

A MAJOR PROJECT REPORT ON

**A Personalized Cut-off Predictor And College Recommender for
Engineering Admissions**

Submitted in fulfillment for the degree of Bachelor of Technology in
Computer Science and Technology

Submitted by

Twinkle Bothara(1912007)

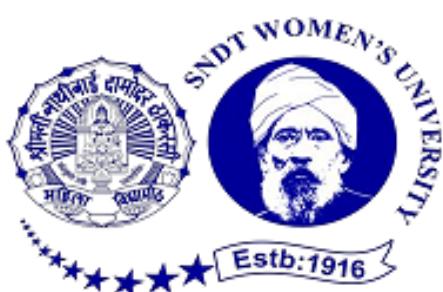
Rashi Chaturvedi(1812006)

Shruti Chavan(1912011)

Under the guidance of

Prof. Narendra Gawai

Prof. Sudhakar Yerne



USHA MITTAL INSTITUTE OF TECHNOLOGY

S.N.D.T. WOMEN'S UNIVERSITY

MUMBAI – 400049

2022 – 2023.

CERTIFICATE

This is to certify that Twinkle Bothara, Rashi Chaturvedi, Shruti Chavan has successfully completed Major project work on A Personalized Cut-off Predictor And College Recommender for Engineering Admissions in the complete fulfillment for the bachelor's degree in Computer Science and Technology during the year 2022-2023 as prescribed by S.N.D.T. Women's University, Mumbai.

GUIDE

Prof. Narendra Gawai

Prof. Sudhakar Yerne

HEAD OF THE DEPARTMENT

Prof. Kumud Wasnik

PRINCIPAL

Dr. Shikha nema

Examinar 1

Examinar 2

DECLARATION

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed

Signature

Twinkle Bothara(1912007)

Rashi Chaturvedi(1812006)

Shruti Chavan(1912011)

Date:

ACKNOWLEDGEMENT

It is indeed a great pleasure and proud opportunity for us to present this major-project report for the fourth-year degree at Usha Mittal Institute of Technology. The success of this project has throughout depended upon a combination of hard work and unending co-operation and guidance provided to us by our project guide. Ultimately no words could describe the deep sense of cooperation and ready nature to help us.

We would like to thank, **Dr. Shikha Nema (Principal)**, **Prof. Kumud Wasnik (H.O.D. of CST Department)**, **Prof. Narendra Gawai**, and **Prof. Sudhakar Yerne (Project Guide)**, who made very valuable guidance and cooperation during our project.

Further, we are thankful to all the teaching and non-teaching staff of the Computer Science and Technology Department for their cooperation during the project work. We are very grateful to those who in the form of books had conveyed guidance in this project work.

Contents

1	INTRODUCTION	6
2	PROBLEM STATEMENT	8
3	LITERATURE SURVEY	10
4	EXISTING SYSTEM	13
5	PROPOSED SYSTEM	15
6	SYSTEM ANALYSIS	17
6.1	Model Phases:	17
6.2	Process:	18
6.3	Flow Diagram:	21
7	SYSTEM REQUIREMENT	25
7.1	TECHNOLOGIES TO BE USED	25
7.2	ALGORITHMS TO BE USED	27
8	HARDWARE AND SOFTWARE REQUIREMENTS:	31
9	OBJECTIVES AND SCOPE	32
10	IMPLEMENTATION	34

List of Figures

1	Proposed System for cut-off predictor and College Recommendation	15
2	Waterfall Model	17
3	Excel Dataset Format	19
4	Working of smart bot to gather user's information	19
5	Format for the final college recommended list	20
6	System Architecture	20
7	DFD 0	21
8	DFD 1 for proposed system	21
9	DFD 2 for user login	22
10	Data Set	34
11	Python code	34
12	Smart bot	39
13	Back-end	43
14	Front-end	46

1 INTRODUCTION

In our digital world, we can see that every facility are available online, one of which is the admission process. Students have to fill out the online CAP round forms after their JEE and MH-CET exams to get admitted into the colleges. So, it is necessary to have subsequent knowledge about colleges. But many times, students make mistakes in their preference list of colleges because of various reasons like inaccurate analysis of colleges, lack of knowledge, and apprehensive prediction. Later, they end up regretting the same after allotment. Our application addresses this issue of the student admission community.

After completion of high school or diploma, students will take admitted to different Engineering colleges. Before taking admission, they are faced with many problems. In Maharashtra, there are around 110 engineering courses wherein throughout every year approximately 90,000 to 1.5 lakh students pass out. In addition, there are only 540 engineering colleges. Students and their parents are concerned about various queries related to the CAP round and filling out a list of colleges to apply to. Students get confused when selecting a good engineering college. Percentile, Rank, reservation, category, preferred branches, preferred district, and preferred colleges are taken as input and the preference list, on thorough analysis of the last four year's cut-off data, is generated.

In this paper, an attempt has been made to develop an automated web application prediction model for a college admission system which can be used to make a wise choice of college before allotment and predict the cutoff for every college . The system will uses the historical data of the different col-

leges, and using this data predict the most probable college for students. It will use machine learning algorithms, data analysis algorithms and techniques. As a machine learning problem, for training data, the system reads an internal database for college cut-off. User can predict the colleges within his/her range rather than waiting for the decision.

2 PROBLEM STATEMENT

The college recommendation system assumes a significant role as college choice requires a great deal of study of the quantity of variables according to the future perspective. Looking for a decent university is a tough job for a high schooler who needs to seek after his/her graduate studies. These students look for different angles like college campus, teaching staffs, extracurricular exercises in college, foundation of a college, and so on, even the surveys of college is looked to get additional affirmation about the subtleties. Our personalized college recommender and cut-off predictor will prescribe great colleges to a graduate depending on his/her decision of field,top evaluations, area and past rate.

In today's world, it is increasingly important for students to be able to find the best colleges for their individual strengths and abilities. In fact, the admission process for many colleges has become so competitive that it comes down to how well they perform in school and on JEE and MH-CET tests. In past students need to make a handwritten or excel list for every MH-CET CAP round and select the colleges and rank them according to their cutoff (which usually changes every year) and also requires the student to read and reread each college's cutoff list. Thus, by using personalized college recommender and cutoff predictor it will be significant as it will diminish the manual work and robotize this with the assistance of software.

Here, we are trying to design a model for developing a web application that serves as an extra and efficient online helpdesk capable of providing a 24/7 service for students in need of guidance to choose their career stream. It will be capable of proving acute services on single search option where students

will get a proper list of campuses in proper rank based format based on their JEE or MHT-CET scores. Our approach will be designed to help less tech-savvy people by offering them a familiar environment and using a conversational agent to ease and guide their interactions. The proposed model will be validated by implementing a chatbot that will provide textual-based communications and uses state-of-the-art technology. This web application will then be tested with the help of students and teachers from the engineering degree sector.

The proposed system will be highly beneficial to students who are looking to pursue engineering education. It will provide them with a data-driven approach to selecting colleges that are a good fit for their profile and preferences, and help them set realistic expectations for admission. Additionally, the system will provide students with a personalized roadmap for their engineering education, making the college selection process less stressful and more manageable.

3 LITERATURE SURVEY

The International Journal for Research in Engineering Application & Management (IJREAM), by Prof. Vishal R. Shinde, Miss. Anagha Bagul, Mr. Amit Gupta, Miss. Sneha Javeri, explained a system presenting college FAQs chatbot that answers the queries of the users that are related to the college. It is developed in Python. This system uses different APIs to get information from the database/corpus by using NLTK (Natural Language Tool Kit).

In IEEE Dec 2016, International Conference on Computational Intelligence and Computing Research (ICCIC), by Naveen Kumar M, Linga Chandar P C, Venkatesh Prasad A, Sumangali K, a system was explained that has two different modules: i) Rule-based system & ii) AI based system. Rule based system is dependent on rules and conditions. Therefore it is not reliable for all the cases. On the other hand AI based system is a trained system based on natural language processing (NLP) with the data sets, which are conversion dialogs, to extract the combination of conversation, including intent, context, an entity.

The system explained in International Conference on Advances in Computing, Communications, and Informatics (ICACCI), by B. R. Ranoliya, N. Raghuwanshi, and S. Singh, breaks down the user sentence into intent and an entity. It uses a heuristic approach to deliver the suitable response. The system also uses pattern matching to classify the text and produce a suitable or best response for the clients.

P. R. Telang, A. K. Kali, M. Vukovic, R. Pandita and M. P. Singh, in the paper IEEE Internet Computing, vol. 22, mentioned about an Android based

system. This system is made for visually weakened people. Therefore it can be easily launched by using Google Voice Search. The system makes use of two types of resources mediawiki API and AB library. Pattern matching technique is also used in this method.

The research paper published in International Journal of Creative Research Thoughts (IJCRT), by Emil Babu, Geethu Wilson, Msc Scholar, Assistant Professor, focuses on automating the process of communication by use of chat-bot. It is an Android based system. It attempts to create a chat-bot using Artificial Intelligence Markup Language and using various algorithms such as Keyword Matching.

Elementary Education Online mentions about a system, by Dr. Arunakumari B.N, Vishnu Sastry H.K, Sheetal Neeraj, Shashidhar R, which is used to deploy the web application worldwide. Heroku is used for deployment. The system is made with python machine learning libraries i.e. panda and numpy. Through this system the candidate obtains a rough idea regarding the seat he or she is likely to get depending on his or her rank and category.

In 2017 International Conference on Advances in Computing, Communication and Control (ICAC3), IEEE Xplore: 19 March 2018, Varsha Powar, Sheetal Girase, Debajyoti Mukhopadhyay, Anuja Jadhav, Shweta Khude, Shital Mandlik mentioned about a system that provides a recommendation system which generates the user interested colleges list. This system stores the college list in the dataset which calculates the accuracy and precision of the system. The students can also give feedback for their performance.

Zhenrong Zhang, in 2021 IEEE International Conference on Industrial Ap-

plication of Artificial Intelligence (IAAI), mentioned about a system that is based on improved collaborative filtering technology. The research includes two aspects: i) Designing the system framework & ii) Personalized recommendation model. This system maintains a more accurate recommendation under any number of nearest neighbors, which proves the effectiveness of the system.

4 EXISTING SYSTEM

To find the cutoff of Engineering colleges students first need to visit the DTE website and search for colleges that fit their criteria based on their reservations. Then students start searching for colleges matching their percentile and rank based on the cutoff of CAP rounds. The list of colleges typically involves students manually researching and comparing different colleges based on various factors such as location, reputation, courses offered, fees, and admission criteria. Students may consult various sources such as college websites, brochures, and third-party websites to gather information about colleges. However, this process can be time-consuming and overwhelming, especially for students who lack guidance or access to reliable information.

After making the list of colleges, the existing system allows users to search for a particular college and shows the results based on the feedback given by alumni or students currently studying in the colleges in the form of ratings. Some colleges may provide cut-off scores for admission to their engineering programs, but this information may not be easily accessible or transparent. As a result, students may have difficulty setting realistic expectations for their chances of admission to different colleges. The information with its college info, courses, fees structure, placements, cutoffs of previous years, infrastructure, and faculty are shown of one particular college. However, some key points like placements with average salary and course fee structure will be available as they are not published every year by colleges. Also, one has to keep on searching for the desired colleges and check for the cutoff of every college and write them down in sequential order. The complete process

takes more than a day to complete and one cannot make any mistakes while preparing the list or their future can be in jeopardy.

There may be some websites or platforms that attempt to provide recommendations and information about engineering colleges based on various criteria. However, these recommendations may not be personalized to the individual student's profile and preferences, and may not provide accurate information about cut-off scores or admission criteria. Some system uses deep learning and natural language processing to analyze the previous search history of the users. The system then recommends similar colleges to students, this system tracks the user's behavior to recommend colleges that align with their search interests.

Overall, the existing system for college selection in engineering admissions may be cumbersome, stressful, and potentially inaccurate. There is a need for a more personalized and data-driven approach to college selection that takes into account the individual student's profile and preferences, as well as the admission criteria and cut-off scores of different engineering colleges.

5 PROPOSED SYSTEM

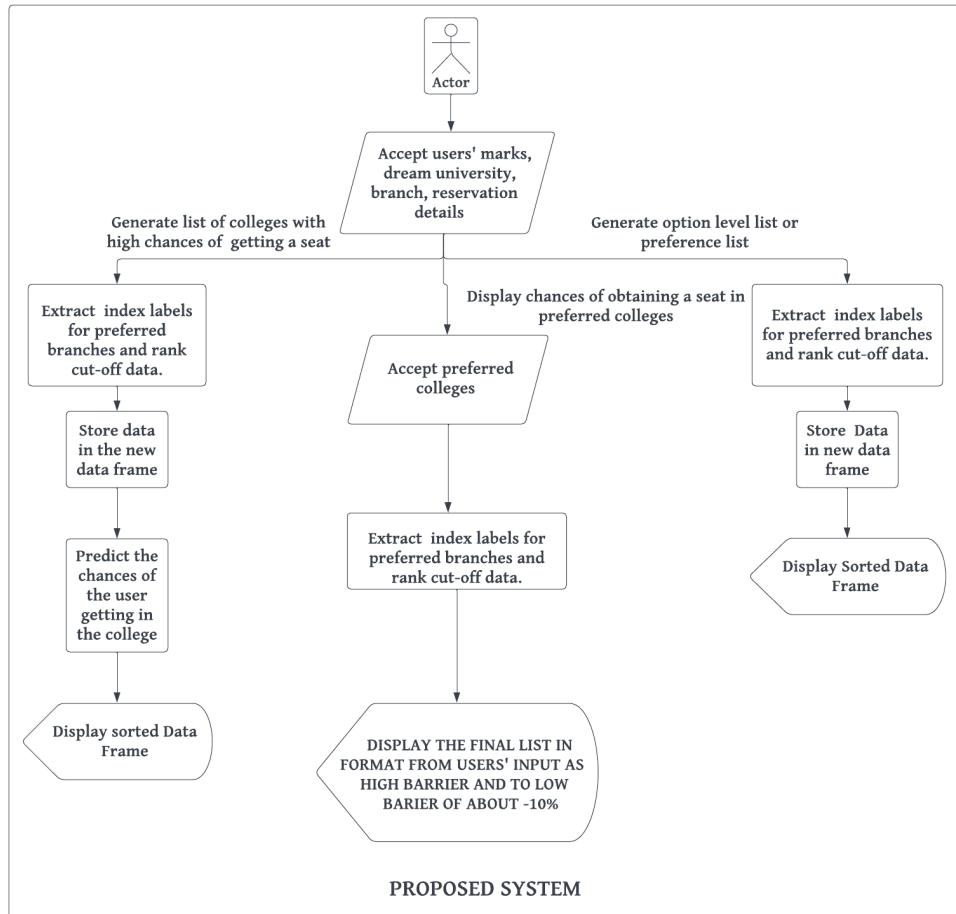


Figure 1: Proposed System for cut-off predictor and College Recommendation

Current system which is similar to our system provides solutions to the student by individually searching for the colleges. But other than a few prominent colleges and the ones in our vicinity, the students don't have much idea about the rest. Also there is no such system available for where the students can get a general idea of the colleges they can get in all over Maharashtra in a single search. Our system aims in guiding a student for his career right through this.

The data for cutoff prediction will be collected from past four years of cutoff based on MH-CET exam cutoff list by DTE website and converted into excel sheet for evaluation. The average cutoff with the range of +/- 1.5% will be

predicted by studying the data collected. The Cutoff will be calculated as per DTE format of three CAP rounds process.

The system will contain a basic text-based messaging system that will collect the tentative information about the student's curriculum marks, his/her reservation, and the branch of engineering he/she desires. The system will help students to get a list of colleges that are fitting their MHT-CET and JEE-Based marks in a proper higher to lower order Our aim is to develop a web application for graduate aspirants in order to help them in their admission process.

This project will help high school students to get the list of colleges based on their JEE and MH-CET scores. The input provided by the students will be compared with past years' college cut-offs and a list would be populated within the range of +5 and -5 of the student's score. The generated list would be ordered based on the government and best-rated college and cut-off. The rating of the college will be collected from various students across different colleges. Along with the cut-off, the resulting list would be filtered based on other criteria like caste information, region, and minority provided by the student. Thus, the populated list can be used by the user to fill the B.E and B.Tech admission form hence reducing their stress of manually preparing the list or roaming everywhere to get information about admissions from their seniors. Also, this project will help the student to get into college based on their career's perspective and also based on the field of interests.

6 SYSTEM ANALYSIS

6.1 Model Phases:

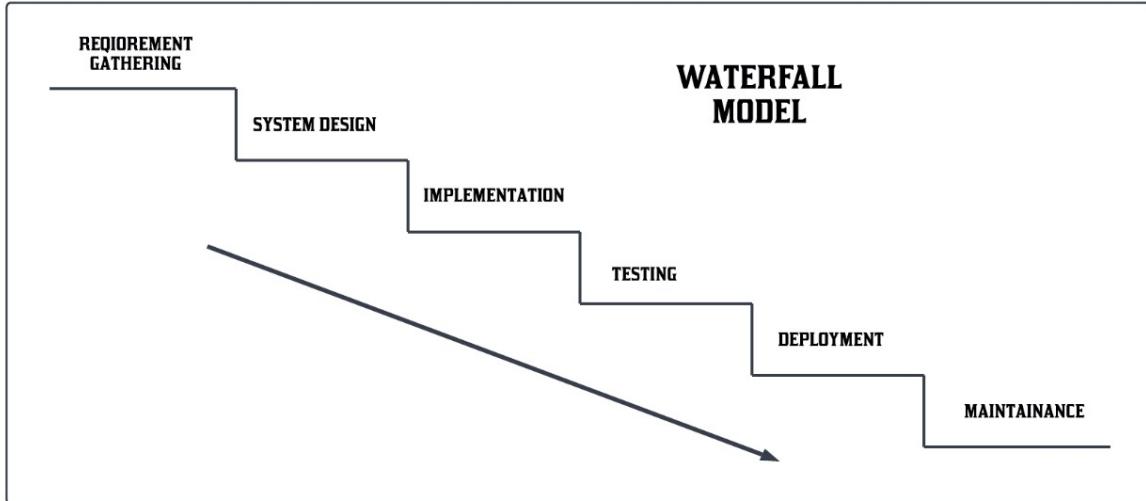


Figure 2: Waterfall Model

The waterfall model is a software development process model that follows a sequential, linear approach to software development. Here are the steps involved in developing the Personalized Cut-off Predictor and College Recommender for Engineering Admissions using the waterfall model:

1. Requirements gathering: Gather all the requirements and specifications for the system. This includes understanding the user's needs, the features and functionalities required, and the constraints of the system.
2. System design: Create a high-level design of the system, including its architecture, data model, and user interface. This involves mapping out the system's components and defining their interactions.
3. Implementation: Write the code for the system according to the design specifications. This involves developing the frontend and backend components of the system.

4. Testing: Test the system to ensure that it meets the specified requirements and functions as expected. This includes unit testing, integration testing, and system testing.
5. Deployment: Deploy the system to a production environment and make it available to end-users. This involves installing the system on a web server and configuring it for use.
6. Maintenance: Maintain the system by fixing bugs, making updates, and adding new features as required. This includes providing technical support to users and ensuring that the system remains up-to-date and secure.

Using the waterfall model for the Personalized Cut-off Predictor and College Recommender for Engineering Admissions system can provide a structured and organized approach to software development. It can also help ensure that the system meets the specified requirements and is delivered on time and within budget.

6.2 Process:

In the first stage of the project we will start preparing the database which will contain the list of all Engineering colleges, college codes, home universities, branch codes, branch names, and reservation categories cut-off with all 3 CAP rounds held that year. The cut-off details will be taken from the official MH-CET Cell website. In addition, the main database will consist of the average of the previous four years' percentile cut-off data. The cut-off database will consist of the ranks with respect to branch, college, and category. All the rest of the information about the colleges that we are going to put in the database will be based on students' reviews. These reviews will be collected by us personally from some of the students of various colleges

and by way of google sheets.

COLLEGE CODE	COLLEGE NAME	HOME UNIVERSITY	BRANCH NAME	BRANCH CODE	CAP - 1			CAP-2			CAP-3		
					OPEN	OBC	etc.	OPEN	OBC	etc.	OPEN	OBC	etc.

Figure 3: Excel Dataset Format

The second stage is to collect the user's MH-CET rank and percentile, JEE rank and percentile, dream universities, branches if they want to opt for multiple branches for the same college, their reservation, their high school percentage and the CAp round to apply for. The information will be collected using text based messaging.

