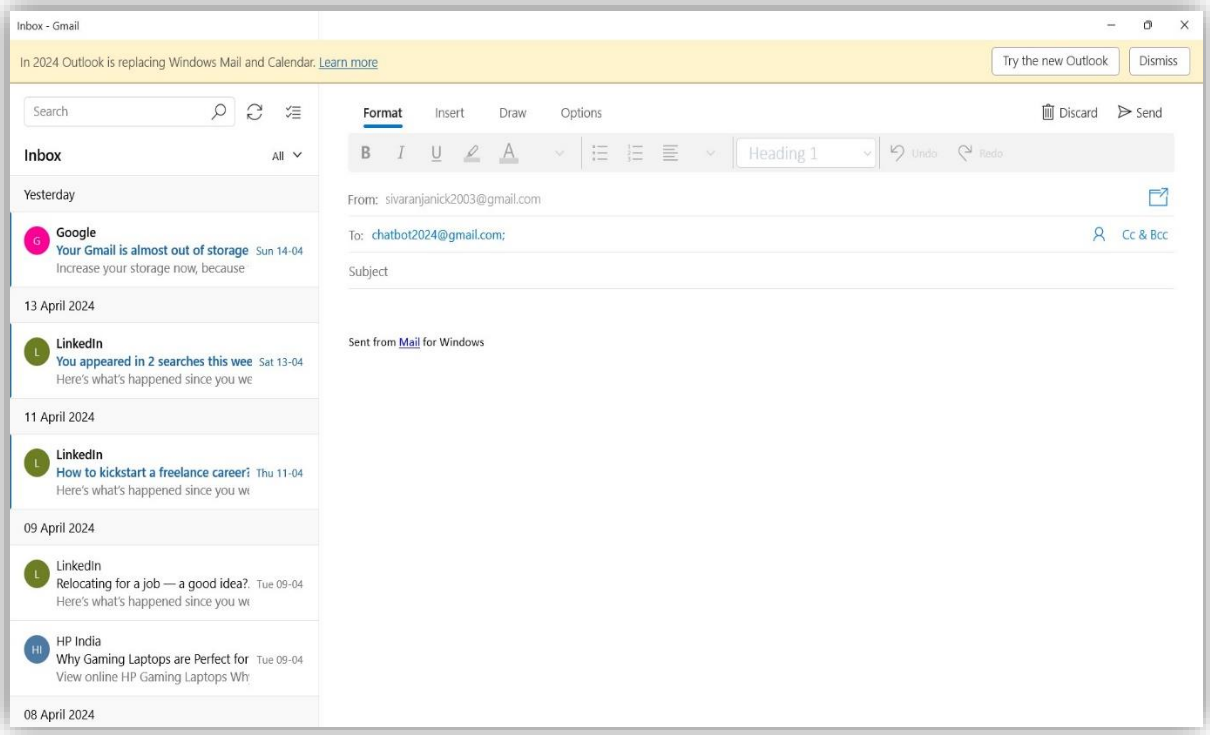
Get Started Today

Ready to embark on your journey to finding the perfect career path? Start exploring with AskDu today! Whether you're interested in exploring traditional career paths or emerging fields, our chatbot is here to guide you every step of the way.

Get in Touch

Have questions or feedback? We'd love to hear from you! Feel free to contact us and one of our friendly team members will be happy to assist you.

Thank you for choosing AskDu as your trusted career companion. We're excited to help you navigate the exciting world of career possibilities and embark on a fulfilling professional journey!



Yesterday
Google
Your Gmail is almost out of storage Sun 14-04
Increase your storage now, because

13 April 2024
LinkedIn
You appeared in 2 searches this wee Sat 13-04
Here's what's happened since you we

11 April 2024
LinkedIn
How to kickstart a freelance career: Thu 11-04
Here's what's happened since you w

09 April 2024
LinkedIn
Relocating for a job — a good idea?. Tue 09-04
Here's what's happened since you w

HP India
Why Gaming Laptops are Perfect for Tue 09-04
View online HP Gaming Laptops Wh