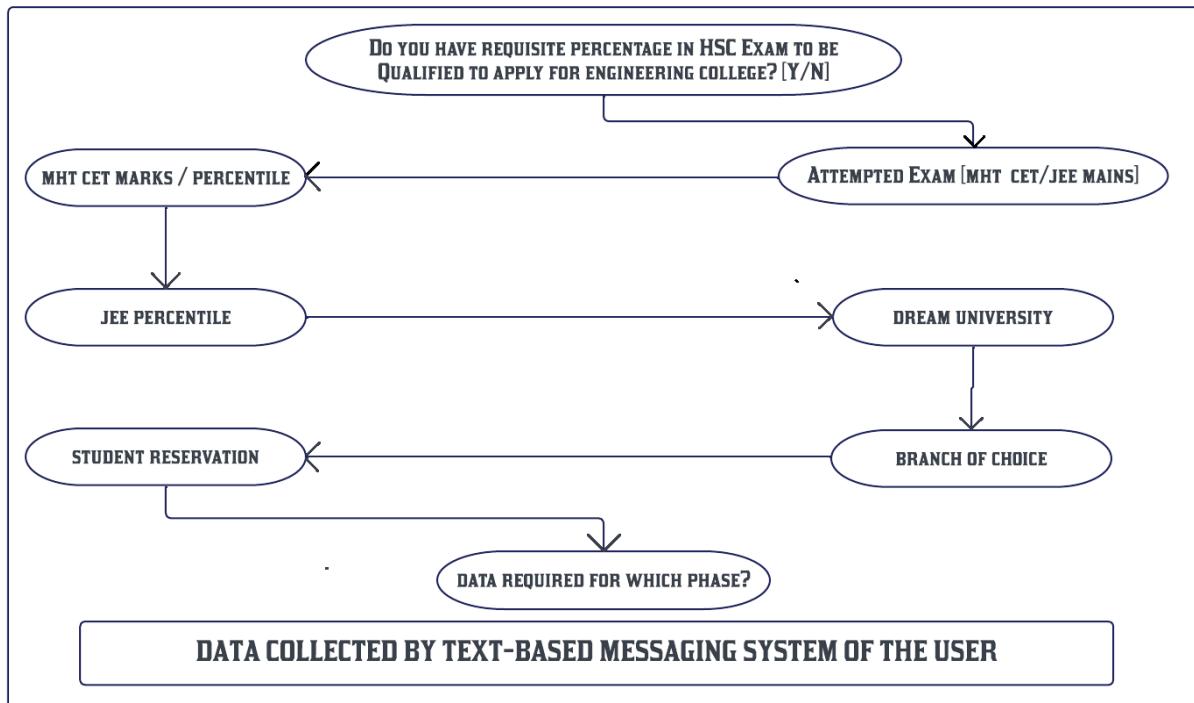


Figure 4: Working of smart bot to gather user's information

Our system will compare students' scores with college's cut-off and generate a list. The list will display the name of college, location, chances, cutoff range, branch, branch intake and official link of particular college. System also generates ratings based on users provided reviews. Our system will gen-

erate ratings using an efficient rating prediction algorithm. Simultaneously we will be working on the front-end and back-end of the system.

College name	Branch	Cut-off range	Chances	Branch Intake	Location	Official Website

Figure 5: Format for the final college recommended list

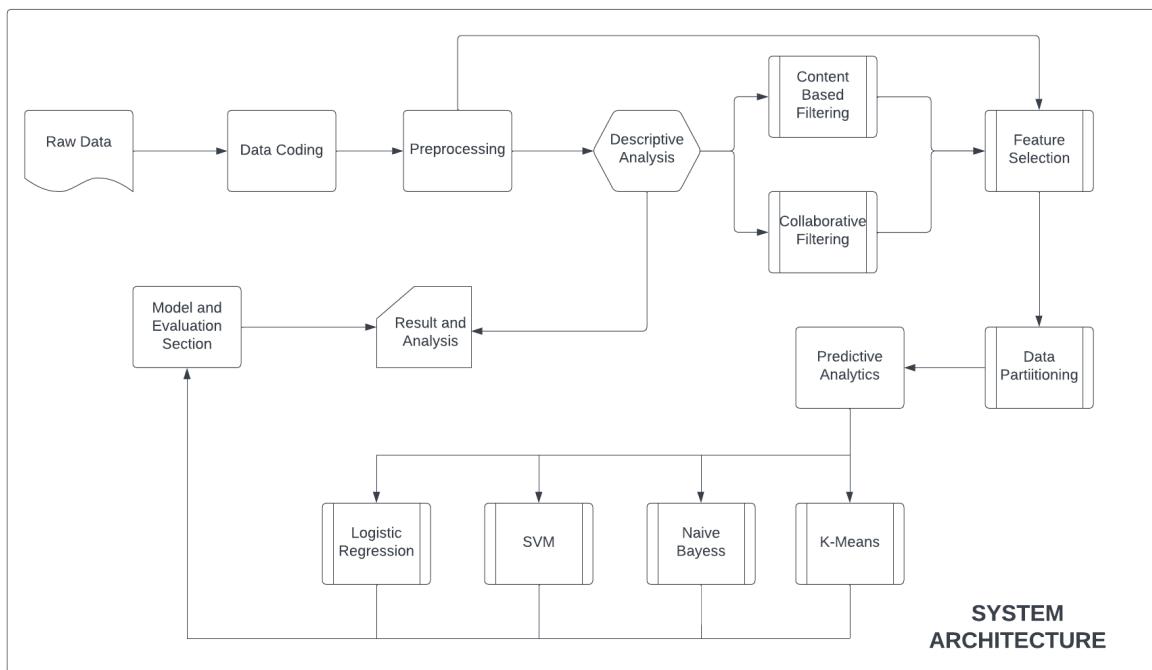


Figure 6: System Architecture

6.3 Flow Diagram:

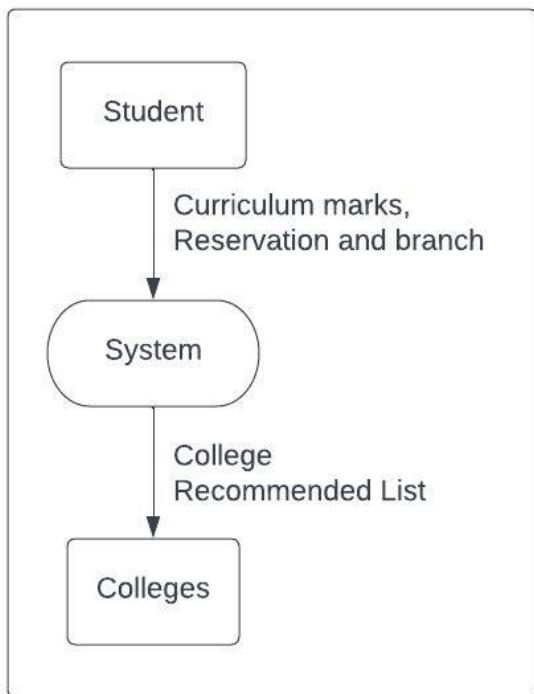


Figure 7: DFD 0



Figure 8: DFD 1 for proposed system

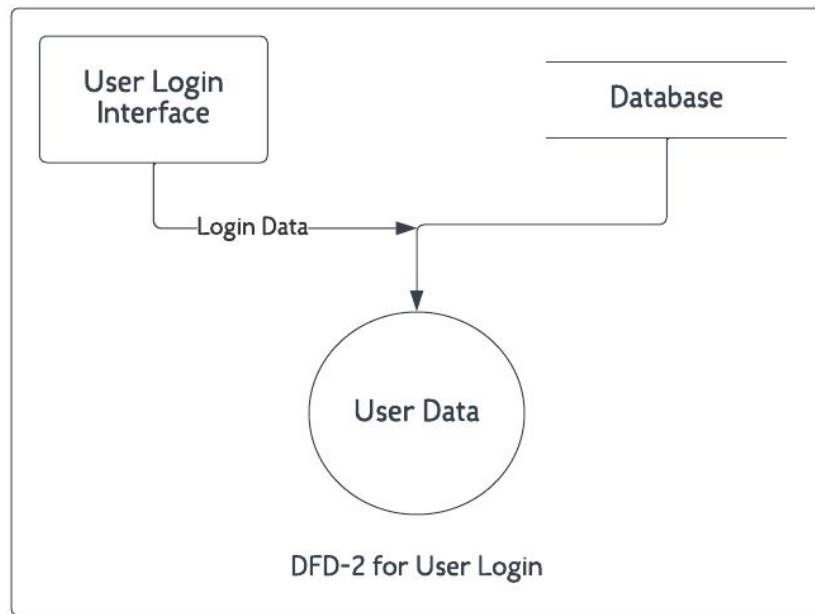
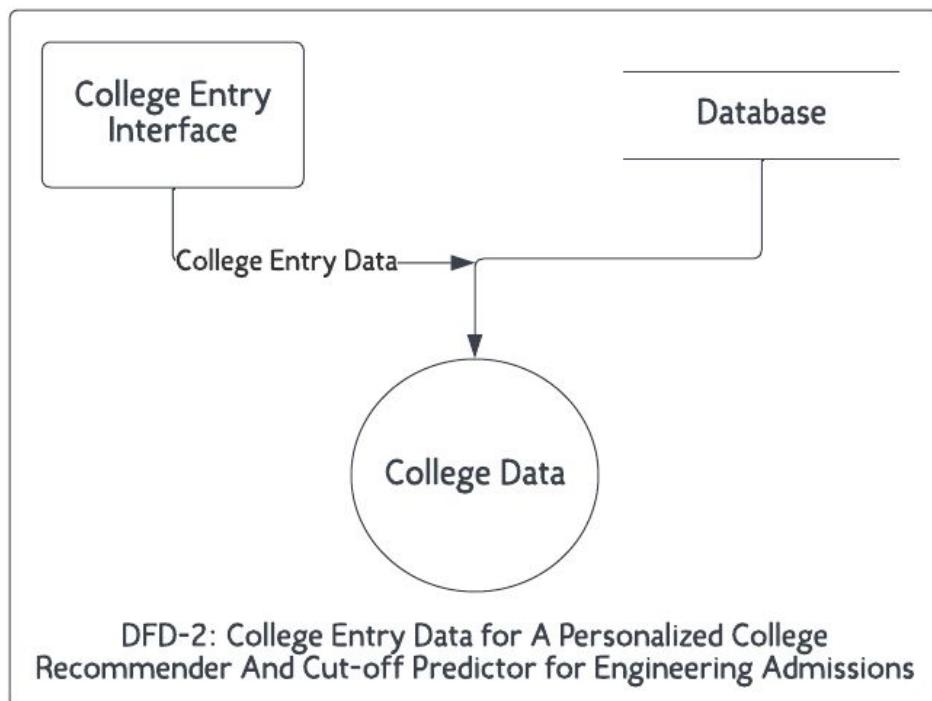
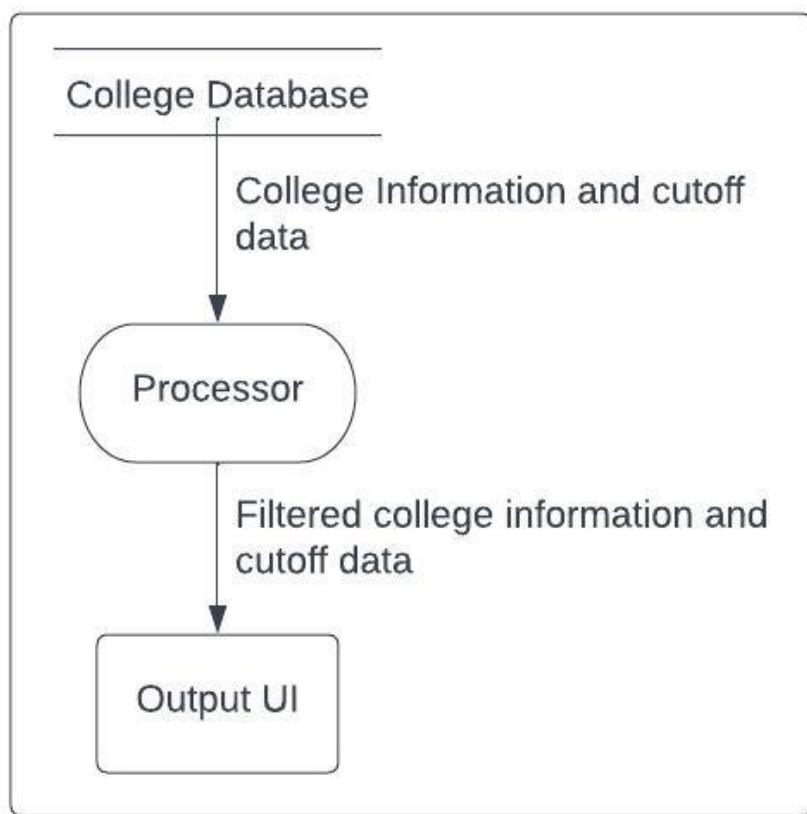
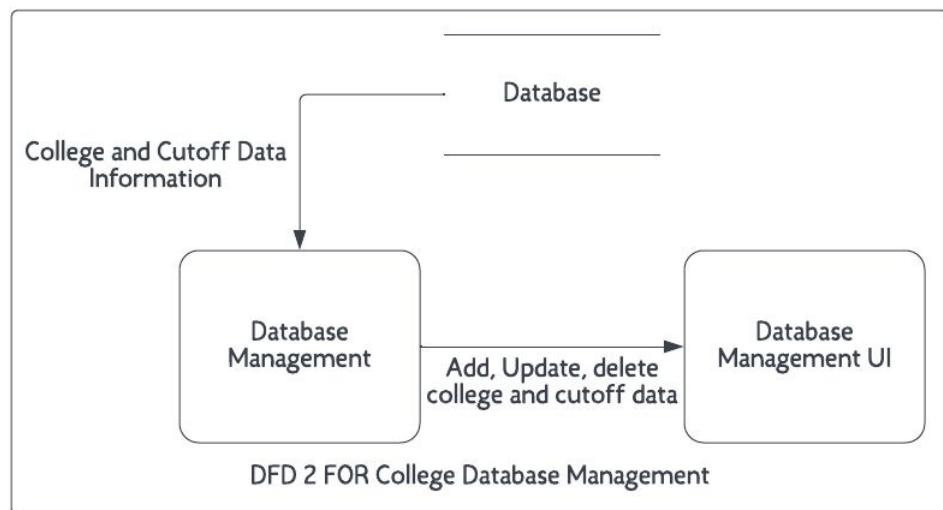
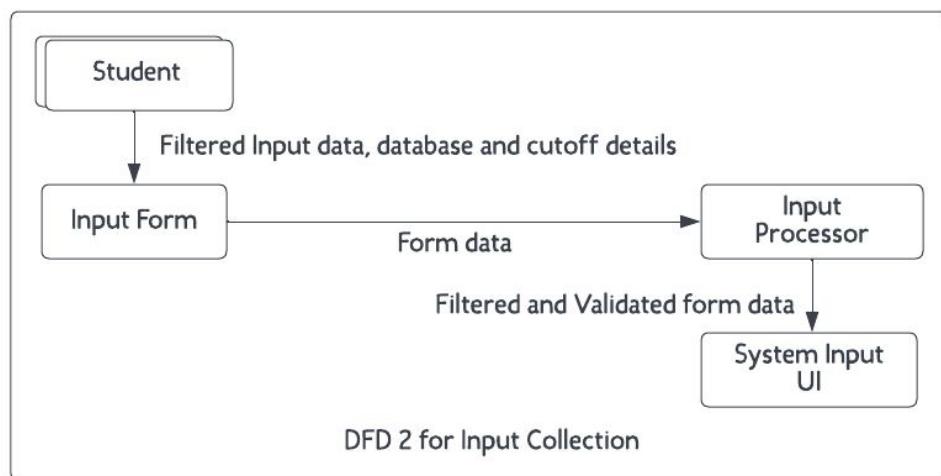
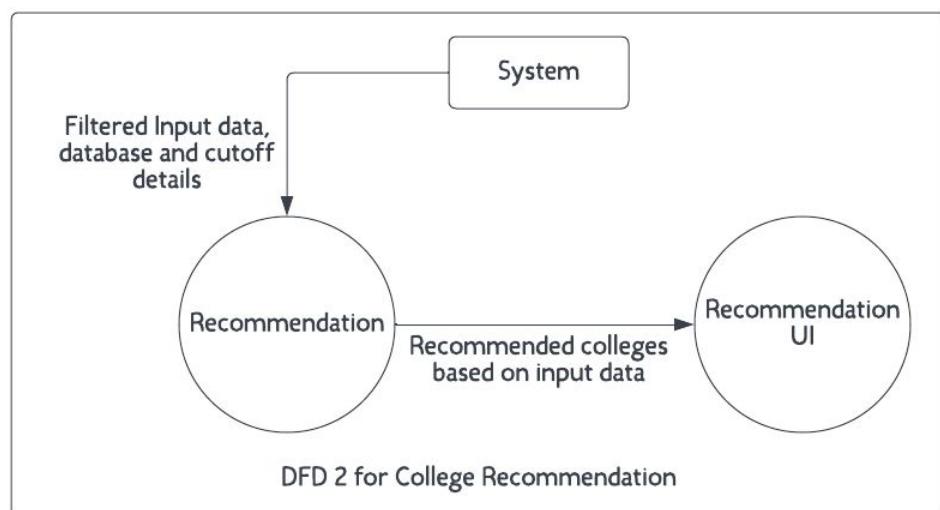


Figure 9: DFD 2 for user login







7 SYSTEM REQUIREMENT

A software requirements specification is a description of a software system to be developed, describing all the functional and non-functional requirements which also include a set of use cases that describe interactions between the users. In this system following are the functional requirements

7.1 TECHNOLOGIES TO BE USED

- Python:

It is a general-purpose programming language designed to be used in a range of applications, including data science, software, automation as well as for web development.

The python code can be embedded into HTML code, or can be combined with various web template systems, and web frameworks. The web server will combine the results of the interpreted and executed Python code, which may be any type of data, including images, with the generated web page. This code may also be executed with a command-line interface (CLI) and can be used to implement standalone graphical applications.

It can also be used for Prediction via Machine Learning as Python libraries allow us to access, process, and transform our data.

- JSON:

JSON (JavaScript Object Notation) is a lightweight data interchange format that is widely used for data storage and exchange. In the context of the Personalized Cut-off Predictor and College Recommender for Engineering Admissions, JSON can be used to store and exchange

data between different components of the system, such as the frontend interface and the backend server.

The proposed system using JSON will involve the following steps:

1. Data collection and storage: Data will be collected from various sources, such as college websites, admission criteria, and student profiles. This data will be stored in JSON format, using a structured schema that allows for easy retrieval and manipulation of data.
2. API development: An application programming interface (API) will be developed to allow for easy retrieval and exchange of data between the frontend interface and the backend server. The API will use JSON as the data format, allowing for easy parsing and manipulation of data.
3. Frontend interface development: A user-friendly frontend interface will be developed using HTML, CSS, and JavaScript. The frontend interface will interact with the backend server through the API, allowing users to input their academic records and other relevant information, and receive personalized recommendations for engineering colleges.
4. College recommendation: Based on the input provided by the user and the machine learning models stored in JSON format, the backend server will generate personalized recommendations for engineering colleges, which will be sent back to the frontend interface in JSON format.

Overall, the proposed system using JSON will provide a flexible and scalable approach to college selection in engineering admissions. By using a structured data format that is easily retrievable and manipulable, the system will enable seamless integration between different components, and facilitate the development of user-friendly interfaces and APIs.

- HTML and CSS::

The most basic form of web development includes designing page with the help of the most prominent and consistent web technology which not only contributes in designing the front end i.e. the web page in this case, but, also helps in organizing it with different forms of styles using CSS (cascading style sheets). HTML also allows developers to work along with other database applications that are necessary to store the details of the students.

- Java Script: JavaScript can be used to add interactivity to the frontend interface, including the validation of user input, the handling of user events, and the communication with the backend server through APIs. Developing a smart bot using JavaScript for the Personalized Cut-off Predictor and College Recommender for Engineering Admissions system involves determining the bot's purpose, choosing a platform that supports JavaScript, designing the conversation flow, training the bot, implementing the bot using JavaScript and APIs, testing the bot, and integrating the bot with the system.

7.2 ALGORITHMS TO BE USED

- Cut-Off Predictor Using Logistic Regression Algorithm Model:

Logistic Regression is a binary classification algorithm that predicts the probability of an outcome being either 1 or 0 (or Yes or No) based on a set of input features. The cutoff predictor using logistic regression is a machine learning algorithm that predicts whether an outcome is positive (1) or negative (0) based on a set of input features and a given cutoff value.

It models the relationship between the input features and the probability of the outcome being positive using the logistic function (also called the sigmoid function). The logistic function takes a linear combination of the input features and applies a sigmoid transformation to map the output to a value between 0 and 1. This value represents the probability of the outcome being positive given the input features.

To train a logistic regression model, we use a set of labeled data (also called the training data) where we know the outcome for each set of input features. The model adjusts the coefficients of the linear combination of the input features to maximize the likelihood of predicting the correct outcome for each set of input features in the training data. The optimization is typically done using a method called maximum likelihood estimation.

Once the model is trained, we can use it to predict the probability of the outcome being positive for new sets of input features. To convert the predicted probability to a binary outcome, we apply a cutoff value. If the predicted probability is greater than the cutoff value, we predict a positive outcome (1), and if it is less than or equal to the cutoff value, we predict a negative outcome (0).

The choice of cutoff value can have a significant impact on the performance of the model. A higher cutoff value leads to a lower false positive rate (i.e., predicting a positive outcome when the true outcome is negative), but a higher false negative rate (i.e., predicting a negative outcome when the true outcome is positive). A lower cutoff value leads to a higher false positive rate but a lower false negative rate.

In summary, the cutoff predictor logistic regression algorithm uses logistic regression to model the relationship between the input features and the probability of the outcome being positive and applies a cutoff value to convert the predicted probability to a binary outcome.

- College Recommender Using Logistic Regression Algorithm Model:

The college recommender algorithm predicts the likelihood of a student being admitted to different colleges based on their input features (such as percentile, university, and caste). The algorithm then recommends colleges based on the probability of admission.

The algorithm uses a machine learning model (such as logistic regression) trained on a dataset of historical admissions data to predict the probability of admission for a given student. The model is trained on features such as percentile, university, and caste, and other relevant factors that are known to affect admission decisions.

Once the model is trained, the algorithm applies it to the input features of the current student to predict their probability of admission to different colleges. Based on the predicted probability, the algorithm recommends colleges that are most likely to admit the student. The recommended colleges may vary depending on the cutoff probability used and the number of colleges that the algorithm is trained on.

The algorithm may also incorporate additional factors such as the student's preferences, location, and academic interests to further refine the list of recommended colleges. For example, the algorithm may recommend colleges that are located in the same state as the student or colleges with strong programs in the student's area of interest.

Overall, the college recommender algorithm aims to help students make informed decisions about where to apply to college by using machine learning to predict their likelihood of admission and recommend colleges accordingly.

- College Recommender List Generator:

To make a list of recommended colleges based on predicted cutoff using logistic regression, we first train a logistic regression model on a dataset of historical admissions data. The model is trained to predict the probability of admission based on input features such as GPA, test scores, and extracurricular activities.

Once the model is trained, we use it to predict the probability of admission for a new student based on their input features. We can then set a cutoff value for the predicted probability of admission, which determines whether a student is likely to be admitted or not.

Based on the predicted probability and the chosen cutoff value, we can make a list of recommended colleges for the student. For example, we can recommend colleges where the predicted probability of admission is above the cutoff value.

The list of recommended colleges can be generated using a simple loop that iterates through the predicted probabilities and adds the corresponding college to the list if the probability is above the cutoff. The final list of recommended colleges can then be returned to the user.

Overall, making a list of recommended colleges based on predicted cutoff using logistic regression involves training a model to predict the probability of admission, setting a cutoff value, and generating a list

of recommended colleges based on the predicted probabilities and the cutoff.

8 HARDWARE AND SOFTWARE REQUIREMENTS:

1. Hardware Interface Requirements

- Processor speed of 0.5 Ghz or more for mobile gadgets
- Processor speed of 1.5ghz or more for desktop and computer gadgets
- Ram of 500mb and above for all devices
- Free storage memory capacity of more than 100mb

2. Software Interface Requirements

- Windows/ android/ Linux/ mac/ chrome or any other operating system
- Mozilla Firefox / Google chrome / opera mini / UC browser or internet explorer

3. Development tools

- Ms-Excel/ Google sheets
- MySQL database
- Entity framework
- GitHub
- Google Colab

4. Communication Interface Requirements

- Internet connectivity

9 OBJECTIVES AND SCOPE

The objectives of this system are:

- To help students to fill in their preferences at the time of the option-entry process accurately and to help make better choices of college before allotment.
- Whatever may be the scores, this application will help to find the best colleges for their desired branches. Hence, our proposed computer aided system will help the students to get the list of all colleges in which they could get the admission at the click of a button.
- The list will include colleges in which student has at least 50% of chance getting placed.
- The system will be designed with the use of different machine learning algorithms to come up with a good recommendation and predictor system. To provide a more personalized and interactive experience for users of the Personalized Cut-off Predictor and College Recommender for Engineering Admissions system.
- To improve user engagement and satisfaction by providing a conversational interface for users to interact with the system.
- To reduce the burden on human staff by automating routine tasks such as answering frequently asked questions.
- To increase the accuracy of the system by using natural language processing (NLP) to understand user input and generate appropriate responses.

The scope of this system are:

- This system will be recommending colleges that will predict the chances

of admission to the students to pursue their higher studies so that they will get a clear idea of where they can progress or which factor like MHT-CET or JEE is imp. And this system will be useful for High School students.

- The systems available do not consider a lot of parameters, so this system can easily help students for getting appropriate guidance and also check where they can stand.
- The idea if successfully implemented, then can be used to predict cutoff and recommend colleges for various other fields of education.

The Future scope of this system is:

- Enhance the NLP capabilities of the bot to better understand user input and provide more accurate responses.
- Add more features to the bot, such as the ability to provide real-time updates on college admissions and events.
- Expand the bot's integration with other platforms, such as social media and messaging apps, to reach a wider audience.
- Use machine learning techniques to improve the bot's performance over time by analyzing user interactions and feedback.
- Develop a voice-enabled version of the bot to provide users with a more natural and intuitive interface.

10 IMPLEMENTATION

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	c_code	h_u	cn	B_N	B_C	OPEN	SC	ST	VJ	NT1	NT2	NT3	OBC	OPEN_OH
2	1002	Sant Gadge Baba Ar Government College Civil Engineering		100219110	80.62704755	69.97183145	47.1582093	76.180108	77.0666919	63.4083064	72.5765063	75.58145155		
3	1002	Sant Gadge Baba Ar Government College Computer Science a		100224210	96.9102154	94.87715005	79.785196	91.1419891	95.8448803	93.64662915	94.1471246	96.3612956		
4	1002	Sant Gadge Baba Ar Government College Information Technol		100224610	95.36346875	92.3722212	79.3248888	91.1038771	91.614983	92.5294414	88.9823699	95.1168025		
5	1002	Sant Gadge Baba Ar Government College Electrical Engineers		100229310	85.65706305	82.2850355	61.2289793	75.4178045	85.4043578	82.56708505	78.1473584	84.7480263		
6	1002	Sant Gadge Baba Ar Government College Electronics and Tele		100237210	91.9691783	87.7136738	44.6079615	79.99497345	89.6209172	87.2640994	85.8374792	91.9691038		
7	1002	Sant Gadge Baba Ar Government College Instrumentation Engi		100246810	73.3888902	61.92369805	16.5730741	52.07723512	53.0375377	54.1664697	57.04777	71.80974225		
8	1002	Sant Gadge Baba Ar Government College Mechanical Enginee		100261210	79.728043	70.49344775	26.291459	72.8974176	61.00709175	72.6171856	62.3799839	77.3852017		
9	1005	Sant Gadge Baba Ar Sant Gadge Baba Ar Food Technology		100550310	77.3709066	42.0206383	39.0927708	62.0823842	30.5619889	57.03753	57.04777	70.85422105	76.99	
10	1005	Sant Gadge Baba Ar Sant Gadge Baba Ar Oil and Paints Tech		100551710	52.2301039	46.463102	39.0927708	52.07723512	53.0375377	57.03753	57.04777	45.9187639	21.50	
11	1005	Sant Gadge Baba Ar Sant Gadge Baba Ar Petro Chemical Eng		100552710	30.510627	45.463102	39.0927708	52.07723512	53.0375377	57.03753	57.04777	52.067967	73.24	
12	1012	Dr. Babasaheb Ambe Government Engene Civil Engineering		101219110	57.9144939	38.18699065	12.9353233	44.986801	48.2193247	46.4456321	57.04777	47.019656	51.21	
13	1012	Dr. Babasaheb Ambe Government Engene Computer Engineeri		101224510	87.97321335	90.22366865	61.7323366	86.402198	83.8902317	84.1949276	82.3877908	86.84905475	92.01	
14	1012	Dr. Babasaheb Ambe Government Engene Electrical Engineeri		101229310	88.88478	42.1425094	35.3945703	60.086804	45.9187639	62.3799839	56.3628567	64.4635095	76.106	
15	1012	Dr. Babasaheb Ambe Government Engene Electronics and Tele		101237210	77.7746454	65.89129965	36.7166438	60.7344231	70.9370842	71.2367422	59.4192464	75.1668255	84.390	
16	1012	Dr. Babasaheb Ambe Government Engene Mechanical Engineer		101261210	46.29447635	24.8571126	4.0235666	56.3628587	38.6417086	35.6893743	21.2317049	37.3270793	64.141	
17	1101	Sant Gadge Baba Ar Shri Sant Gajanan N Computer Science a		110124210	93.36998235	86.9107544	55.9208135	90.7746978	87.0728364	92.2448203	79.5368669	92.5749935	93.42	
18	1101	Sant Gadge Baba Ar Shri Sant Gajanan N Information Technol		110124610	91.06444155	83.00264745	45.87439605	84.2234647	87.2542051	89.2369593	57.04777	90.7006121	90.50	
19	1101	Sant Gadge Baba Ar Shri Sant Gajanan N Electrical Engg/Elec		110135610	74.13393675	57.7305878	39.3128682	52.07723512	44.9304271	69.1689566	54.1786057	68.3373619	71.91	
20	1101	Sant Gadge Baba Ar Shri Sant Gajanan N Electronics and Tele		110137210	85.0962893	68.5147352	4.2387378	81.410415	80.5501725	81.1494135	72.9031949	82.14260055	84	
21	1101	Sant Gadge Baba Ar Shri Sant Gajanan N Mechanical Engin		110161210	80.8058065	23.121138	39.0927708	51.9313711	44.9304271	55.8902921	23.9542468	53.4740279	53.361	
22	1105	Sant Gadge Baba Ar Prof. Ram Meghe In Civil Engineering		110519110	40.8850239	10.1476239	39.0927708	17.3037313	28.0116452	25.63665605	22.2500723	27.1781041	10.15	
23	1105	Sant Gadge Baba Ar Prof. Ram Meghe In Computer Science a		110524210	88.42593195	82.09522895	46.5400405	83.88958335	83.24029865	84.0554048	81.0304563	87.20014725	85.05	
24	1105	Sant Gadge Baba Ar Prof. Ram Meghe In Information Technol		110524610	84.71038345	80.61221	10.0599652	78.1473584	83.1461738	77.5734204	77.0311313	82.65658745	79.010	
25	1105	Sant Gadge Baba Ar Prof. Ram Meghe In Artificial Intelligence		110526310	75.428039	62.81562225	39.0927708	57.2112851	74.364085	62.0823842	14.7866072	73.87737	69.150	
26	1105	Sant Gadge Baba Ar Prof. Ram Meghe In Industrial IoT		110526510	50.7285732	23.9925869	39.0927708	20.8205322	53.0375377	57.03753	57.04777	41.3006846	20.82	
27	1105	Sant Gadge Baba Ar Prof. Ram Meghe In Electronics and Tele		110537210	70.6719101	57.01063495	39.0927708	54.31023505	68.2985381	59.90535395	11.2093545	68.16351635	69.71	
28								39.00537203	53.7005443	8.40533045	18.74050703	53.0375377	53.0375377	

Figure 10: Data Set

```
[ ] import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, confusion_matrix

[ ] df = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/dataset/fyedata1.csv',encoding='windows-1254')

df
```

	c_code	h_u	cn	B_N	B_C	OPEN	SC	ST	VJ	NT1	...	VJ_OH	OBC_OH	NT1_OH
0	1002.0	Sant Gadge Baba Amravati University	Government College of Engineering, Amravati	Civil Engineering	100219110.0	80.627048	69.971831	47.158209	76.180109	77.066692	...	49.90823	49.293596	53.492
1	1002.0	Sant Gadge Baba Amravati University	Government College of Engineering, Amravati	Computer Science and Engineering	100224210.0	96.910215	94.877150	79.785140	91.141989	95.844880	...	49.90823	49.293596	53.492

Figure 11: Python code

```
[ ] df1=pd.DataFrame(df,columns= ['c_code','B_C','OPEN','SC','ST','VJ','NT1','NT2','NT3','OBC','OPEN_OH','SC_OH','ST_OH','VJ_OH','OBC_OH','NT2_OH'])
df1
```

	c_code	B_C	OPEN	SC	ST	VJ	NT1	NT2	NT3	OBC	...	ST_OH	VJ_OH	OBC_OH	NT2_OH
0	1002.0	100219110.0	80.627048	69.971831	47.158209	76.180109	77.066692	63.408306	72.576506	75.581452	...	39.696712	49.90823	49.293596	50.845
1	1002.0	100224210.0	96.910215	94.877150	79.785140	91.141989	95.844880	93.646629	94.147125	96.361296	...	39.696712	49.90823	49.293596	50.845
2	1002.0	100224610.0	95.363469	92.372221	79.324888	91.103877	91.614983	92.5294414	88.9823699	95.1168025	...	39.696712	49.90823	49.293596	50.845
3	1002.0	100229310.0	85.657063	82.285904	61.228979	75.4178045	85.4043578	82.56708505	78.1473584	84.7480263	...	39.696712	49.90823	49.293596	50.845
4	1002.0	100237210.0	91.969178	87.713673	44.607596	79.994973	89.020917	87.264099	85.8374792	91.969104	...	39.696712	49.90823	49.293596	50.845
...	
1153	6649.0	664961210.0	10.771849	46.463102	39.092771	52.077235	53.037538	57.037530	57.047770	52.067967	...	39.696712	49.90823	49.293596	50.845
1154	6732.0	673219110.0	31.364660	46.463102	39.092771	52.077235	53.037538	57.037530	57.047770	52.067967	...	39.696712	49.90823	49.293596	50.845
1155	6732.0	673224510.0	86.873430	74.752348	35.698374	65.785425	72.550832	80.388682	87.027500	84.723732	...	39.696712	49.90823	49.293596	50.845
1156	6732.0	673237210.0	NaN	46.463102	39.092771	52.077235	53.037538	57.037530	57.047770	52.067967	...	39.696712	49.90823	49.293596	50.845
1157	6732.0	673261210.0	NaN	46.463102	39.092771	52.077235	53.037538	57.037530	57.047770	52.067967	...	39.696712	49.90823	49.293596	50.845

```
[ ] df1.isnull().values.any()
True

[ ] df1.isnull().sum().sum()
190

[ ] df1.dropna(inplace=True)
print(df1)

      c_code      B_C      OPEN      SC      ST      VJ \
0    1002.0  100219110.0  80.627048  69.971831  47.158209  76.180109
1    1002.0  100224210.0  96.910215  94.877150  79.785140  91.141989
2    1002.0  100224610.0  95.363469  92.372221  79.324889  91.103877
3    1002.0  100229310.0  85.687063  82.285904  61.226979  75.417805
4    1002.0  100237210.0  91.969178  87.713674  44.607596  79.994973
...
1151   6649.0  664929310.0   5.026062  46.463102  39.092771  52.077235
1152   6649.0  664937210.0   65.645289  18.840167  39.092771  52.077235
1153   6649.0  664961210.0   10.771849  46.463102  39.092771  52.077235
1154   6732.0  673219110.0   31.364660  46.463102  39.092771  52.077235
1155   6732.0  673224510.0   86.873430  74.752348  35.698374  65.785425
```

```
[ ] train_data, test_data = train_test_split(df1, test_size=0.2)
test_labels = train_test_split(df1.drop('B_C', axis=1), df1['B_C'], test_size=0.2)

[ ] train_data

      c_code      B_C      OPEN      SC      ST      VJ      NT1      NT2      NT3      OBC ...
705   4139.0  413961210.0  43.013843  46.463102  39.092771  52.077235  53.037538  57.037530  57.047770  52.067967 ...
656   4123.0  413319110.0   7.997339  46.463102  39.092771  52.077235  53.037538  57.037530  57.047770  52.067967 ...
77    1127.0  112724510.0  64.604603  59.705222  22.640509  59.682905  58.800608  51.107204  19.499854  58.857805 ...
32    1107.0  110799510.0  73.336538  68.125071  39.092771  59.682905  53.730279  65.240059  57.047770  71.751413 ...
739   4147.0  414761210.0  31.377133  13.325511   7.445900  52.077235  35.147588  57.037530  57.047770  14.391677 ...
...
1118   6632.0  663224510.0  69.679720  43.266998  39.092771  20.176312  66.446544  64.060928  59.419246  59.339320 ...
386   3193.0  319324610.0  81.814588  75.406292  39.092771  39.419647  59.682905  70.213583  57.047770  77.469568 ...
1042   5399.0  539929310.0  12.928728  46.463102  39.092771  52.077235  53.037538  57.037530  57.047770  52.067967 ...
307   3154.0  315419110.0  19.373650  46.463102  39.092771  52.077235  53.037538  57.037530  57.047770  52.067967 ...
```

```
[ ] # Train the model
model = LogisticRegression()

[ ] score=pd.DataFrame(df1,columns= ['OPEN','SC','ST','VJ','NT1','NT2','NT3','OBC','OPEN_OH','SC_OH','ST_OH','VJ_OH','OBC_OH','NT2_OH','NT3_OH','PWD','DEF','TFWS','EWS','MI'])
score
```

	OPEN	SC	ST	VJ	NT1	NT2	NT3	OBC	OPEN_OH	SC_OH	ST_OH
0	80.627048	69.971831	47.158209	76.180109	77.066692	63.408306	72.576506	75.581452	58.560730	46.897037	39.6967
1	96.910215	94.877150	79.785140	91.141989	95.844880	93.646629	94.147125	96.361296	58.560730	46.897037	39.6967
2	95.363469	92.372221	79.324889	91.103877	91.614983	92.529441	88.982370	95.116803	58.560730	46.897037	39.6967
3	85.687063	82.285904	61.226979	75.417805	85.404358	82.567085	78.147358	84.748026	58.560730	46.897037	39.6967
4	91.969178	87.713674	44.607596	79.994973	89.020917	87.264099	85.837479	91.969104	58.560730	46.897037	39.6967
...
1151	5.026062	46.463102	39.092771	52.077235	53.037538	57.037530	57.047770	52.067967	53.740118	38.641709	39.6967
1152	65.645289	18.840167	39.092771	52.077235	43.305113	51.260095	56.765059	61.839352	73.341193	48.222423	39.6967
1153	10.771849	46.463102	39.092771	52.077235	53.037538	57.037530	57.047770	52.067967	22.250072	1.595488	39.6967

```
[ ] # Function to generate list of recommended colleges
def get_recommended_colleges(c_code, B_C, OPEN, SC, ST, VJ, NT1, NT2, NT3, OBC, PWD, DEF, TFWS, EWS, MI):
    user_data = pd.score

[ ] # Function to generate list of recommended colleges
def get_recommended_colleges(df1):
    user_data_path = pd.score
    user_data = get_df(user_data_path)
    user_data.head()
    predicted_prob = model.predict_proba(user_data)[:, 1]

    for i in range(df.shape[0]):
        if df.loc[i].OPEN>=95:
            print (i,df.loc[i].OPEN)

1 96.910215
2 95.36346875
116 97.006547
117 95.4133248
122 95.08491245
250 96.47195435
```

```

▶ from IPython.utils.text import string
# Accept user input for College data and mhtcet score
score = int(input("MHT CET Score: "))

□ MHT CET Score: 95

[ ] if recommended_colleges:
    print("Based on your input, we recommend applying to the following colleges:")
    print("\n".join(recommended_colleges))
else:
    print("Based on your input, we do not recommend applying to any colleges at this time.")

Based on your input, we do not recommend applying to any colleges at this time.