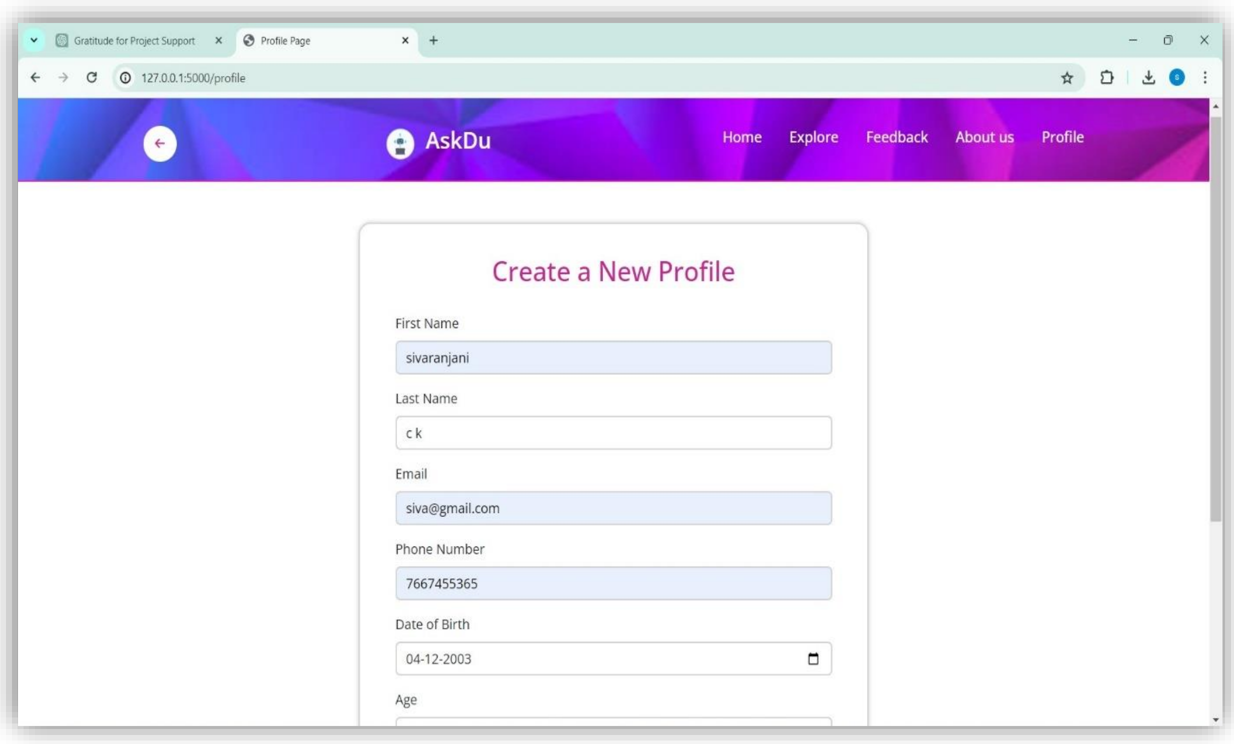
08 April 2024

From: sivaranjanick2003@gmail.com
To: chatbot2024@gmail.com;
Subject

Sent from Mail for Windows

Profile Page

Access your Profile Page to manage your account settings, update your information, and customize your preferences, ensuring a personalized and tailored experience every step of the way.