```

```

# Filter the college data to only include colleges that are part of Mumbai University, have an open reservation, and offer Computer Engineering
r_c = df[(df['h_u'] == 'Mumbai University') & (df['B_N'] == 'Computer Engineering')]

# Filter the recommended colleges to only include those with a cutoff within 5% of the student's score of 95%
r_c = r_c[(r_c['OPEN'] >= 93) & (r_c['OPEN'] <= 97)]

# Sort the recommended colleges by intake in descending order
r_c = r_c.sort_values(by='OPEN', ascending=False)

# Display the recommended college list with the desired columns
r_c[['h_u', 'c_code', 'cn', 'B_C', 'B_N', 'OPEN']]

```

[2]

	h_u	c_code	cn	B_C	B_N	OPEN
289	Mumbai University	3139.0	Vidyalankar Institute of Technology,Wadala, M...	313924510.0	Computer Engineering	96.668239
345	Mumbai University	3184.0	Fr. Conceicao Rodrigues College of Engineering...	318424510.0	Computer Engineering	96.657040
405	Mumbai University	3197.0	Agnel Charities' FR. C. Rodrigues Institute o...	319724510.0	Computer Engineering	96.516323
326	Mumbai University	3176.0	Thakur College of Engineering and Technology, ...	317624510.0	Computer Engineering	96.118317
453	Mumbai University	3208.0	Don Bosco Institute of Technology, Mumbai	320824510.0	Computer Engineering	95.836975
442	Mumbai University	3204.0	St. Francis Institute of Technology,Borivali,...	320424510.0	Computer Engineering	95.296693
464	Mumbai University	3211.0	S.I.E.S. Graduate School of Technology, Nerul,...	321124510.0	Computer Engineering	95.222522
474	Mumbai University	3214.0	Xavier Institute Of Engineering C/O Xavier Tec...	321424510.0	Computer Engineering	94.518918
284	Mumbai University	3135.0	Manjara Charitable Trust's Rajiv Gandhi Instit...	313524510.0	Computer Engineering	94.380211
266	Mumbai University	3189.0	Pednekar Education Foundation's Datta Dhankar...	318924510.0	Computer Engineering	93.777554

Jupyter Server: Local

```

r_c = df[(df['h_u'] == 'Savitribai Phule Pune University') & (df['B_N'] == 'Information Technology')]
r_c = r_c[(r_c['OPEN'] >= 70) & (r_c['OPEN'] <= 80)]
r_c = r_c.sort_values(by='OPEN', ascending=False)
r_c[['h_u', 'c_code', 'cn', 'B_C', 'B_N', 'OPEN']]

```

[103]

	h_u	c_code	cn	B_C	B_N	OPEN
1125	Savitribai Phule Pune University	6634.0	KJEI's Trinity Academy of Engineering, Yewale...	663424610.0	Information Technology	79.194732
922	Savitribai Phule Pune University	5161.0	Dr. Vithalrao Vikhe Patil College of Engineer...	516124610.0	Information Technology	78.912409
1009	Savitribai Phule Pune University	5330.0	Pune Vidyarthi Griha's College Of Engineering...	533024610.0	Information Technology	77.924510
881	Savitribai Phule Pune University	5125.0	Pravara Rural Education Society's Sir Visvesv...	512524610.0	Information Technology	72.472683
974	Savitribai Phule Pune University	5177.0	Matoshri College of Engineering and Research ...	517724610.0	Information Technology	71.424304

[98]	r_c = df[((df['h_u'] == 'Sant Gadge Baba Amravati University') (df['h_u'] == 'Dr. Babasaheb Ambedkar Technological University,Lonere')) & r_c = r_c[(r_c['OPEN'] >= 35) & (r_c['OPEN'] <=40)] r_c = r_c.sort_values(by='OPEN', ascending=False) r_c[['h_u', 'c_code', 'cn', 'B_C', 'B_N', 'OPEN']]																																											
...	<table border="1"> <thead> <tr> <th></th><th>h_u</th><th>c_code</th><th></th><th>cn</th><th>B_C</th><th>B_N</th><th>OPEN</th></tr> </thead> <tbody> <tr> <td>1092</td><td>Dr. Babasaheb Ambedkar Technological Universit...</td><td>6545.0</td><td>Samarth Education Trust's Arvind Gavali Colleg...</td><td>654519110.0</td><td>Civil Engineering</td><td>38.173880</td></tr> <tr> <td>76</td><td>Sant Gadge Baba Amravati University</td><td>1127.0</td><td>Jagadamba Bahuuddesha Gramin Vikas Sanstha...</td><td>112719110.0</td><td>Civil Engineering</td><td>37.206627</td></tr> <tr> <td>181</td><td>Dr. Babasaheb Ambedkar Technological Universit...</td><td>2131.0</td><td>- Terna Public Charitable Trust's College of E...</td><td>213119110.0</td><td>Civil Engineering</td><td>36.682710</td></tr> <tr> <td>81</td><td>Sant Gadge Baba Amravati University</td><td>1128.0</td><td>Prof Ram Meghe College of Engineering and Mana...</td><td>112819110.0</td><td>Civil Engineering</td><td>35.484093</td></tr> <tr> <td>262</td><td>Dr. Babasaheb Ambedkar Technological Universit...</td><td>3033.0</td><td>Dr. Babasaheb Ambedkar Technological Universit...</td><td>303319110.0</td><td>Civil Engineering</td><td>35.090615</td></tr> </tbody> </table>		h_u	c_code		cn	B_C	B_N	OPEN	1092	Dr. Babasaheb Ambedkar Technological Universit...	6545.0	Samarth Education Trust's Arvind Gavali Colleg...	654519110.0	Civil Engineering	38.173880	76	Sant Gadge Baba Amravati University	1127.0	Jagadamba Bahuuddesha Gramin Vikas Sanstha...	112719110.0	Civil Engineering	37.206627	181	Dr. Babasaheb Ambedkar Technological Universit...	2131.0	- Terna Public Charitable Trust's College of E...	213119110.0	Civil Engineering	36.682710	81	Sant Gadge Baba Amravati University	1128.0	Prof Ram Meghe College of Engineering and Mana...	112819110.0	Civil Engineering	35.484093	262	Dr. Babasaheb Ambedkar Technological Universit...	3033.0	Dr. Babasaheb Ambedkar Technological Universit...	303319110.0	Civil Engineering	35.090615
	h_u	c_code		cn	B_C	B_N	OPEN																																					
1092	Dr. Babasaheb Ambedkar Technological Universit...	6545.0	Samarth Education Trust's Arvind Gavali Colleg...	654519110.0	Civil Engineering	38.173880																																						
76	Sant Gadge Baba Amravati University	1127.0	Jagadamba Bahuuddesha Gramin Vikas Sanstha...	112719110.0	Civil Engineering	37.206627																																						
181	Dr. Babasaheb Ambedkar Technological Universit...	2131.0	- Terna Public Charitable Trust's College of E...	213119110.0	Civil Engineering	36.682710																																						
81	Sant Gadge Baba Amravati University	1128.0	Prof Ram Meghe College of Engineering and Mana...	112819110.0	Civil Engineering	35.484093																																						
262	Dr. Babasaheb Ambedkar Technological Universit...	3033.0	Dr. Babasaheb Ambedkar Technological Universit...	303319110.0	Civil Engineering	35.090615																																						

Page	MarkDown	Run All	Clear All Outputs	Outline		
289	Mumbai University	3139.0	Vidyalankar Institute of Technology,Wadala, M...	313924510.0	Computer Engineering	96.668239
345	Mumbai University	3184.0	Fr. Conceicao Rodrigues College of Engineering...	318424510.0	Computer Engineering	96.657040
458	Mumbai University	3209.0	K J Somaiya Institute of Engineering and Infor...	320924610.0	Information Technology	96.573961
405	Mumbai University	3197.0	Agnel Charities' FR. C. Rodrigues Institute o...	319724510.0	Computer Engineering	96.516323
326	Mumbai University	3176.0	Thakur College of Engineering and Technology, ...	317624510.0	Computer Engineering	96.118317
453	Mumbai University	3208.0	Don Bosco Institute of Technology, Mumbai	320824510.0	Computer Engineering	95.836975
406	Mumbai University	3197.0	Agnel Charities' FR. C. Rodrigues Institute o...	319724610.0	Information Technology	95.761400
290	Mumbai University	3139.0	Vidyalankar Institute of Technology,Wadala, M...	313924610.0	Information Technology	95.617153
442	Mumbai University	3204.0	St. Francis Institute of Technology,Borivali,...	320424510.0	Computer Engineering	95.296693
464	Mumbai University	3211.0	S.I.E.S. Graduate School of Technology, Nerul,...	321124510.0	Computer Engineering	95.222522
454	Mumbai University	3208.0	Don Bosco Institute of Technology, Mumbai	320824610.0	Information Technology	95.061748
465	Mumbai University	3211.0	S.I.E.S. Graduate School of Technology, Nerul,...	321124610.0	Information Technology	95.045866
327	Mumbai University	3176.0	Thakur College of Engineering and Technology, ...	317624610.0	Information Technology	94.882958
474	Mumbai University	3214.0	Xavier Institute Of Engineering C/O Xavier Tec...	321424510.0	Computer Engineering	94.518918
284	Mumbai University	3135.0	Manjara Charitable Trust's Rajiv Gandhi Instit...	313524510.0	Computer Engineering	94.380211
1071	Savitribai Phule Pune University	6004.0	Government College of Engineering & Research,...	600424510.0	Computer Engineering	94.334795
443	Mumbai University	3204.0	St. Francis Institute of Technology,Borivali,...	320424610.0	Information Technology	93.972153
366	Mumbai University	3189.0	Padmabhushan Vasantdada Patil Pratishthans Col...	318924510.0	Computer Engineering	93.777554
866	Savitribai Phule Pune University	5121.0	K. K. Wagh Institute of Engineering Education ...	512124510.0	Computer Engineering	93.705224
430	Mumbai University	3201.0	Rizvi Education Society's Rizvi College of Eng...	320124510.0	Computer Engineering	93.562109

Jupyter Server: Local

	h_u	c_code		cn	B.C	B.N	OPEN_OH
333	Mumbai University	3182.0	Thadomal Shahani Engineering College, Bandra,...	318224510.0	Computer Engineering	96.948517	
349	Mumbai University	3185.0	Vivekanand Education Society's Institute of Te...	318524510.0	Computer Engineering	96.930696	
474	Mumbai University	3214.0	Xavier Institute Of Engineering C/O Xavier Tec...	321424510.0	Computer Engineering	96.333425	
1071	Savitribai Phule Pune University	6004.0	Government College of Engineering & Research,...	600424510.0	Computer Engineering	95.845273	
345	Mumbai University	3184.0	Fr. Conceicao Rodrigues College of Engineering...	318424510.0	Computer Engineering	94.974869	
405	Mumbai University	3197.0	Agnel Charities' FR. C. Rodrigues Institute o...	319724510.0	Computer Engineering	94.424351	
366	Mumbai University	3189.0	Padmabhushan Vasantdada Patil Pratishthans Col...	318924510.0	Computer Engineering	93.066001	

```

r_c = df[((df['h_u'] == 'Savitribai Phule Pune University') | (df['h_u'] == 'Rashtrasant Tukadoji Maharaj Nagpur University')) & (df['B_N'] == r_c[(r_c['VJ'] >= 30) & (r_c['VJ'] <=35)]]
r_c = r_c.sort_values(by='VJ', ascending=False)
r_c[['h_u', 'c_code', 'cn', 'B_C', 'B_N', 'VJ']]

```

	h_u	c_code		cn	B.C	B.N	VJ
764	Rashtrasant Tukadoji Maharaj Nagpur University	4167.0	Yeshwantrao Chavan College of Engineering,Wana...	416761210.0	Mechanical Engineering	32.150713	
618	Rashtrasant Tukadoji Maharaj Nagpur University	4025.0	Government College of Engineering, Nagpur	402561210.0	Mechanical Engineering	31.501519	
932	Savitribai Phule Pune University	5162.0	Amrutvahini Sheti & Shikshan Vikas Sanstha's ...	516261210.0	Mechanical Engineering	31.501519	

Smart Bot code:

```

let VSBoxCounter = function() {
    let e = [],
        i = [];
    return {
        set: function(t) {
            i.push({
                offset: ++e,
                ptr: t
            });
            return i[i.length - 1].offset
        },
        remove: function(e) {
            let t = i.filter(function(t) {
                return t.offset != e
            });
            i = t.splice(0)
        },
        closeAllButMe: function(e) {
            i.forEach(function(t) {
                t.offset != e && t.ptr.closeOrder()
            })
        }
    }
};

function vanillaSelectBox(t, e) {
    let u = this;
    this.instanceOffset = VSBoxCounter.set(u);
    this.domSelector = t;
    this.root = document.querySelector(t);
    this.rootToken = null;
}

```

Figure 12: Smart bot

```
F: > study > 4th year > Major project > www.smatbot.com > main.js > JS vanillaSelectBox.js - Visual Studio Code
Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More
JS responsivevoice.js JS vanillaSelectBox.js Release Notes: 1.77.3
59     this.forbiddenAttributes = ["class", "selected", "disabled", "data-text", "data-value", "style"];
60     this.forbiddenClasses = ["active", "disabled"];
61     this.userOptions = {
62         maxWidth: 500,
63         minWidth: -1,
64         maxHeight: 400,
65         translations: {
66             all: "All",
67             items: "Items",
68             selectAll: "Select All",
69             clearAll: "Clear All"
70         },
71         search: !1,
72         placeHolder: "",
73         stayOpen: !1,
74         disableSelectAll: !1
75     };
76     if (e) {
77         null != e.maxWidth && (this.userOptions.maxWidth = e.maxWidth);
78         null != e.minLength && (this.userOptions.minLength = e.minLength);
79         null != e.maxLength && (this.userOptions.maxLength = e.maxLength);
80         if (null != e.translations)
81             for (var i in e.translations) e.translations.hasOwnProperty(i) && this.userOptions.translations[i] && (this.userOptions.translations[i] = e.translations[i]);
82         null != e.placeHolder && (this.userOptions.placeHolder = e.placeHolder);
83         null != e.search && (this.userOptions.search = e.search);
84         if (null != e.remote && e.remote) {
85             if (null != e.remote.onInit && "function" == typeof e.remote.onInit) {
86                 this.onInit = e.remote.onInit;
87                 this.isInitRemote = !0
88             }
89             if (null != e.remote.onInitSize) {
90                 this.onInitSize = e.remote.onInitSize;
91                 this.onInitSize < 3 && (this.onInitSize = 3)
92             }
93             if (null != e.remote.onSearch && "function" == typeof e.remote.onSearch) {
94                 this.onSearch = e.remote.onSearch;
95                 this.isSearchRemote = !0
96             }
97             null != e.stayOpen && (this.userOptions.stayOpen = e.stayOpen);
98             null != e.disableSelectAll && (this.userOptions.disableSelectAll = e.disableSelectAll);
99             if (null != e.maxSelect && isNaN(e.maxSelect) && 1 <= e.maxSelect)
100                 this.maxSelect = e.maxSelect;
101                 this.userOptions.disableSelectAll = !0
102             }
103             if (null != e.maxOptionWidth && !isNaN(e.maxOptionWidth) && 20 <= e.maxOptionWidth) {
104                 this.maxOptionWidth = e.maxOptionWidth;
105                 this.ulMinWidth = e.maxOptionWidth + 60;
106                 this.ulMaxWidth = e.maxOptionWidth + 60
107             }
108         }
109         this.closeOrder = function() {
110             var t = this;
111             if (!t.userOptions.stayOpen)
112                 t.drop.style.visibility = "hidden";
113                 if (t.search) {
114                     t.inputBox.value = "";
115                     Array.prototype.slice.call(t.listElements).forEach(function(t) {
116                         t.classList.remove("hide")
117                     })
118                 }
119         }
120     }
121     this.openOrder = function() {
122         var t = this;
123         if (!t.userOptions.stayOpen)
124             t.drop.style.visibility = "visible";
125             if (t.search) {
126                 t.inputBox.value = "";
127                 Array.prototype.slice.call(t.listElements).forEach(function(t) {
128                     t.classList.add("hide")
129                 })
130             }
131     }
132     this.setSearchValue = function(t) {
133         var e = this;
134         if (e.search)
135             e.inputBox.value = t;
136     }
137     this.setPlaceholder = function(t) {
138         var e = this;
139         if (e.placeHolder)
140             e.inputBox.placeholder = t;
141     }
142     this.setTranslations = function(t) {
143         var e = this;
144         if (e.translations)
145             e.translations = t;
146     }
147     this.setMinLength = function(t) {
148         var e = this;
149         if (e.minLength)
150             e.minLength = t;
151     }
152     this.setMaxLength = function(t) {
153         var e = this;
154         if (e.maxLength)
155             e.maxLength = t;
156     }
157     this.setStayOpen = function(t) {
158         var e = this;
159         if (e.stayOpen)
160             e.stayOpen = t;
161     }
162     this.setDisableSelectAll = function(t) {
163         var e = this;
164         if (e.disableSelectAll)
165             e.disableSelectAll = t;
166     }
167     this.setRemote = function(t) {
168         var e = this;
169         if (e.remote)
170             e.remote = t;
171     }
172     this.setInit = function(t) {
173         var e = this;
174         if (e.onInit)
175             e.onInit = t;
176     }
177     this.setInitSize = function(t) {
178         var e = this;
179         if (e.onInitSize)
180             e.onInitSize = t;
181     }
182     this.setSearch = function(t) {
183         var e = this;
184         if (e.search)
185             e.search = t;
186     }
187     this.setPlaceHolder = function(t) {
188         var e = this;
189         if (e.placeHolder)
190             e.placeHolder = t;
191     }
192     this.setTranslations = function(t) {
193         var e = this;
194         if (e.translations)
195             e.translations = t;
196     }
197     this.setMinWidth = function(t) {
198         var e = this;
199         if (e.minLength)
200             e.minLength = t;
201     }
202     this.setMaxWidth = function(t) {
203         var e = this;
204         if (e.maxLength)
205             e.maxLength = t;
206     }
207     this.setStayOpen = function(t) {
208         var e = this;
209         if (e.stayOpen)
210             e.stayOpen = t;
211     }
212     this.setDisableSelectAll = function(t) {
213         var e = this;
214         if (e.disableSelectAll)
215             e.disableSelectAll = t;
216     }
217     this.setRemote = function(t) {
218         var e = this;
219         if (e.remote)
220             e.remote = t;
221     }
222     this.setInit = function(t) {
223         var e = this;
224         if (e.onInit)
225             e.onInit = t;
226     }
227     this.setInitSize = function(t) {
228         var e = this;
229         if (e.onInitSize)
230             e.onInitSize = t;
231     }
232     this.setSearch = function(t) {
233         var e = this;
234         if (e.search)
235             e.search = t;
236     }
237     this.setPlaceHolder = function(t) {
238         var e = this;
239         if (e.placeHolder)
240             e.placeHolder = t;
241     }
242     this.setTranslations = function(t) {
243         var e = this;
244         if (e.translations)
245             e.translations = t;
246     }
247     this.setMinWidth = function(t) {
248         var e = this;
249         if (e.minLength)
250             e.minLength = t;
251     }
252     this.setMaxWidth = function(t) {
253         var e = this;
254         if (e.maxLength)
255             e.maxLength = t;
256     }
257     this.setStayOpen = function(t) {
258         var e = this;
259         if (e.stayOpen)
260             e.stayOpen = t;
261     }
262     this.setDisableSelectAll = function(t) {
263         var e = this;
264         if (e.disableSelectAll)
265             e.disableSelectAll = t;
266     }
267     this.setRemote = function(t) {
268         var e = this;
269         if (e.remote)
270             e.remote = t;
271     }
272     this.setInit = function(t) {
273         var e = this;
274         if (e.onInit)
275             e.onInit = t;
276     }
277     this.setInitSize = function(t) {
278         var e = this;
279         if (e.onInitSize)
280             e.onInitSize = t;
281     }
282     this.setSearch = function(t) {
283         var e = this;
284         if (e.search)
285             e.search = t;
286     }
287     this.setPlaceHolder = function(t) {
288         var e = this;
289         if (e.placeHolder)
290             e.placeHolder = t;
291     }
292     this.setTranslations = function(t) {
293         var e = this;
294         if (e.translations)
295             e.translations = t;
296     }
297     this.setMinWidth = function(t) {
298         var e = this;
299         if (e.minLength)
300             e.minLength = t;
301     }
302     this.setMaxWidth = function(t) {
303         var e = this;
304         if (e.maxLength)
305             e.maxLength = t;
306     }
307     this.setStayOpen = function(t) {
308         var e = this;
309         if (e.stayOpen)
310             e.stayOpen = t;
311     }
312     this.setDisableSelectAll = function(t) {
313         var e = this;
314         if (e.disableSelectAll)
315             e.disableSelectAll = t;
316     }
317     this.setRemote = function(t) {
318         var e = this;
319         if (e.remote)
320             e.remote = t;
321     }
322     this.setInit = function(t) {
323         var e = this;
324         if (e.onInit)
325             e.onInit = t;
326     }
327     this.setInitSize = function(t) {
328         var e = this;
329         if (e.onInitSize)
330             e.onInitSize = t;
331     }
332     this.setSearch = function(t) {
333         var e = this;
334         if (e.search)
335             e.search = t;
336     }
337     this.setPlaceHolder = function(t) {
338         var e = this;
339         if (e.placeHolder)
340             e.placeHolder = t;
341     }
342     this.setTranslations = function(t) {
343         var e = this;
344         if (e.translations)
345             e.translations = t;
346     }
347     this.setMinWidth = function(t) {
348         var e = this;
349         if (e.minLength)
350             e.minLength = t;
351     }
352     this.setMaxWidth = function(t) {
353         var e = this;
354         if (e.maxLength)
355             e.maxLength = t;
356     }
357     this.setStayOpen = function(t) {
358         var e = this;
359         if (e.stayOpen)
360             e.stayOpen = t;
361     }
362     this.setDisableSelectAll = function(t) {
363         var e = this;
364         if (e.disableSelectAll)
365             e.disableSelectAll = t;
366     }
367     this.setRemote = function(t) {
368         var e = this;
369         if (e.remote)
370             e.remote = t;
371     }
372     this.setInit = function(t) {
373         var e = this;
374         if (e.onInit)
375             e.onInit = t;
376     }
377     this.setInitSize = function(t) {
378         var e = this;
379         if (e.onInitSize)
380             e.onInitSize = t;
381     }
382     this.setSearch = function(t) {
383         var e = this;
384         if (e.search)
385             e.search = t;
386     }
387     this.setPlaceHolder = function(t) {
388         var e = this;
389         if (e.placeHolder)
390             e.placeHolder = t;
391     }
392     this.setTranslations = function(t) {
393         var e = this;
394         if (e.translations)
395             e.translations = t;
396     }
397     this.setMinWidth = function(t) {
398         var e = this;
399         if (e.minLength)
400             e.minLength = t;
401     }
402     this.setMaxWidth = function(t) {
403         var e = this;
404         if (e.maxLength)
405             e.maxLength = t;
406     }
407     this.setStayOpen = function(t) {
408         var e = this;
409         if (e.stayOpen)
410             e.stayOpen = t;
411     }
412     this.setDisableSelectAll = function(t) {
413         var e = this;
414         if (e.disableSelectAll)
415             e.disableSelectAll = t;
416     }
417     this.setRemote = function(t) {
418         var e = this;
419         if (e.remote)
420             e.remote = t;
421     }
422     this.setInit = function(t) {
423         var e = this;
424         if (e.onInit)
425             e.onInit = t;
426     }
427     this.setInitSize = function(t) {
428         var e = this;
429         if (e.onInitSize)
430             e.onInitSize = t;
431     }
432     this.setSearch = function(t) {
433         var e = this;
434         if (e.search)
435             e.search = t;
436     }
437     this.setPlaceHolder = function(t) {
438         var e = this;
439         if (e.placeHolder)
440             e.placeHolder = t;
441     }
442     this.setTranslations = function(t) {
443         var e = this;
444         if (e.translations)
445             e.translations = t;
446     }
447     this.setMinWidth = function(t) {
448         var e = this;
449         if (e.minLength)
450             e.minLength = t;
451     }
452     this.setMaxWidth = function(t) {
453         var e = this;
454         if (e.maxLength)
455             e.maxLength = t;
456     }
457     this.setStayOpen = function(t) {
458         var e = this;
459         if (e.stayOpen)
460             e.stayOpen = t;
461     }
462     this.setDisableSelectAll = function(t) {
463         var e = this;
464         if (e.disableSelectAll)
465             e.disableSelectAll = t;
466     }
467     this.setRemote = function(t) {
468         var e = this;
469         if (e.remote)
470             e.remote = t;
471     }
472     this.setInit = function(t) {
473         var e = this;
474         if (e.onInit)
475             e.onInit = t;
476     }
477     this.setInitSize = function(t) {
478         var e = this;
479         if (e.onInitSize)
480             e.onInitSize = t;
481     }
482     this.setSearch = function(t) {
483         var e = this;
484         if (e.search)
485             e.search = t;
486     }
487     this.setPlaceHolder = function(t) {
488         var e = this;
489         if (e.placeHolder)
490             e.placeHolder = t;
491     }
492     this.setTranslations = function(t) {
493         var e = this;
494         if (e.translations)
495             e.translations = t;
496     }
497     this.setMinWidth = function(t) {
498         var e = this;
499         if (e.minLength)
500             e.minLength = t;
501     }
502     this.setMaxWidth = function(t) {
503         var e = this;
504         if (e.maxLength)
505             e.maxLength = t;
506     }
507     this.setStayOpen = function(t) {
508         var e = this;
509         if (e.stayOpen)
510             e.stayOpen = t;
511     }
512     this.setDisableSelectAll = function(t) {
513         var e = this;
514         if (e.disableSelectAll)
515             e.disableSelectAll = t;
516     }
517     this.setRemote = function(t) {
518         var e = this;
519         if (e.remote)
520             e.remote = t;
521     }
522     this.setInit = function(t) {
523         var e = this;
524         if (e.onInit)
525             e.onInit = t;
526     }
527     this.setInitSize = function(t) {
528         var e = this;
529         if (e.onInitSize)
530             e.onInitSize = t;
531     }
532     this.setSearch = function(t) {
533         var e = this;
534         if (e.search)
535             e.search = t;
536     }
537     this.setPlaceHolder = function(t) {
538         var e = this;
539         if (e.placeHolder)
540             e.placeHolder = t;
541     }
542     this.setTranslations = function(t) {
543         var e = this;
544         if (e.translations)
545             e.translations = t;
546     }
547     this.setMinWidth = function(t) {
548         var e = this;
549         if (e.minLength)
550             e.minLength = t;
551     }
552     this.setMaxWidth = function(t) {
553         var e = this;
554         if (e.maxLength)
555             e.maxLength = t;
556     }
557     this.setStayOpen = function(t) {
558         var e = this;
559         if (e.stayOpen)
560             e.stayOpen = t;
561     }
562     this.setDisableSelectAll = function(t) {
563         var e = this;
564         if (e.disableSelectAll)
565             e.disableSelectAll = t;
566     }
567     this.setRemote = function(t) {
568         var e = this;
569         if (e.remote)
570             e.remote = t;
571     }
572     this.setInit = function(t) {
573         var e = this;
574         if (e.onInit)
575             e.onInit = t;
576     }
577     this.setInitSize = function(t) {
578         var e = this;
579         if (e.onInitSize)
580             e.onInitSize = t;
581     }
582     this.setSearch = function(t) {
583         var e = this;
584         if (e.search)
585             e.search = t;
586     }
587     this.setPlaceHolder = function(t) {
588         var e = this;
589         if (e.placeHolder)
590             e.placeHolder = t;
591     }
592     this.setTranslations = function(t) {
593         var e = this;
594         if (e.translations)
595             e.translations = t;
596     }
597     this.setMinWidth = function(t) {
598         var e = this;
599         if (e.minLength)
600             e.minLength = t;
601     }
602     this.setMaxWidth = function(t) {
603         var e = this;
604         if (e.maxLength)
605             e.maxLength = t;
606     }
607     this.setStayOpen = function(t) {
608         var e = this;
609         if (e.stayOpen)
610             e.stayOpen = t;
611     }
612     this.setDisableSelectAll = function(t) {
613         var e = this;
614         if (e.disableSelectAll)
615             e.disableSelectAll = t;
616     }
617     this.setRemote = function(t) {
618         var e = this;
619         if (e.remote)
620             e.remote = t;
621     }
622     this.setInit = function(t) {
623         var e = this;
624         if (e.onInit)
625             e.onInit = t;
626     }
627     this.setInitSize = function(t) {
628         var e = this;
629         if (e.onInitSize)
630             e.onInitSize = t;
631     }
632     this.setSearch = function(t) {
633         var e = this;
634         if (e.search)
635             e.search = t;
636     }
637     this.setPlaceHolder = function(t) {
638         var e = this;
639         if (e.placeHolder)
640             e.placeHolder = t;
641     }
642     this.setTranslations = function(t) {
643         var e = this;
644         if (e.translations)
645             e.translations = t;
646     }
647     this.setMinWidth = function(t) {
648         var e = this;
649         if (e.minLength)
650             e.minLength = t;
651     }
652     this.setMaxWidth = function(t) {
653         var e = this;
654         if (e.maxLength)
655             e.maxLength = t;
656     }
657     this.setStayOpen = function(t) {
658         var e = this;
659         if (e.stayOpen)
660             e.stayOpen = t;
661     }
662     this.setDisableSelectAll = function(t) {
663         var e = this;
664         if (e.disableSelectAll)
665             e.disableSelectAll = t;
666     }
667     this.setRemote = function(t) {
668         var e = this;
669         if (e.remote)
670             e.remote = t;
671     }
672     this.setInit = function(t) {
673         var e = this;
674         if (e.onInit)
675             e.onInit = t;
676     }
677     this.setInitSize = function(t) {
678         var e = this;
679         if (e.onInitSize)
680             e.onInitSize = t;
681     }
682     this.setSearch = function(t) {
683         var e = this;
684         if (e.search)
685             e.search = t;
686     }
687     this.setPlaceHolder = function(t) {
688         var e = this;
689         if (e.placeHolder)
690             e.placeHolder = t;
691     }
692     this.setTranslations = function(t) {
693         var e = this;
694         if (e.translations)
695             e.translations = t;
696     }
697     this.setMinWidth = function(t) {
698         var e = this;
699         if (e.minLength)
700             e.minLength = t;
701     }
702     this.setMaxWidth = function(t) {
703         var e = this;
704         if (e.maxLength)
705             e.maxLength = t;
706     }
707     this.setStayOpen = function(t) {
708         var e = this;
709         if (e.stayOpen)
710             e.stayOpen = t;
711     }
712     this.setDisableSelectAll = function(t) {
713         var e = this;
714         if (e.disableSelectAll)
715             e.disableSelectAll = t;
716     }
717     this.setRemote = function(t) {
718         var e = this;
719         if (e.remote)
720             e.remote = t;
721     }
722     this.setInit = function(t) {
723         var e = this;
724         if (e.onInit)
725             e.onInit = t;
726     }
727     this.setInitSize = function(t) {
728         var e = this;
729         if (e.onInitSize)
730             e.onInitSize = t;
731     }
732     this.setSearch = function(t) {
733         var e = this;
734         if (e.search)
735             e.search = t;
736     }
737     this.setPlaceHolder = function(t) {
738         var e = this;
739         if (e.placeHolder)
740             e.placeHolder = t;
741     }
742     this.setTranslations = function(t) {
743         var e = this;
744         if (e.translations)
745             e.translations = t;
746     }
747     this.setMinWidth = function(t) {
748         var e = this;
749         if (e.minLength)
750             e.minLength = t;
751     }
752     this.setMaxWidth = function(t) {
753         var e = this;
754         if (e.maxLength)
755             e.maxLength = t;
756     }
757     this.setStayOpen = function(t) {
758         var e = this;
759         if (e.stayOpen)
760             e.stayOpen = t;
761     }
762     this.setDisableSelectAll = function(t) {
763         var e = this;
764         if (e.disableSelectAll)
765             e.disableSelectAll = t;
766     }
767     this.setRemote = function(t) {
768         var e = this;
769         if (e.remote)
770             e.remote = t;
771     }
772     this.setInit = function(t) {
773         var e = this;
774         if (e.onInit)
775             e.onInit = t;
776     }
777     this.setInitSize = function(t) {
778         var e = this;
779         if (e.onInitSize)
780             e.onInitSize = t;
781     }
782     this.setSearch = function(t) {
783         var e = this;
784         if (e.search)
785             e.search = t;
786     }
787     this.setPlaceHolder = function(t) {
788         var e = this;
789         if (e.placeHolder)
790             e.placeHolder = t;
791     }
792     this.setTranslations = function(t) {
793         var e = this;
794         if (e.translations)
795             e.translations = t;
796     }
797     this.setMinWidth = function(t) {
798         var e = this;
799         if (e.minLength)
800             e.minLength = t;
801     }
802     this.setMaxWidth = function(t) {
803         var e = this;
804         if (e.maxLength)
805             e.maxLength = t;
806     }
807     this.setStayOpen = function(t) {
808         var e = this;
809         if (e.stayOpen)
810             e.stayOpen = t;
811     }
812     this.setDisableSelectAll = function(t) {
813         var e = this;
814         if (e.disableSelectAll)
815             e.disableSelectAll = t;
816     }
817     this.setRemote = function(t) {
818         var e = this;
819         if (e.remote)
820             e.remote = t;
821     }
822     this.setInit = function(t) {
823         var e = this;
824         if (e.onInit)
825             e.onInit = t;
826     }
827     this.setInitSize = function(t) {
828         var e = this;
829         if (e.onInitSize)
830             e.onInitSize = t;
831     }
832     this.setSearch = function(t) {
833         var e = this;
834         if (e.search)
835             e.search = t;
836     }
837     this.setPlaceHolder = function(t) {
838         var e = this;
839         if (e.placeHolder)
840             e.placeHolder = t;
841     }
842     this.setTranslations = function(t) {
843         var e = this;
844         if (e.translations)
845             e.translations = t;
846     }
847     this.setMinWidth = function(t) {
848         var e = this;
849         if (e.minLength)
850             e.minLength = t;
851     }
852     this.setMaxWidth = function(t) {
853         var e = this;
854         if (e.maxLength)
855             e.maxLength = t;
856     }
857     this.setStayOpen = function(t) {
858         var e = this;
859         if (e.stayOpen)
860             e.stayOpen = t;
861     }
862     this.setDisableSelectAll = function(t) {
863         var e = this;
864         if (e.disableSelectAll)
865             e.disableSelectAll = t;
866     }
867     this.setRemote = function(t) {
868         var e = this;
869         if (e.remote)
870             e.remote = t;
871     }
872     this.setInit = function(t) {
873         var e = this;
874         if (e.onInit)
875             e.onInit = t;
876     }
877     this.setInitSize = function(t) {
878         var e = this;
879         if (e.onInitSize)
880             e.onInitSize = t;
881     }
882     this.setSearch = function(t) {
883         var e = this;
884         if (e.search)
885             e.search = t;
886     }
887     this.setPlaceHolder = function(t) {
888         var e = this;
889         if (e.placeHolder)
890             e.placeHolder = t;
891     }
892     this.setTranslations = function(t) {
893         var e = this;
894         if (e.translations)
895             e.translations = t;
896     }
897     this.setMinWidth = function(t) {
898         var e = this;
899         if (e.minLength)
900             e.minLength = t;
901     }
902     this.setMaxWidth = function(t) {
903         var e = this;
904         if (e.maxLength)
905             e.maxLength = t;
906     }
907     this.setStayOpen = function(t) {
908         var e = this;
909         if (e.stayOpen)
910             e.stayOpen = t;
911     }
912     this.setDisableSelectAll = function(t) {
913         var e = this;
914         if (e.disableSelectAll)
915             e.disableSelectAll = t;
916     }
917     this.setRemote = function(t) {
918         var e = this;
919         if (e.remote)
920             e.remote = t;
921     }
922     this.setInit = function(t) {
923         var e = this;
924         if (e.onInit)
925             e.onInit = t;
926     }
927     this.setInitSize = function(t) {
928         var e = this;
929         if (e.onInitSize)
930             e.onInitSize = t;
931     }
932     this.setSearch = function(t) {
933         var e = this;
934         if (e.search)
935             e.search = t;
936     }
937     this.setPlaceHolder = function(t) {
938         var e = this;
939         if (e.placeHolder)
940             e.placeHolder = t;
941     }
942     this.setTranslations = function(t) {
943         var e = this;
944         if (e.translations)
945             e.translations = t;
946     }
947     this.setMinWidth = function(t) {
948         var e = this;
949         if (e.minLength)
950             e.minLength = t;
951     }
952     this.setMaxWidth = function(t) {
953         var e = this;
954         if (e.maxLength)
955             e.maxLength = t;
956     }
957     this.setStayOpen = function(t) {
958         var e = this;
959         if (e.stayOpen)
960             e.stayOpen = t;
961     }
962     this.setDisableSelectAll = function(t) {
963         var e = this;
964         if (e.disableSelectAll)
965             e.disableSelectAll = t;
966     }
967     this.setRemote = function(t) {
968         var e = this;
969         if (e.remote)
970             e.remote = t;
971     }
972     this.setInit = function(t) {
973         var e = this;
974         if (e.onInit)
975             e.onInit = t;
976     }
977     this.setInitSize = function(t) {
978         var e = this;
979         if (e.onInitSize)
980             e.onInitSize = t;
981     }
982     this.setSearch = function(t) {
983         var e = this;
984         if (e.search)
985             e.search = t;
986     }
987     this.setPlaceHolder = function(t) {
988         var e = this;
989         if (e.placeHolder)
990             e.placeHolder = t;
991     }
992     this.setTranslations = function(t) {
993         var e = this;
994         if (e.translations)
995             e.translations = t;
996     }
997     this.setMinWidth = function(t) {
998         var e = this;
999         if (e.minLength)
1000            e.minLength = t;
1001        }
1002        this.setMaxWidth = function(t) {
1003            var e = this;
1004            if (e.maxLength)
1005               e.maxLength = t;
1006            }
1007            this.setStayOpen = function(t) {
1008                var e = this;
1009                if (e.stayOpen)
1010                   e.stayOpen = t;
1011                }
1012                this.setDisableSelectAll = function(t) {
1013                    var e = this;
1014                    if (e.disableSelectAll)
1015                      e.disableSelectAll = t;
1016                    }
1017                    this.setRemote = function(t) {
1018                        var e = this;
1019                        if (e.remote)
1020                          e.remote = t;
1021                        }
1022                        this.setInit = function(t) {
1023                            var e = this;
1024                            if (e.onInit)
1025                              e.onInit = t;
1026                            }
1027                            this.setInitSize = function(t) {
1028                                var e = this;
1029                                if (e.onInitSize)
1030                                  e.onInitSize = t;
1031                                }
1032                                this.setSearch = function(t) {
1033                                    var e = this;
1034                                    if (e.search)
1035                                      e.search = t;
1036                                    }
1037                                    this.setPlaceHolder = function(t) {
1038                                        var e = this;
1039                                        if (e.placeHolder)
1040                                          e.placeHolder = t;
1041                                        }
1042                                        this.setTranslations = function(t) {
1043                                            var e = this;
1044                                            if (e.translations)
1045                                              e.translations = t;
1046                                            }
1047                                            this.setMinWidth = function(t) {
1048                                                var e = this;
1049                                                if (e.minLength)
1050                                                  e.minLength = t;
1051                                                }
1052                                                this.setMaxWidth = function(t) {
1053                                                    var e = this;
1054                                                    if (e.maxLength)
1055                                                      e.maxLength = t;
1056                                                    }
1057                                                    this.setStayOpen = function(t) {
1058                                                        var e = this;
1059                                                        if (e.stayOpen)
1060                                                          e.stayOpen = t;
1061                                                        }
1062                                                        this.setDisableSelectAll = function(t) {
1063                                                            var e = this;
1064                                                            if (e.disableSelectAll)
1065                                                              e.disableSelectAll = t;
1066                                                            }
1067                                                            this.setRemote = function(t) {
1068                                                                var e = this;
1069                                                                if (e.remote)
1070                                                                  e.remote = t;
1071                                                                }
1072                                                                this.setInit = function(t) {
1073                                                                    var e = this;
1074                                                                    if (e.onInit)
1075                                                                      e.onInit = t;
1076                                                                    }
1077                                                                    this.setInitSize = function(t) {
1078                                                                        var e = this;
1079                                                                        if (e.onInitSize)
1080                                                                          e.onInitSize = t;
1081                                                                        }
1082                                                                        this.setSearch = function(t) {
1083                                var e = this;
1084                                if (e.search)
1085                                  e.search = t;
1086                                }
1087                                this.setPlaceHolder = function(t) {
1088                                    var e = this;
1089                                    if (e.placeHolder)
1090                                      e.placeHolder = t;
1091                                    }
1092                                    this.setTranslations = function(t) {
1093                                        var e = this;
1094                                        if (e.translations)
1095                                          e.translations = t;
1096                                        }
1097                                        this.setMinWidth = function(t) {
1098                                            var e = this;
1099                                            if (e.minLength)
1100                                              e.minLength = t;
1101                                              }
1102                                              this.setMaxWidth = function(t) {
1103                                                  var e = this;
1104                                                  if (e.maxLength)
1105                                                    e.maxLength = t;
1106                                                    }
1107                                                    this.setStayOpen = function(t) {
1108                                                        var e = this;
1109                                                        if (e.stayOpen)
1110                                                          e.stayOpen = t;
1111                                                          }
1112                                                          this.setDisableSelectAll = function(t) {
1113                                                              var e = this;
1114                                                              if (e.disableSelectAll)
1115                                                                e.disableSelectAll = t;
1116                                                                }
1117                                                                this.setRemote = function(t) {
1118                                                                    var e = this;
1119                                                                    if (e.remote)
1120                                                                      e.remote = t;
1121                                                                      }
1122                                                                      this.setInit = function(t) {
1123                                                                          var e = this;
1124                                                                          if (e.onInit)
1125                                                                            e.onInit = t;
1126                                                                          }
1127                                                                          this.setInitSize = function(t) {
1128                                                                              var e = this;
1129                                                                              if (e.onInitSize)
1130                                                                                e.onInitSize = t;
1131                                                                              }
1132                                                                              this.setSearch = function(t) {
1133                                    var e = this;
1134                                    if (e.search)
1135                                      e.search = t;
1136                                    }
1137                                    this.setPlaceHolder = function(t) {
1138                                        var e = this;
1139                                        if (e.placeHolder)
1140                                          e.placeHolder = t;
1141                                        }
1142                                        this.setTranslations = function(t) {
1143                                            var e = this;
1144                                            if (e.translations)
1145                                              e.translations = t;
1146                                            }
1147                                            this.setMinWidth = function(t) {
1148                                                var e = this;
1149                                                if (e.minLength)
1150                                                  e.minLength = t;
1151                                                  }
1152                                                  this.setMaxWidth = function(t) {
1153                                                      var e = this;
1154                                                      if (e.maxLength)
1155                                                        e.maxLength = t;
1156                                                        }
1157                                                        this.setStayOpen = function(t) {
1158                                                            var e = this;
1159                                                            if (e.stayOpen)
1160                                                              e.stayOpen = t;
1161                                                              }
1162                                                              this.setDisableSelectAll = function(t) {
1163                                                                  var e = this;
1164                                                                  if (e.disableSelectAll)
1165                                                                    e.disableSelectAll = t;
1166                                                                    }
1167                                                                    this.setRemote = function(t) {
1168                                                                        var e = this;
1169                                                                        if (e.remote)
1170                                                                          e.remote = t;
1171                                                                          }
1172                                                                          this.setInit = function(t) {
1173                                                                              var e = this;
1174                                                                              if (e.onInit)
1175                                                                                e.onInit = t;
1176                                                                              }
1177                                                                              this.setInitSize = function(t) {
1178                                                                                  var e = this;
1179                                                                                  if (e.onInitSize)
1180                                                                                    e.onInitSize = t;
1181                                                                                  }
1182                                                                                  this.setSearch = function(t) {
1183                        var e = this;
1184                        if (e.search)
1185                          e.search = t;
1186                        }
1187                        this.setPlaceHolder = function(t) {
1188                            var e = this;
1189                            if (e.placeHolder)
1190                              e.placeHolder = t;
1191                            }
1192                            this.setTranslations = function(t) {
1193                                var e = this;
1194                                if (e.translations)