Create a New Profile

First Name
sivaranjani

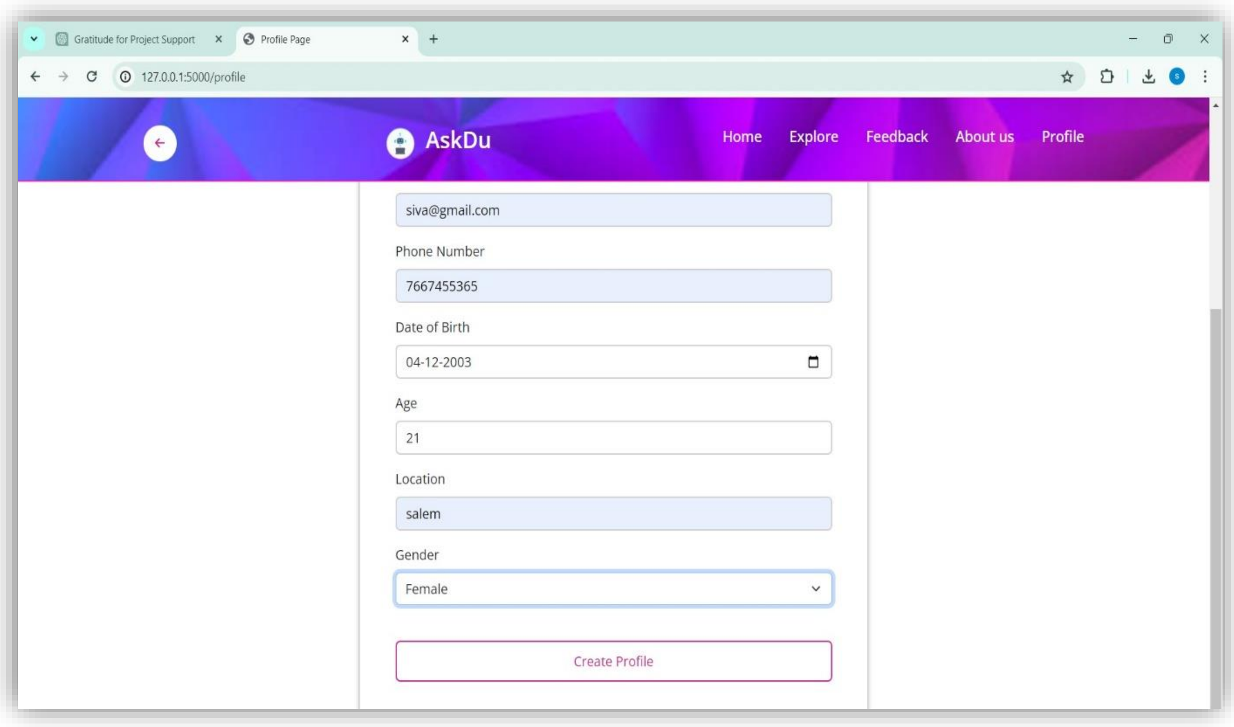
Last Name
c k

Email
siva@gmail.com

Phone Number
7667455365

Date of Birth
04-12-2003

Age



siva@gmail.com

Phone Number
7667455365