```

```
File Edit Selection View Go Run Terminal Help
F: study > 4th year > Major project > www.smatbot.com > main.js > JS vanillaSelectBox.js - Visual Studio Code
Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More
JS responsivevoice.js JS vanillaSelectBox.js x Release Notes: 1.77.3
30     this.root = document.querySelector(t);
31     this.rootToken = null;
32     this.main;
33     this.button;
34     this.title;
35     this.isMultiple = this.root.hasAttribute("multiple");
36     this.multipleSize = this.isMultiple && this.root.getAttribute("size") ? parseInt(this.root.getAttribute("size")) : -1;
37     this.isOptgroups = !1;
38     this.currentOptgroup = 0;
39     this.drop;
40     this.top;
41     this.left;
42     this.options;
43     this.listElements;
44     this.isDisabled = !1;
45     this.search = !1;
46     this.searchZone = null;
47     this.inputBox = null;
48     this.disabledItems = [];
49     this.ulMinWidth = 140;
50     this.ulMaxWidth = 280;
51     this.ulMinHeight = 25;
52     this.maxOptionWidth = 1 / 0;
53     this.maxSelect = 1 / 0;
54     this.isInitRemote = !1;
55     this.isSearchRemote = !1;
56     this.onInit = null;
57     this.onSearch = null;
58     this.onInitSize = null;
59     this.forbiddenAttributes = ["class", "selected", "disabled", "data-text", "data-value", "style"];
60     this.forbiddenClasses = ["active", "disabled"];
Ln 1, Col 1 Spaces: 4 UTF-8 LF () JavaScript
```

```
File Edit Selection View Go Run Terminal Help
F: study > 4th year > Major project > guest.getResponse.chat > JS initialization.js - Visual Studio Code
Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More
JS initialization.js x
1 function signInUser() {
2     if (window._chatBackendUrl) {
3         let unsubscribe = firebase.auth().onAuthStateChanged(function(user) {
4             unsubscribe();
5             if (!user) {
6                 let xhr = new XMLHttpRequest();
7                 xhr.open('GET', `${window._chatBackendUrl}/createGuest`);
8                 xhr.setRequestHeader('Content-Type', 'text/plain');
9                 xhr.onload = function() {
10                     if (xhr.responseText) {
11                         let response = JSON.parse(xhr.responseText);
12                         let token = response.data.success;
13                         firebase.auth().signInWithCustomToken(token);
14                     }
15                 };
16                 xhr.send();
17             }
18         });
19     }
20 }
21
22 function isSyncFlutter() {
23     return !window.location.href.includes("asyncFlutter=true");
24 }
25
26 function isAnalyticsDisabled() {
27     return window.location.href.includes("analyticsDisabled=true");
28 }
29
30 function loadFlutter() {
31     var script = document.createElement("script");
Ln 1, Col 1 Spaces: 4 UTF-8 LF () JavaScript
```

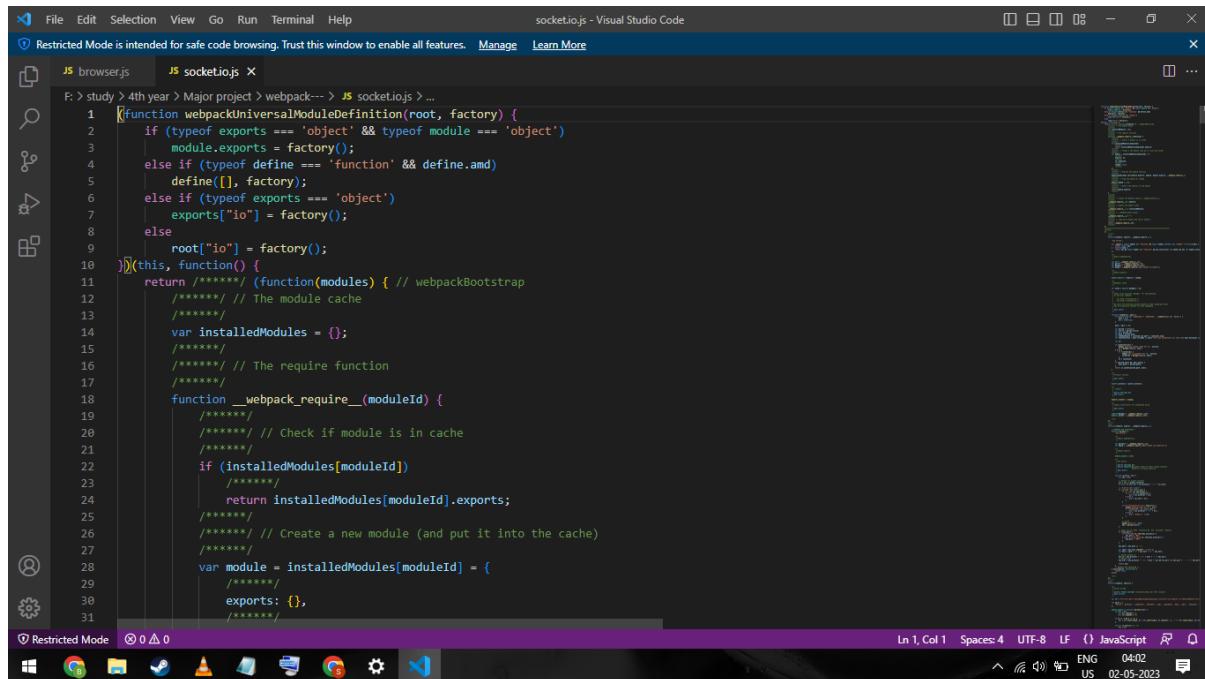
```
F: study > 4th year > Major project > guest.getResponse.chat > initialization.js > signInUser
30     function loadFlutter() {
31         var script = document.createElement("script");
32         script.src = "main.dart.js";
33         script.fetchPriority = "high";
34         document.documentElement.firstChild.appendChild(script);
35     }
36
37     function gtag() {
38         window.dataLayer.push(arguments);
39     }
40
41     function initializeAnalytics(tag) {
42         window.dataLayer = window.dataLayer || [];
43         if (!isAnalyticsDisabled()) {
44             gtag('js', new Date());
45             gtag('config', tag);
46             (function(w, d, s, i) {
47                 var f = d.getElementsByTagName(s)[0],
48                     j = d.createElement(s);
49                     j.async = true;
50                     j.src =
51                         'https://www.googletagmanager.com/gtag/js?id=' + i;
52                     f.parentNode.insertBefore(j, (parameter) tag: any);
53             })(window, document, 'script', tag);
54         }
55     }
56
57     function initializeApp(options) {
58         firebase.initializeApp(options);
59         firebase.performance();
60         firebase.analytics = function() {
61             return {};
62         };
63         window.addEventListener("message", function(event) {
64             if (event.data && event.data.action) {
65                 if (event.data.action === "@gr/chat/ready") {
66                     window._chatBackendUrl = event.data.data.backendUrl;
67                     window._chatSettings = JSON.stringify(event.data.data.settings);
68                     window.chatWindowInfo = {
69                         width: event.data.data.windowWidth,
70                         height: event.data.data.windowHeight,
71                         alignment: event.data.data.alignment
72                     };
73                     if (isSyncFlutter()) signInUser();
74                 } else if (event.data.action === "@gr/chat/windowResize") {
75                     window.chatWindowInfo = {
76                         width: event.data.data.windowWidth,
77                         height: event.data.data.windowHeight
78                     };
79                 } else if (event.data.action === "@gr/chat/loadFlutter") {
80                     loadFlutter();
81                     signInUser();
82                 }
83             }
84         }, false);
85         if (isSyncFlutter()) loadFlutter();
86     }

```

```
F: study > 4th year > Major project > guest.getResponse.chat > initialization.js > signInUser
56
57     function initializeApp(options) {
58         firebase.initializeApp(options);
59         firebase.performance();
60         firebase.analytics = function() {
61             return {};
62         };
63         window.addEventListener("message", function(event) {
64             if (event.data && event.data.action) {
65                 if (event.data.action === "@gr/chat/ready") {
66                     window._chatBackendUrl = event.data.data.backendUrl;
67                     window._chatSettings = JSON.stringify(event.data.data.settings);
68                     window.chatWindowInfo = {
69                         width: event.data.data.windowWidth,
70                         height: event.data.data.windowHeight,
71                         alignment: event.data.data.alignment
72                     };
73                     if (isSyncFlutter()) signInUser();
74                 } else if (event.data.action === "@gr/chat/windowResize") {
75                     window.chatWindowInfo = {
76                         width: event.data.data.windowWidth,
77                         height: event.data.data.windowHeight
78                     };
79                 } else if (event.data.action === "@gr/chat/loadFlutter") {
80                     loadFlutter();
81                     signInUser();
82                 }
83             }
84         }, false);
85         if (isSyncFlutter()) loadFlutter();
86     }

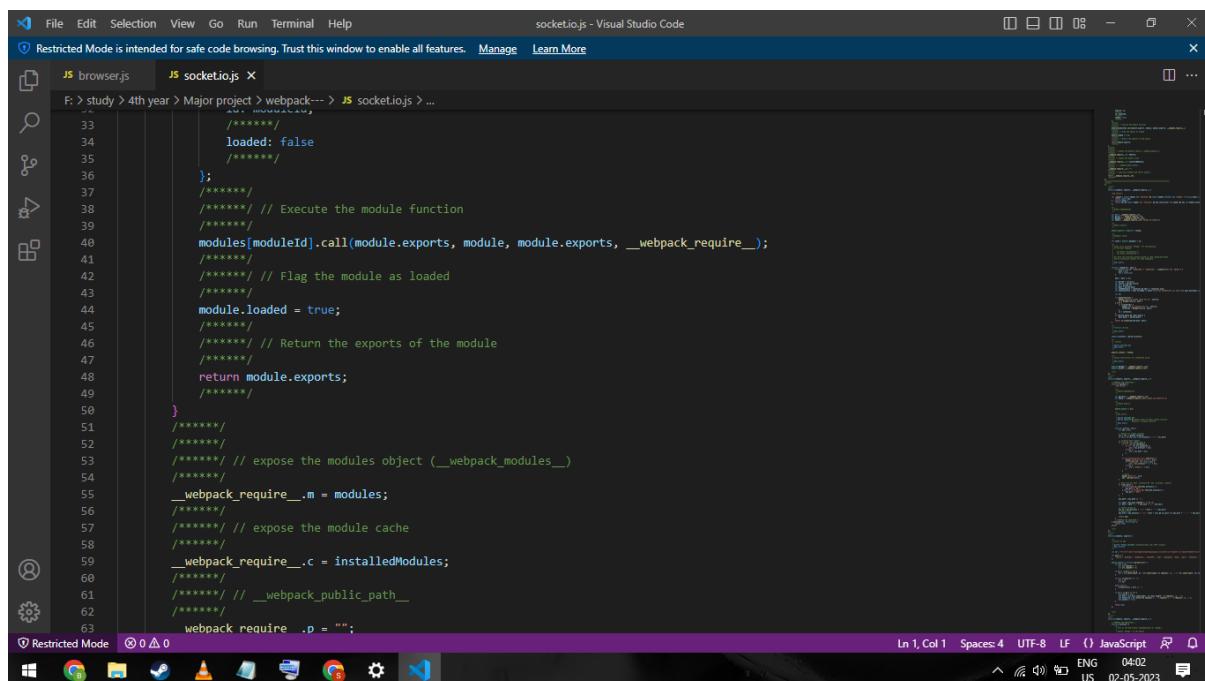
```

Website back-end and front-end:



A screenshot of the Visual Studio Code interface. The title bar says "socket.io.js - Visual Studio Code". The left sidebar shows file icons for "browser.js" and "socket.io.js". The main editor area displays the "socket.io.js" file, which contains approximately 31 lines of JavaScript code related to webpack Universal Module Definition (UMD) and module loading. The status bar at the bottom shows "Ln 1, Col 1" and "Spaces: 4".

Figure 13: Back-end



A screenshot of the Visual Studio Code interface, identical to Figure 13 but showing a different section of the "socket.io.js" file. The code continues from line 33 to line 63, dealing with module execution, loading, and exposing the modules object. The status bar at the bottom shows "Ln 1, Col 1" and "Spaces: 4".

The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** socketio.js - Visual Studio Code.
- Status Bar:** Restricted Mode is enabled. It also shows the current file path: F: > study > 4th year > Major project > webpack--- > JS socketio.js > ...
- Code Editor:** The main area displays the `socketio.js` file content. The code is a module that exports a `lookup` function. This function takes a `uri` and `opts` parameter. It first checks if `uri` is undefined or an object. If so, it sets `opts` to `uri` and `uri` to undefined. Then it initializes `opts` to an empty object if it's not already defined. The code includes comments explaining the logic for multiplexing based on the scheme and host.
- Sidebar:** On the left, there are several icons for file operations like Open, Save, Find, and Search.
- Right Panel:** A sidebar panel on the right shows a tree view of the project structure, including files like `index.html`, `server.js`, and `client.js`.
- Bottom Bar:** Shows the current line (Ln 1, Col 1), spaces used (Spaces: 4), and encoding (UTF-8). It also has tabs for Java and JavaScript, and a status message for the current file.

The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** socketio.js - Visual Studio Code.
- Status Bar:** Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More.
- Left Sidebar:** Includes icons for File, Find, Replace, Go To, and others.
- Central Area:** A code editor with the following content:

```
125     opts = opts || {};
126
127     var parsed = url(uri);
128     var source = parsed.source;
129     var id = parsed.id;
130     var path = parsed.path;
131     var sameNamespace = cache[id] && path in cache[id].nsps;
132     var newConnection = opts.forceNew || opts['force new connection'] || false === opts.multiplex || sameNamespace;
133
134     var io;
135
136     if (newConnection) {
137         debug('ignoring socket cache for %s', source);
138         io = Manager(source, opts);
139     } else {
140         if (!cache[id]) {
141             debug('new io instance for %s', source);
142             cache[id] = Manager(source, opts);
143         }
144         io = cache[id];
145     }
146     if (parsed.query && opts.query) {
147         opts.query = parsed.query;
148     }
149     return io.socket(parsed.path, opts);
150 }
151 /**
152 * Protocol version.
153 *
154 * @api public
155 */
```

The code is a function that creates an io socket based on configuration options and a parsed URL. It handles new connections and reuses existing ones from a cache. The protocol version is also checked and set.