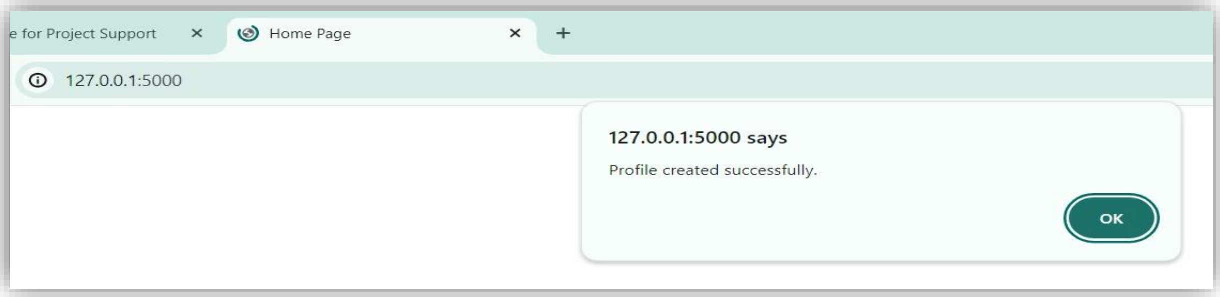
Date of Birth
04-12-2003

Age
21

Location
salem

Gender
Female

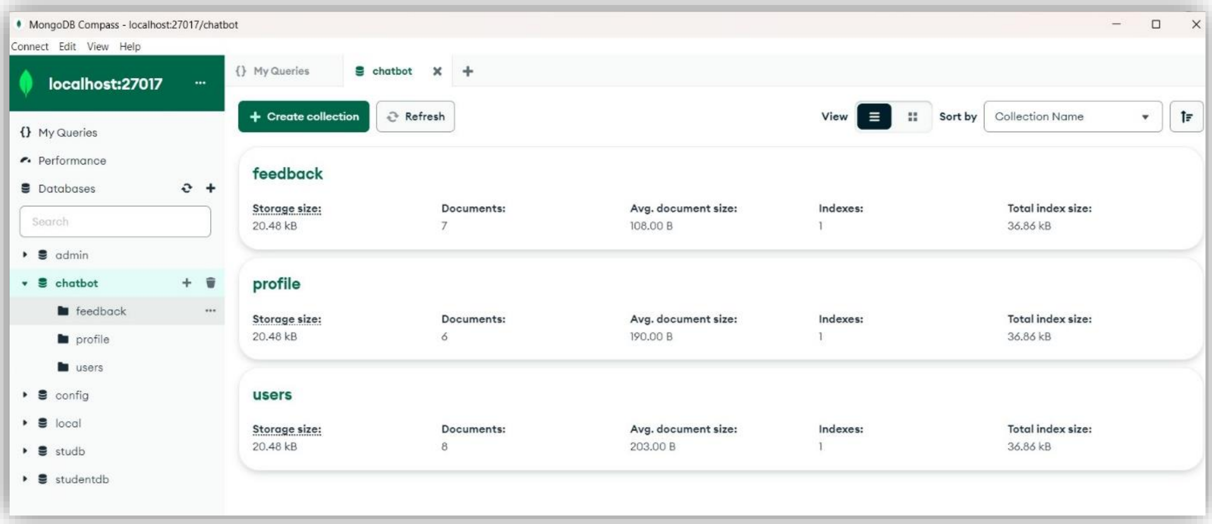
Create Profile



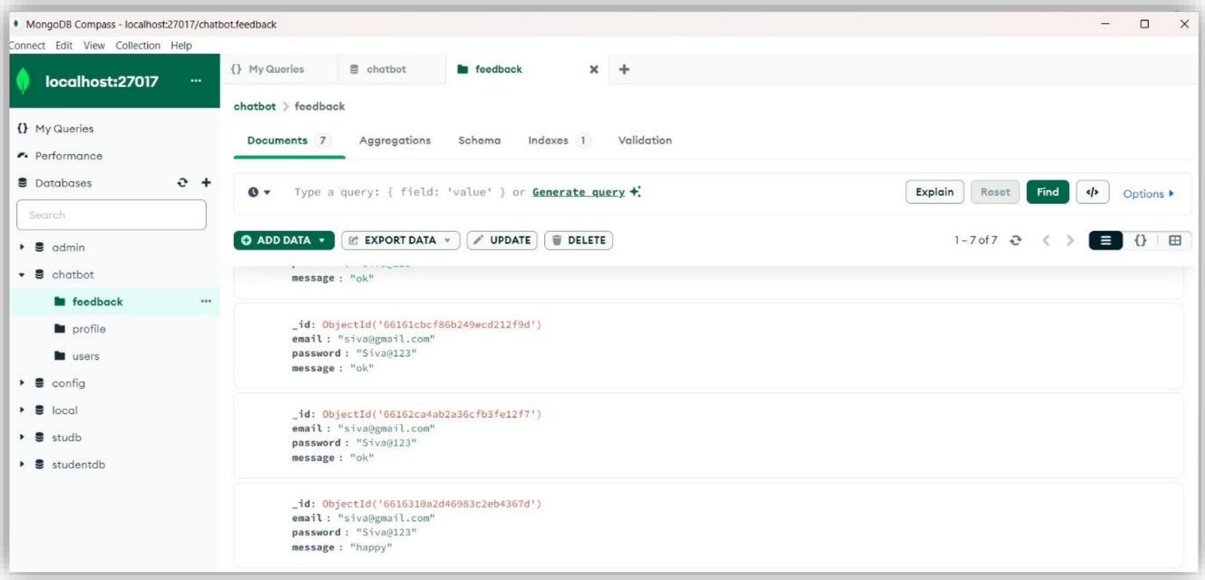
127.0.0.1:5000 says

Profile created successfully.