The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** socketio.js - Visual Studio Code.
- Status Bar:** Restricted Mode is enabled. It also shows Ln 1, Col 1, Spaces: 4, UTF-8, LF, and a JavaScript icon.
- Left Sidebar:** Includes icons for File, Find, Replace, Go To, and a search bar.
- Right Sidebar:** Shows a large preview pane displaying the code structure and content of the file.
- Code Editor:** The main area displays the `socketio.js` file content. The code is a webpack module for the `socket.io` library, handling protocol parsing and exposing Manager and Socket constructors.

```
File Edit Selection View Go Run Terminal Help
Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More

JS browserjs JS socketio.js ×

F: > study > 4th year > Major project > webpack--- > JS socketio.js > ...
156     /*
157
158     exports.protocol = parser.protocol;
159
160     /**
161      * "connect".
162      *
163      * @param {String} uri
164      * @api public
165     */
166
167     exports.connect = lookup;
168
169     /**
170      * Expose constructors for standalone build.
171      *
172      * @api public
173     */
174
175     exports.Manager = __webpack_require__(12);
176     exports.Socket = __webpack_require__(37);
177
178     /**
179   },
180   /* 1 */
181   /**
182   (function(module, exports, __webpack_require__) {
183
184     /* WEBPACK VAR INJECTION */
185     (function(global) {
186       'use strict';
187
188       /**
189       * @param {Object} options
190       * @param {Function} [onClose]
191       */
192       function createSocket(options, onClose) {
193         const socket = new Manager();
194
195         if (options) {
196           if ('uri' in options) {
197             socket.connect(options.uri);
198           }
199
200           if ('onClose' in options) {
201             socket.on('close', options.onClose);
202           }
203         }
204
205         return socket;
206       }
207
208       module.exports = createSocket;
209     })(global, exports, __webpack_require__);
210   }
211   /**
212   (function() {
213     'use strict';
214
215     /**
216      * @param {Object} options
217      * @param {Function} [onClose]
218      */
219     function createManager(options, onClose) {
220       const manager = new Manager();
221
222       if (options) {
223         if ('uri' in options) {
224           manager.connect(options.uri);
225         }
226
227         if ('onClose' in options) {
228           manager.on('close', options.onClose);
229         }
230       }
231
232       return manager;
233     }
234
235     module.exports = createManager;
236   })();
237
238   /**
239   (function() {
240     'use strict';
241
242     /**
243      * @param {Object} options
244      * @param {Function} [onClose]
245      */
246     function createParser(options, onClose) {
247       const parser = new Parser();
248
249       if (options) {
250         if ('uri' in options) {
251           parser.connect(options.uri);
252         }
253
254         if ('onClose' in options) {
255           parser.on('close', options.onClose);
256         }
257       }
258
259       return parser;
260     }
261
262     module.exports = createParser;
263   })();
264
265   /**
266   (function() {
267     'use strict';
268
269     /**
270      * @param {Object} options
271      * @param {Function} [onClose]
272      */
273     function createParser(options, onClose) {
274       const parser = new Parser();
275
276       if (options) {
277         if ('uri' in options) {
278           parser.connect(options.uri);
279         }
280
281         if ('onClose' in options) {
282           parser.on('close', options.onClose);
283         }
284       }
285
286       return parser;
287     }
288
289     module.exports = createParser;
290   })();
291
292   /**
293   (function() {
294     'use strict';
295
296     /**
297      * @param {Object} options
298      * @param {Function} [onClose]
299      */
300     function createParser(options, onClose) {
301       const parser = new Parser();
302
303       if (options) {
304         if ('uri' in options) {
305           parser.connect(options.uri);
306         }
307
308         if ('onClose' in options) {
309           parser.on('close', options.onClose);
310         }
311       }
312
313       return parser;
314     }
315
316     module.exports = createParser;
317   })();
318
319   /**
320   (function() {
321     'use strict';
322
323     /**
324      * @param {Object} options
325      * @param {Function} [onClose]
326      */
327     function createParser(options, onClose) {
328       const parser = new Parser();
329
330       if (options) {
331         if ('uri' in options) {
332           parser.connect(options.uri);
333         }
334
335         if ('onClose' in options) {
336           parser.on('close', options.onClose);
337         }
338       }
339
340       return parser;
341     }
342
343     module.exports = createParser;
344   })();
345
346   /**
347   (function() {
348     'use strict';
349
350     /**
351      * @param {Object} options
352      * @param {Function} [onClose]
353      */
354     function createParser(options, onClose) {
355       const parser = new Parser();
356
357       if (options) {
358         if ('uri' in options) {
359           parser.connect(options.uri);
360         }
361
362         if ('onClose' in options) {
363           parser.on('close', options.onClose);
364         }
365       }
366
367       return parser;
368     }
369
370     module.exports = createParser;
371   })();
372
373   /**
374   (function() {
375     'use strict';
376
377     /**
378      * @param {Object} options
379      * @param {Function} [onClose]
380      */
381     function createParser(options, onClose) {
382       const parser = new Parser();
383
384       if (options) {
385         if ('uri' in options) {
386           parser.connect(options.uri);
387         }
388
389         if ('onClose' in options) {
390           parser.on('close', options.onClose);
391         }
392       }
393
394       return parser;
395     }
396
397     module.exports = createParser;
398   })();
399
400   /**
401   (function() {
402     'use strict';
403
404     /**
405      * @param {Object} options
406      * @param {Function} [onClose]
407      */
408     function createParser(options, onClose) {
409       const parser = new Parser();
410
411       if (options) {
412         if ('uri' in options) {
413           parser.connect(options.uri);
414         }
415
416         if ('onClose' in options) {
417           parser.on('close', options.onClose);
418         }
419       }
420
421       return parser;
422     }
423
424     module.exports = createParser;
425   })();
426
427   /**
428   (function() {
429     'use strict';
430
431     /**
432      * @param {Object} options
433      * @param {Function} [onClose]
434      */
435     function createParser(options, onClose) {
436       const parser = new Parser();
437
438       if (options) {
439         if ('uri' in options) {
440           parser.connect(options.uri);
441         }
442
443         if ('onClose' in options) {
444           parser.on('close', options.onClose);
445         }
446       }
447
448       return parser;
449     }
450
451     module.exports = createParser;
452   })();
453
454   /**
455   (function() {
456     'use strict';
457
458     /**
459      * @param {Object} options
460      * @param {Function} [onClose]
461      */
462     function createParser(options, onClose) {
463       const parser = new Parser();
464
465       if (options) {
466         if ('uri' in options) {
467           parser.connect(options.uri);
468         }
469
470         if ('onClose' in options) {
471           parser.on('close', options.onClose);
472         }
473       }
474
475       return parser;
476     }
477
478     module.exports = createParser;
479   })();
480
481   /**
482   (function() {
483     'use strict';
484
485     /**
486      * @param {Object} options
487      * @param {Function} [onClose]
488      */
489     function createParser(options, onClose) {
490       const parser = new Parser();
491
492       if (options) {
493         if ('uri' in options) {
494           parser.connect(options.uri);
495         }
496
497         if ('onClose' in options) {
498           parser.on('close', options.onClose);
499         }
500       }
501
502       return parser;
503     }
504
505     module.exports = createParser;
506   })();
507
508   /**
509   (function() {
510     'use strict';
511
512     /**
513      * @param {Object} options
514      * @param {Function} [onClose]
515      */
516     function createParser(options, onClose) {
517       const parser = new Parser();
518
519       if (options) {
520         if ('uri' in options) {
521           parser.connect(options.uri);
522         }
523
524         if ('onClose' in options) {
525           parser.on('close', options.onClose);
526         }
527       }
528
529       return parser;
530     }
531
532     module.exports = createParser;
533   })();
534
535   /**
536   (function() {
537     'use strict';
538
539     /**
540      * @param {Object} options
541      * @param {Function} [onClose]
542      */
543     function createParser(options, onClose) {
544       const parser = new Parser();
545
546       if (options) {
547         if ('uri' in options) {
548           parser.connect(options.uri);
549         }
550
551         if ('onClose' in options) {
552           parser.on('close', options.onClose);
553         }
554       }
555
556       return parser;
557     }
558
559     module.exports = createParser;
560   })();
561
562   /**
563   (function() {
564     'use strict';
565
566     /**
567      * @param {Object} options
568      * @param {Function} [onClose]
569      */
570     function createParser(options, onClose) {
571       const parser = new Parser();
572
573       if (options) {
574         if ('uri' in options) {
575           parser.connect(options.uri);
576         }
577
578         if ('onClose' in options) {
579           parser.on('close', options.onClose);
580         }
581       }
582
583       return parser;
584     }
585
586     module.exports = createParser;
587   })();
588
589   /**
590   (function() {
591     'use strict';
592
593     /**
594      * @param {Object} options
595      * @param {Function} [onClose]
596      */
597     function createParser(options, onClose) {
598       const parser = new Parser();
599
600       if (options) {
601         if ('uri' in options) {
602           parser.connect(options.uri);
603         }
604
605         if ('onClose' in options) {
606           parser.on('close', options.onClose);
607         }
608       }
609
610       return parser;
611     }
612
613     module.exports = createParser;
614   })();
615
616   /**
617   (function() {
618     'use strict';
619
620     /**
621      * @param {Object} options
622      * @param {Function} [onClose]
623      */
624     function createParser(options, onClose) {
625       const parser = new Parser();
626
627       if (options) {
628         if ('uri' in options) {
629           parser.connect(options.uri);
630         }
631
632         if ('onClose' in options) {
633           parser.on('close', options.onClose);
634         }
635       }
636
637       return parser;
638     }
639
640     module.exports = createParser;
641   })();
642
643   /**
644   (function() {
645     'use strict';
646
647     /**
648      * @param {Object} options
649      * @param {Function} [onClose]
650      */
651     function createParser(options, onClose) {
652       const parser = new Parser();
653
654       if (options) {
655         if ('uri' in options) {
656           parser.connect(options.uri);
657         }
658
659         if ('onClose' in options) {
660           parser.on('close', options.onClose);
661         }
662       }
663
664       return parser;
665     }
666
667     module.exports = createParser;
668   })();
669
670   /**
671   (function() {
672     'use strict';
673
674     /**
675      * @param {Object} options
676      * @param {Function} [onClose]
677      */
678     function createParser(options, onClose) {
679       const parser = new Parser();
680
681       if (options) {
682         if ('uri' in options) {
683           parser.connect(options.uri);
684         }
685
686         if ('onClose' in options) {
687           parser.on('close', options.onClose);
688         }
689       }
690
691       return parser;
692     }
693
694     module.exports = createParser;
695   })();
696
697   /**
698   (function() {
699     'use strict';
700
701     /**
702      * @param {Object} options
703      * @param {Function} [onClose]
704      */
705     function createParser(options, onClose) {
706       const parser = new Parser();
707
708       if (options) {
709         if ('uri' in options) {
710           parser.connect(options.uri);
711         }
712
713         if ('onClose' in options) {
714           parser.on('close', options.onClose);
715         }
716       }
717
718       return parser;
719     }
720
721     module.exports = createParser;
722   })();
723
724   /**
725   (function() {
726     'use strict';
727
728     /**
729      * @param {Object} options
730      * @param {Function} [onClose]
731      */
732     function createParser(options, onClose) {
733       const parser = new Parser();
734
735       if (options) {
736         if ('uri' in options) {
737           parser.connect(options.uri);
738         }
739
740         if ('onClose' in options) {
741           parser.on('close', options.onClose);
742         }
743       }
744
745       return parser;
746     }
747
748     module.exports = createParser;
749   })();
750
751   /**
752   (function() {
753     'use strict';
754
755     /**
756      * @param {Object} options
757      * @param {Function} [onClose]
758      */
759     function createParser(options, onClose) {
760       const parser = new Parser();
761
762       if (options) {
763         if ('uri' in options) {
764           parser.connect(options.uri);
765         }
766
767         if ('onClose' in options) {
768           parser.on('close', options.onClose);
769         }
770       }
771
772       return parser;
773     }
774
775     module.exports = createParser;
776   })();
777
778   /**
779   (function() {
780     'use strict';
781
782     /**
783      * @param {Object} options
784      * @param {Function} [onClose]
785      */
786     function createParser(options, onClose) {
787       const parser = new Parser();
788
789       if (options) {
790         if ('uri' in options) {
791           parser.connect(options.uri);
792         }
793
794         if ('onClose' in options) {
795           parser.on('close', options.onClose);
796         }
797       }
798
799       return parser;
800     }
801
802     module.exports = createParser;
803   })();
804
805   /**
806   (function() {
807     'use strict';
808
809     /**
810      * @param {Object} options
811      * @param {Function} [onClose]
812      */
813     function createParser(options, onClose) {
814       const parser = new Parser();
815
816       if (options) {
817         if ('uri' in options) {
818           parser.connect(options.uri);
819         }
820
821         if ('onClose' in options) {
822           parser.on('close', options.onClose);
823         }
824       }
825
826       return parser;
827     }
828
829     module.exports = createParser;
830   })();
831
832   /**
833   (function() {
834     'use strict';
835
836     /**
837      * @param {Object} options
838      * @param {Function} [onClose]
839      */
840     function createParser(options, onClose) {
841       const parser = new Parser();
842
843       if (options) {
844         if ('uri' in options) {
845           parser.connect(options.uri);
846         }
847
848         if ('onClose' in options) {
849           parser.on('close', options.onClose);
850         }
851       }
852
853       return parser;
854     }
855
856     module.exports = createParser;
857   })();
858
859   /**
860   (function() {
861     'use strict';
862
863     /**
864      * @param {Object} options
865      * @param {Function} [onClose]
866      */
867     function createParser(options, onClose) {
868       const parser = new Parser();
869
870       if (options) {
871         if ('uri' in options) {
872           parser.connect(options.uri);
873         }
874
875         if ('onClose' in options) {
876           parser.on('close', options.onClose);
877         }
878       }
879
880       return parser;
881     }
882
883     module.exports = createParser;
884   })();
885
886   /**
887   (function() {
888     'use strict';
889
890     /**
891      * @param {Object} options
892      * @param {Function} [onClose]
893      */
894     function createParser(options, onClose) {
895       const parser = new Parser();
896
897       if (options) {
898         if ('uri' in options) {
899           parser.connect(options.uri);
900         }
901
902         if ('onClose' in options) {
903           parser.on('close', options.onClose);
904         }
905       }
906
907       return parser;
908     }
909
910     module.exports = createParser;
911   })();
912
913   /**
914   (function() {
915     'use strict';
916
917     /**
918      * @param {Object} options
919      * @param {Function} [onClose]
920      */
921     function createParser(options, onClose) {
922       const parser = new Parser();
923
924       if (options) {
925         if ('uri' in options) {
926           parser.connect(options.uri);
927         }
928
929         if ('onClose' in options) {
930           parser.on('close', options.onClose);
931         }
932       }
933
934       return parser;
935     }
936
937     module.exports = createParser;
938   })();
939
940   /**
941   (function() {
942     'use strict';
943
944     /**
945      * @param {Object} options
946      * @param {Function} [onClose]
947      */
948     function createParser(options, onClose) {
949       const parser = new Parser();
950
951       if (options) {
952         if ('uri' in options) {
953           parser.connect(options.uri);
954         }
955
956         if ('onClose' in options) {
957           parser.on('close', options.onClose);
958         }
959       }
960
961       return parser;
962     }
963
964     module.exports = createParser;
965   })();
966
967   /**
968   (function() {
969     'use strict';
970
971     /**
972      * @param {Object} options
973      * @param {Function} [onClose]
974      */
975     function createParser(options, onClose) {
976       const parser = new Parser();
977
978       if (options) {
979         if ('uri' in options) {
980           parser.connect(options.uri);
981         }
982
983         if ('onClose' in options) {
984           parser.on('close', options.onClose);
985         }
986       }
987
988       return parser;
989     }
990
991     module.exports = createParser;
992   })();
993
994   /**
995   (function() {
996     'use strict';
997
998     /**
999      * @param {Object} options
1000      * @param {Function} [onClose]
1001      */
1002     function createParser(options, onClose) {
1003       const parser = new Parser();
1004
1005       if (options) {
1006         if ('uri' in options) {
1007           parser.connect(options.uri);
1008         }
1009
1010         if ('onClose' in options) {
1011           parser.on('close', options.onClose);
1012         }
1013       }
1014
1015       return parser;
1016     }
1017
1018     module.exports = createParser;
1019   })();
1020
1021   /**
1022   (function() {
1023     'use strict';
1024
1025     /**
1026      * @param {Object} options
1027      * @param {Function} [onClose]
1028      */
1029     function createParser(options, onClose) {
1030       const parser = new Parser();
1031
1032       if (options) {
1033         if ('uri' in options) {
1034           parser.connect(options.uri);
1035         }
1036
1037         if ('onClose' in options) {
1038           parser.on('close', options.onClose);
1039         }
1040       }
1041
1042       return parser;
1043     }
1044
1045     module.exports = createParser;
1046   })();
1047
1048   /**
1049   (function() {
1050     'use strict';
1051
1052     /**
1053      * @param {Object} options
1054      * @param {Function} [onClose]
1055      */
1056     function createParser(options, onClose) {
1057       const parser = new Parser();
1058
1059       if (options) {
1060         if ('uri' in options) {
1061           parser.connect(options.uri);
1062         }
1063
1064         if ('onClose' in options) {
1065           parser.on('close', options.onClose);
1066         }
1067       }
1068
1069       return parser;
1070     }
1071
1072     module.exports = createParser;
1073   })();
1074
1075   /**
1076   (function() {
1077     'use strict';
1078
1079     /**
1080      * @param {Object} options
1081      * @param {Function} [onClose]
1082      */
1083     function createParser(options, onClose) {
1084       const parser = new Parser();
1085
1086       if (options) {
1087         if ('uri' in options) {
1088           parser.connect(options.uri);
1089         }
1090
1091         if ('onClose' in options) {
1092           parser.on('close', options.onClose);
1093         }
1094       }
1095
1096       return parser;
1097     }
1098
1099     module.exports = createParser;
1100   })();
1101
1102   /**
1103   (function() {
1104     'use strict';
1105
1106     /**
1107      * @param {Object} options
1108      * @param {Function} [onClose]
1109      */
1110     function createParser(options, onClose) {
1111       const parser = new Parser();
1112
1113       if (options) {
1114         if ('uri' in options) {
1115           parser.connect(options.uri);
1116         }
1117
1118         if ('onClose' in options) {
1119           parser.on('close', options.onClose);
1120         }
1121       }
1122
1123       return parser;
1124     }
1125
1126     module.exports = createParser;
1127   })();
1128
1129   /**
1130   (function() {
1131     'use strict';
1132
1133     /**
1134      * @param {Object} options
1135      * @param {Function} [onClose]
1136      */
1137     function createParser(options, onClose) {
1138       const parser = new Parser();
1139
1140       if (options) {
1141         if ('uri' in options) {
1142           parser.connect(options.uri);
1143         }
1144
1145         if ('onClose' in options) {
1146           parser.on('close', options.onClose);
1147         }
1148       }
1149
1150       return parser;
1151     }
1152
1153     module.exports = createParser;
1154   })();
1155
1156   /**
1157   (function() {
1158     'use strict';
1159
1160     /**
1161      * @param {Object} options
1162      * @param {Function} [onClose]
1163      */
1164     function createParser(options, onClose) {
1165       const parser = new Parser();
1166
1167       if (options) {
1168         if ('uri' in options) {
1169           parser.connect(options.uri);
1170         }
1171
1172         if ('onClose' in options) {
1173           parser.on('close', options.onClose);
1174         }
1175       }
1176
1177       return parser;
1178     }
1179
1180     module.exports = createParser;
1181   })();
1182
1183   /**
1184   (function() {
1185     'use strict';
1186
1187     /**
1188      * @param {Object} options
1189      * @param {Function} [onClose]
1190      */
1191     function createParser(options, onClose) {
1192       const parser = new Parser();
1193
1194       if (options) {
1195         if ('uri' in options) {
1196           parser.connect(options.uri);
1197         }
1198
1199         if ('onClose' in options) {
1200           parser.on('close', options.onClose);
1201         }
1202       }
1203
1204       return parser;
1205     }
1206
1207     module.exports = createParser;
1208   })();
1209
1210   /**
1211   (function() {
1212     'use strict';
1213
1214     /**
1215      * @param {Object} options
1216      * @param {Function} [onClose]
1217      */
1218     function createParser(options, onClose) {
1219       const parser = new Parser();
1220
1221       if (options) {
1222         if ('uri' in options) {
1223           parser.connect(options.uri);
1224         }
1225
1226         if ('onClose' in options) {
1227           parser.on('close', options.onClose);
1228         }
1229       }
1230
1231       return parser;
1232     }
1233
1234     module.exports = createParser;
1235   })();
1236
1237   /**
1238   (function() {
1239     'use strict';
1240
1241     /**
1242      * @param {Object} options
1243      * @param {Function} [onClose]
1244      */
1245     function createParser(options, onClose) {
1246       const parser = new Parser();
1247
1248       if (options) {
1249         if ('uri' in options) {
1250           parser.connect(options.uri);
1251         }
1252
1253         if ('onClose' in options) {
1254           parser.on('close', options.onClose);
1255         }
1256       }
1257
1258       return parser;
1259     }
1260
1261     module.exports = createParser;
1262   })();
1263
1264   /**
1265   (function() {
1266     'use strict';
1267
1268     /**
1269      * @param {Object} options
1270      * @param {Function} [onClose]
1271      */
1272     function createParser(options, onClose) {
1273       const parser = new Parser();
1274
1275       if (options) {
1276         if ('uri' in options) {
1277           parser.connect(options.uri);
1278         }
1279
1280         if ('onClose' in options) {
1281           parser.on('close', options.onClose);
1282         }
1283       }
1284
1285       return parser;
1286     }
1287
1288     module.exports = createParser;
1289   })();
1290
1291   /**
1292   (function() {
1293     'use strict';
1294
1295     /**
1296      * @param {Object} options
1297      * @param {Function} [onClose]
1298      */
1299     function createParser(options, onClose) {
1300       const parser = new Parser();
1301
1302       if (options) {
1303         if ('uri' in options) {
1304           parser.connect(options.uri);
1305         }
1306
1307         if ('onClose' in options) {
1308           parser.on('close', options.onClose);
1309         }
1310       }
1311
1312       return parser;
1313     }
1314
1315     module.exports = createParser;
1316   })();
1317
1318   /**
1319   (function() {
1320     'use strict';
1321
1322     /**
1323      * @param {Object} options
1324      * @param {Function} [onClose]
1325      */
1326     function createParser(options, onClose) {
1327       const parser = new Parser();
1328
1329       if (options) {
1330         if ('uri' in options) {
1331           parser.connect(options.uri);
1332         }
1333
1334         if ('onClose' in options) {
1335           parser.on('close', options.onClose);
1336         }
1337       }
1338
1339       return parser;
1340     }
1341
1342     module.exports = createParser;
1343   })();
1344
1345   /**
1346   (function() {
1347     'use strict';
1348
1349     /**
1350      * @param {Object} options
1351      * @param {Function} [onClose]
1352      */
1353     function createParser(options, onClose) {
1354       const parser = new Parser();
1355
1356       if (options) {
1357         if ('uri' in options) {
1358           parser.connect(options.uri);
1359         }
1360
1361         if ('onClose' in options) {
1362           parser.on('close', options.onClose);
1363         }
1364       }
1365
1366       return parser;
1367     }
1368
1369     module.exports = createParser;
1370   })();
1371
1372   /**
1373   (function() {
1374     'use strict';
1375
1376     /**
1377      * @param {Object} options
1378      * @param {Function} [onClose]
1379      */
1380     function createParser(options, onClose) {
1381       const parser = new Parser();
1382
1383       if (options) {
1384         if ('uri' in options) {
1385           parser.connect(options.uri);
1386         }
1387
1388         if ('onClose' in options) {
1389           parser.on('close', options.onClose);
1390         }
1391       }
1392
1393       return parser;
1394     }
1395
1396     module.exports = createParser;
1397   })();
1398
1399   /**
1400   (function() {
1401     'use strict';
1402
1403     /**
1404      * @param {Object} options
1405      * @param {Function} [onClose]
1406      */
1407     function createParser(options, onClose) {
1408       const parser = new Parser();
1409
1410       if (options) {
1411         if ('uri' in options) {
1412           parser.connect(options.uri);
1413         }
1414
1415         if ('onClose' in options) {
1416           parser.on('close', options.onClose);
1417         }
1418       }
1419
1420       return parser;
1421     }
1422
1423     module.exports = createParser;
1424   })();
1425
1426   /**
1427   (function() {
1428     'use strict';
1429
1430     /**
1431      * @param {Object} options
1432      * @param {Function} [onClose]
1433      */
1434     function createParser(options, onClose) {
1435       const parser = new Parser();
1436
1437       if (options) {
1438         if ('uri' in options) {
1439           parser.connect(options.uri);
1440         }
1441
1442         if ('onClose' in options) {
1443           parser.on('close', options.onClose);
1444         }
1445       }
1446
1447       return parser;
1448     }
1449
1450     module.exports = createParser;
1451   })();
1452
1453   /**
1454   (function() {
1455     'use strict';
1456
1457     /**
1458      * @param {Object} options
1459      * @param {Function} [onClose]
1460      */
1461     function createParser(options, onClose) {
1462       const parser = new Parser();
1463
1464       if (options) {
1465         if ('uri' in options) {
1466           parser.connect(options.uri);
1467         }
1468
1469         if ('onClose' in options) {
1470           parser.on('close', options.onClose);
1471         }
1472       }
1473
1474       return parser;
1475     }
1476
1477     module.exports = createParser;
1478   })();
1479
1480   /**
1481   (function() {
1482     'use strict';
1483
1484     /**
1485      * @param {Object} options
1486      * @param {Function} [onClose]
1487      */
1488     function createParser(options, onClose) {
1489       const parser = new Parser();
1490
1491       if (options) {
1492         if ('uri' in options) {
1493           parser.connect(options.uri);
1494         }
1495
1496         if ('onClose' in options) {
1497           parser.on('close', options.onClose);
1498         }
1499       }
1500
1501       return parser;
1502     }
1503
1504     module.exports = createParser;
1505   })();
1506
1507   /**
1508   (function() {
1509     'use strict';
1510
1511     /**
1512      * @param {Object} options
1513      * @param {Function} [onClose]
1514      */
1515     function createParser(options, onClose) {
1516       const parser = new Parser();
1517
1518       if (options) {
1519         if ('uri' in options) {
1520           parser.connect(options.uri);
1521         }
1522
1523         if ('onClose' in options) {
1524           parser.on('close', options.onClose);
1525         }
1526       }
1527
1528       return parser;
1529     }
1530
1531     module.exports = createParser;
1532   })();
1533
1534   /**
1535   (function() {
1536     'use strict';
1537
1538     /**
1539      * @param {Object} options
1540      * @param {Function} [onClose]
1541      */
1542     function createParser(options, onClose) {
1543       const parser = new Parser();
1544
1545       if (options) {
1546         if ('uri' in options) {
1547           parser.connect(options.uri);
1548         }
1549
1550         if ('onClose' in options) {
1551           parser.on('close', options.onClose);
1552         }
1553       }
1554
1555       return parser;
1556     }
1557
1558     module.exports = createParser;
1559   })();
1560
1561   /**
1562   (function() {
1563     'use strict';
1564
1565     /**
1566      * @param {Object} options
1567      * @param {Function} [onClose]
1568      */
1569     function createParser(options, onClose) {
1570       const parser = new Parser();
1571
1572       if (options) {
1573         if ('uri' in options) {
1574           parser.connect(options.uri);
1575         }
1576
1577         if ('onClose' in options) {
1578           parser.on('close', options.onClose);
1579         }
1580       }
1581
1582       return parser;
1583     }
1584
1585     module.exports = createParser;
1586   })();
1587
1588   /**
1589   (function() {
1590     'use strict';
1591
1592     /**
1593      * @param {Object} options
1594      * @param {Function} [onClose]
1595      */
1596     function createParser(options, onClose) {
1597       const parser = new Parser();
1598
1599       if (options) {
1600         if ('uri' in options) {
1601           parser.connect(options.uri);
1602         }
1603
1604         if ('onClose' in options) {
1605           parser.on('close', options.onClose);
1606         }
1607       }
1608
1609       return parser;
1610     }
1611
1612     module.exports = createParser;
1613   })();
1614
1615   /**
1616   (function() {
1617     'use strict';
1618
1619     /**
1620      * @param {Object} options
1621      * @param {Function} [onClose]
1622      */
1623     function createParser(options, onClose) {
1624       const parser = new Parser();
1625
1626       if (options) {
1627         if ('uri' in options) {
1628           parser.connect(options.uri);
1629         }
1630
1631         if ('onClose' in options) {
1632           parser.on('close', options.onClose);
1633         }
1634       }
1635
1636       return parser;
1637     }
1638
1639     module.exports = createParser;
1640   })();
1641
1642   /**
1643   (function() {
1644     'use strict';
1645
1646     /**
1647      * @param {Object} options
1648      * @param {Function} [onClose]
1649      */
1650     function createParser(options, onClose) {
1651       const parser = new Parser();
1652
1653       if (options) {
1654         if ('uri' in options) {
1655           parser.connect(options.uri);
1656         }
1657
1658         if ('onClose' in options) {
1659           parser.on('close', options.onClose);
1660         }
1661       }
1662
1663       return parser;
1664     }
1665
1666     module.exports = createParser;
1667   })();
1668
1669   /**
1670   (function() {
1671     'use strict';
1672
1673     /**
1674      * @param {Object} options
1675      * @param {Function} [onClose]
1676      */
1677     function createParser(options, onClose) {
1678       const parser = new Parser();
1679
1680       if (options
```