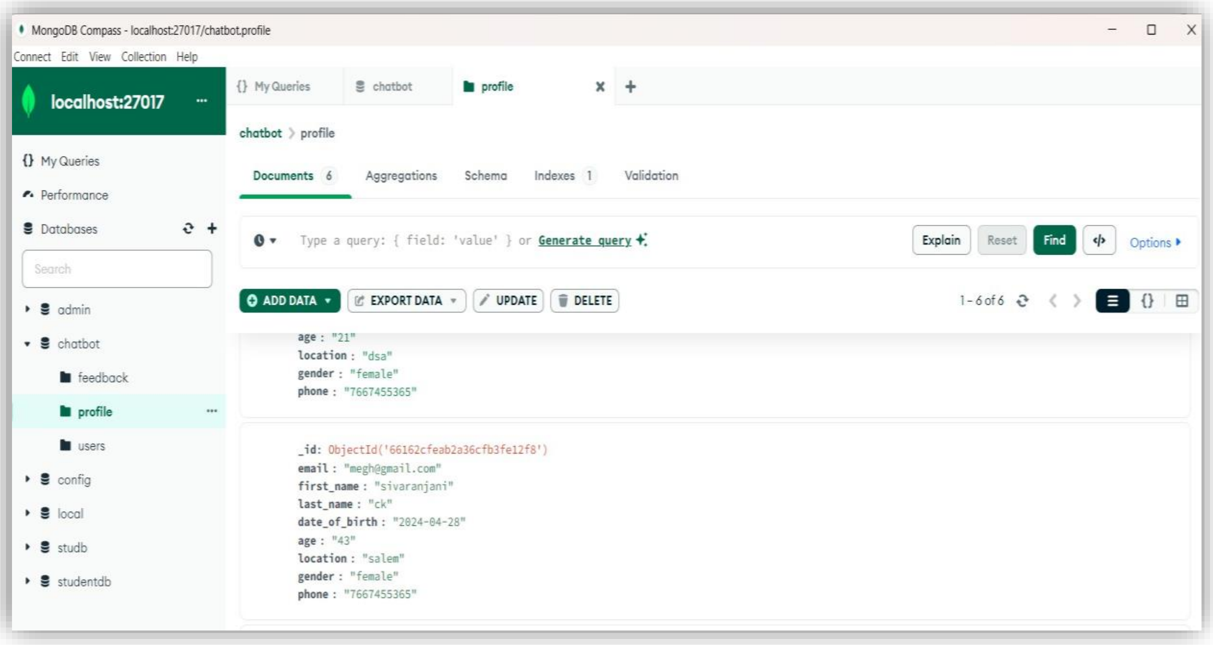
DATABASE



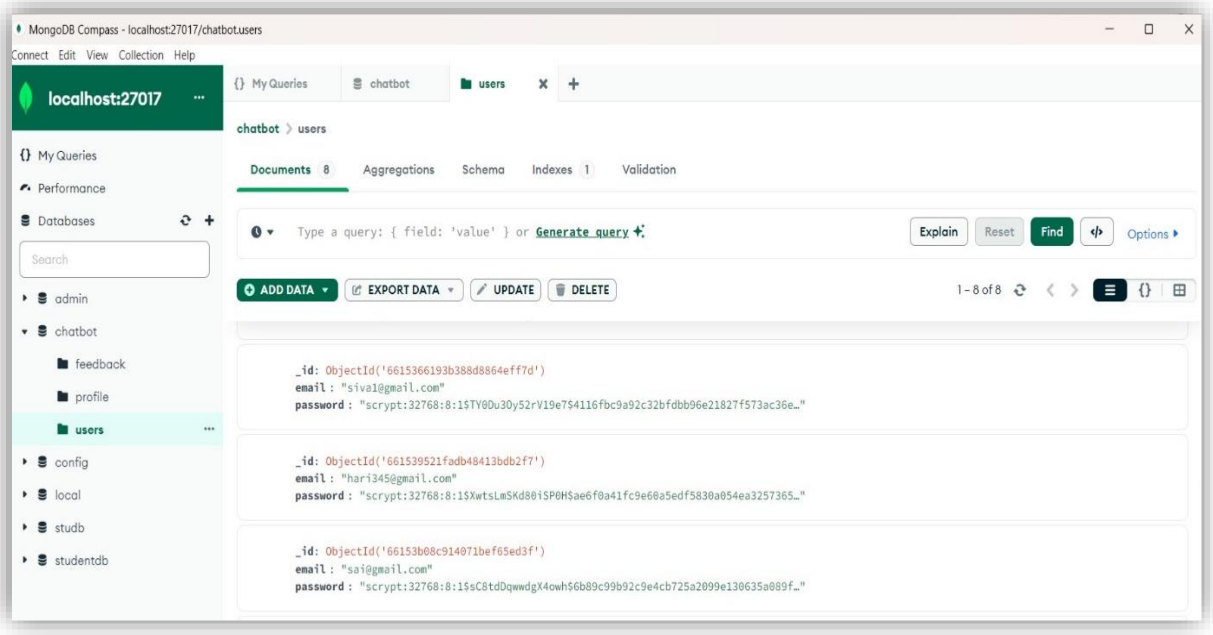
	Storage size:	Documents:	Avg. document size:	Indexes:	Total index size:
feedback	20.48 kB	7	108.00 B	1	36.86 kB
profile	20.48 kB	6	190.00 B	1	36.86 kB
users	20.48 kB	8	203.00 B	1	36.86 kB



```
message : "ok"

_id: ObjectId('66161cbcf86b249ecd212f9d')
email : "siva@gmail.com"
password : "Siva@123"
message : "ok"

_id: ObjectId('66162ca4ab2a36cfb3fe12f7')
email : "siva@gmail.com"
password : "Siva@123"
message : "ok"

_id: ObjectId('6616310a2d46983c2eb4367d')
email : "siva@gmail.com"
password : "Siva@123"
message : "happy"
```



```
age : "21"
location : "dsa"
gender : "female"
phone : "7667455365"

_id: ObjectId('66162cfeab2a36cfb3fe12f8')
email : "megh@gmail.com"
first_name : "sivaranjani"
last_name : "ck"
date_of_birth : "2024-04-28"
age : "43"
location : "salem"
gender : "female"
phone : "7667455365"
```



```
_id: ObjectId('6615366193b388d8864eff7d')
email : "siva1@gmail.com"
password : "scrypt:32768:8:1$TY0Du3Oy52rV19e7$4116fbc9a92c32bfdbb96e21827f573ac36e…"

_id: ObjectId('661539521fadb48413bdb2f7')
email : "hari345@gmail.com"
password : "scrypt:32768:8:1$XwtsLmSKd80iSP0H$ae6f0a41fc9e60a5edf5830a054ea3257365…"

_id: ObjectId('66153b08c914071bef65ed3f')
email : "sai@gmail.com"
password : "scrypt:32768:8:1$sC8tdDqwwdgX4owh$6b89c99b92c9e4cb725a2099e130635a089f…"
```

REFERENCES

1. Mopelola, O. & Benjamin B. (2013) "Career Guidance for Nigerian Students: Why Career Choice Is Becoming More Difficult". Retrieved From, vol. 1, no. 3, pp. 1-16, 2013.
2. Abisoye, O. A., Alabi, I., Ganiyu, S. O., Abisoye, B. O., and Omokore, J. (2015). "A Web-Based Career Guidance Information System for Pre-Tertiary Institutions Students in Nigeria". The International Journal of Scientific Research in Science, Engineering and Technology, vol. 87, no. 2, pp. 80-90, Nigeria 2015.
3. Balogun, V. F., and Thompson, A. F. (2009) "Career Master: A Decision Support System (DSS) for Guidance and Counselling in Nigeria. The Pacific Journal of Science and Technology", vol. 19, no. 6, pp. 126-148, Nigeria 2009.
4. Crystal D'Mello "Online Career Guidance System", In International Journal of Advanced Research in Computer Science and Software. vol 7, no. 8, pp. 83-100, February 2018.
5. JoAnn Harris-Bowlsbey, Ed.D. Kuder Research Faculty "Overview of Career Guidance: Its Foundations, Objectives, and Methodology", White Paper, Kuder.2321– 8169 vol. 1, no. 5, pp. 163-176, December 2017.
6. Bootstrap Documentation

   https://getbootstrap.com/docs/5.0/getting-started/introduction/