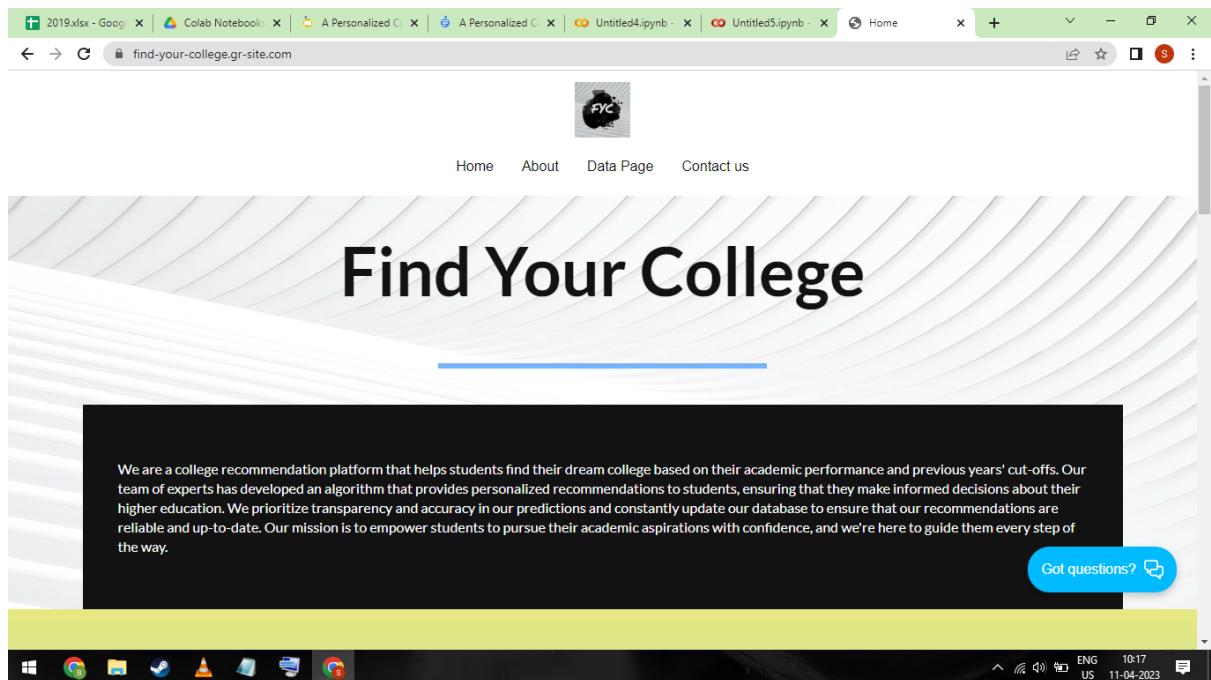
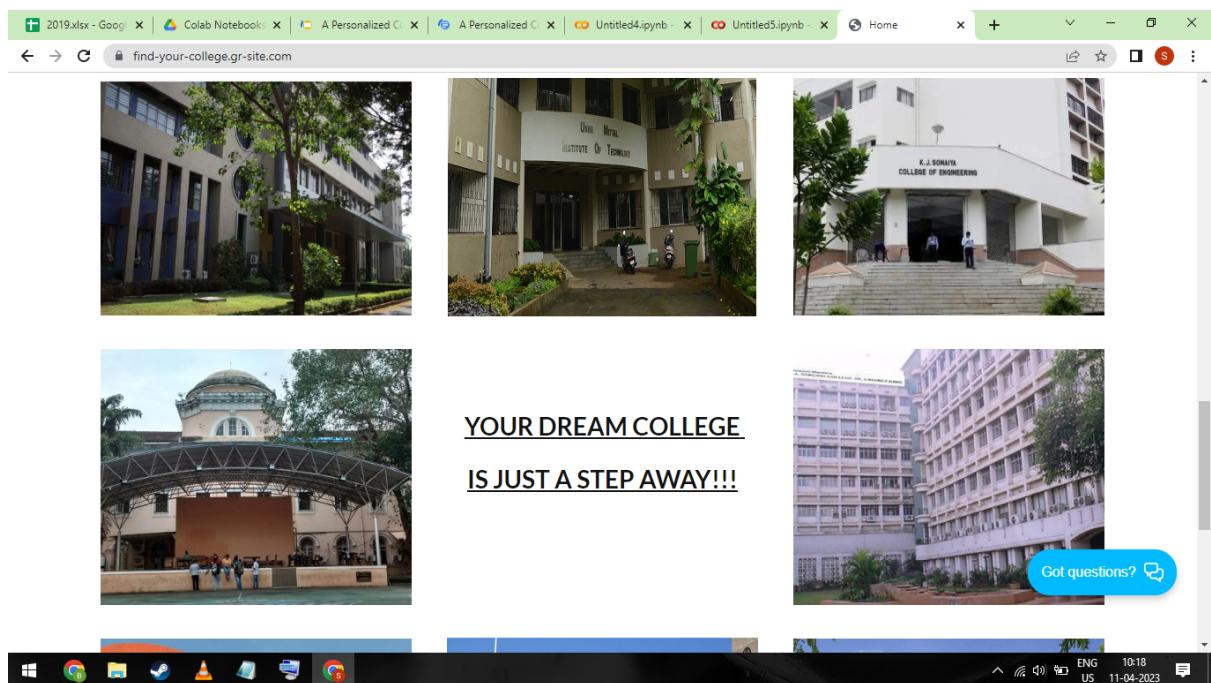
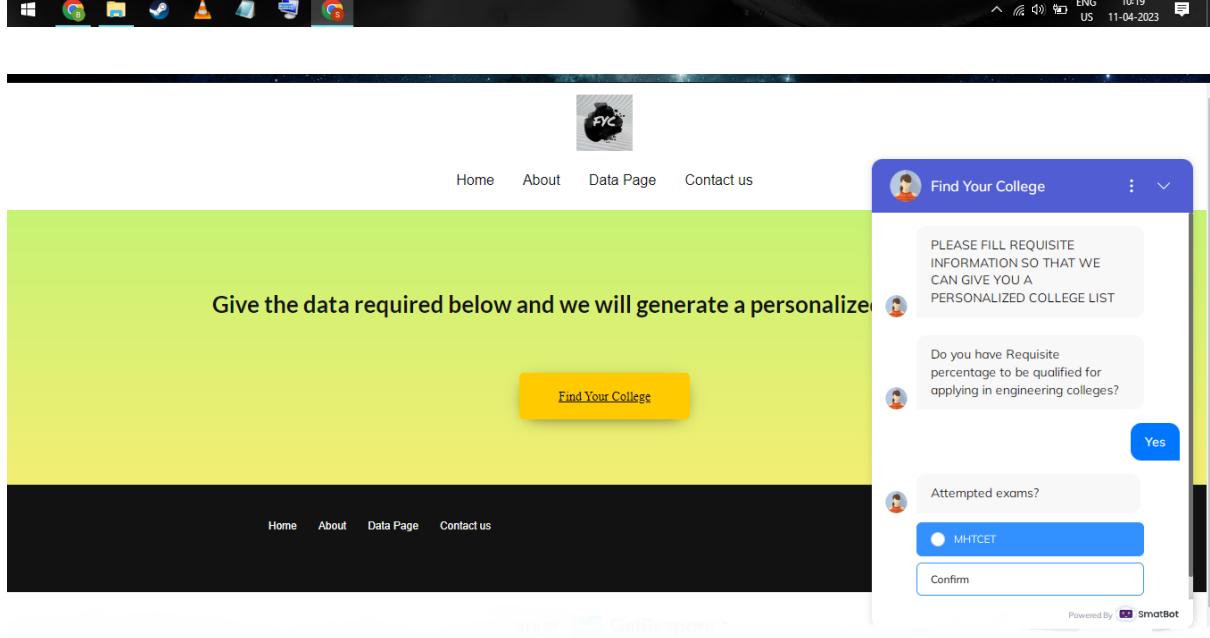
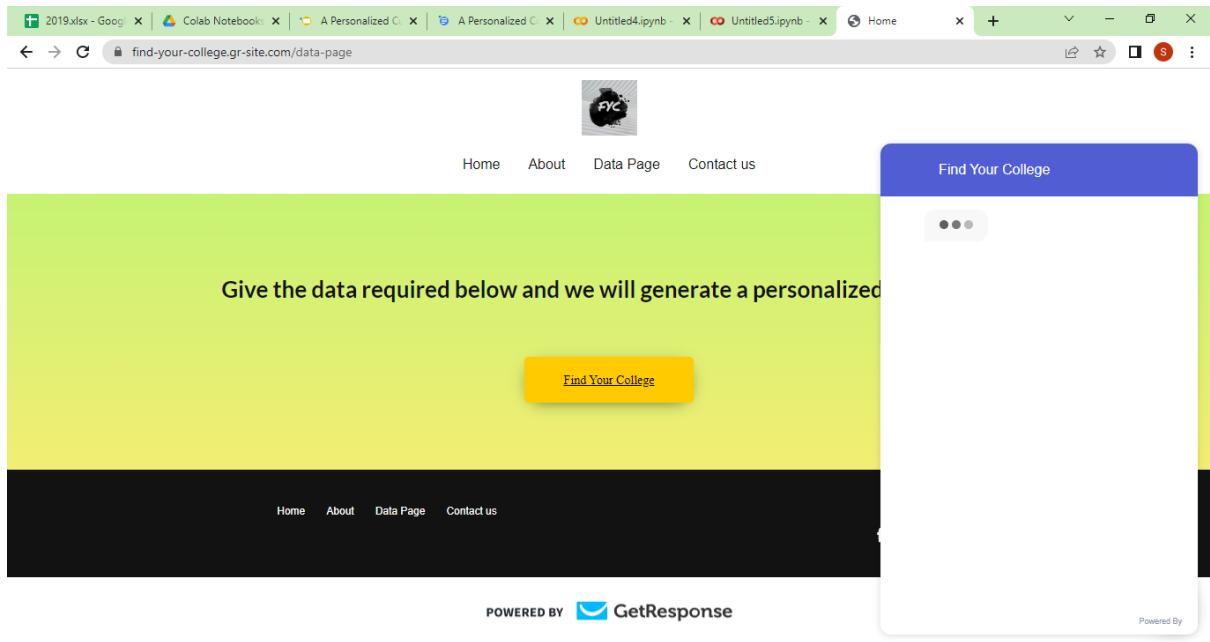


Figure 14: Front-end





MHT CET

56.67932

CHOICE OF UNIVERSITY

- MUMBAI UNIVERSITY
- SAVITRIBAI PHULE PUNE UNIVERSITY
- Dr. Babasaheb Ambedkar Technological University
- Sant Gadge Baba Amravati University
- Dr. Babasaheb Ambedkar Marathwada University
- SNDT Women's University
- Gondwana University

Powered by SmatBot

14:19
IN 02-05-2023

MUMBAI UNIVERSITY,SAVITRIBAI PHULE PUNE UNIVERSITY

OPEN

user's reservation

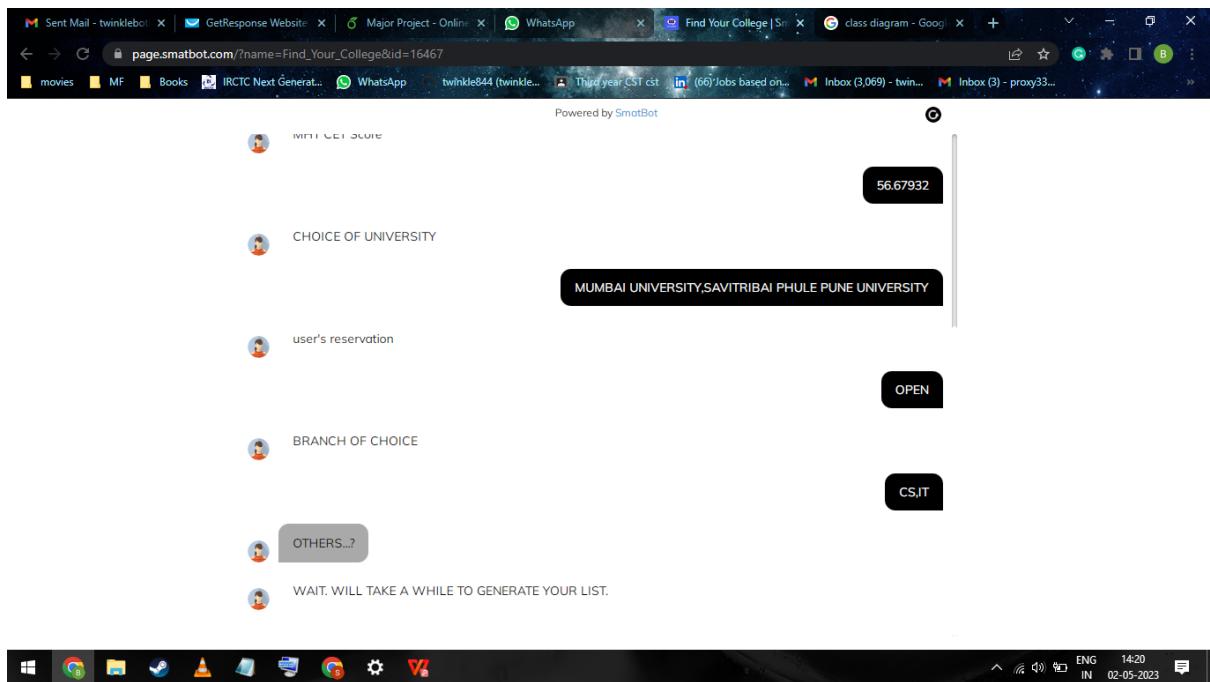
BRANCH OF CHOICE

- CS
- IT
- CE
- ME
- EXTC
- OTHERS

Confirm

Powered by SmatBot

14:20
IN 02-05-2023



References

- [1] Prof. Vishal R. Shinde, Miss. Anagha Bagul, Mr. Amit Gupta, Miss. Sneha Javeri, "Chatbot for college related FAQs", International Journal for Research in Engineering Application & Management (IJREAM), Special Issue - iCreate - 2019
- [2] Naveen Kumar M, Linga Chandar P C, Venkatesh Prasad A, Sumangali K, "Android Based Educational Chatbot for Visually Impaired People" , IEEE Dec 2016, International Conference on Computational Intelligence and Computing Research (ICCIC)
- [3] B. R. Ranoliya, N. Raghuwanshi, and S. Singh, "Chabot for university-related FAQs," 2017 International Conference on Advances in Computing, Communications, and Informatics (ICACCI), Udupi, 2017, pp. 1525-1530.
- [4] P. R. Telang, A. K. Kali, M. Vukovic, R. Pandita and M. P. Singh, "A Conceptual Framework for Engineering Chatbots," in IEEE Internet Computing, vol. 22, no. 6, pp. 54-59, 1 Nov. -Dec. 2018.
- [5] Emil Babu, Geethu Wilson, Msc Scholar, Assistant Professor, " Chatbot For College Enquiry", International Journal of Creative Research Thoughts (IJCRT), Vol-9, Issue 3, March 2021
- [6] Dr. Arunakumari B.N, Vishnu Sastry H.K, Sheetal Neeraj, Shashidhar R, "An Automated Prediction Model For College Admission System" , Elementary Education Online, 2021; Vol 20 (Issue 6): pp. 1172-1180
- [7] Varsha Powar, Sheetal Girase, Debajyoti Mukhopadhyay, Anuja Jadhav, Shweta Khude, Shital Mandlik, "Analyzing recommendation of colleges for students using data mining techniques", IEEE Xplore: 2018

[8] Zhenrong Zhang, "Design of a Personalized Recommendation System for Ideological and Political Courses in Colleges and Universities Based on a Collaborative Filtering Algorithm" IEEE Xplore: 2022

[9] Abdul Majeed Inamdar; Tanmay Mhatre; Pravin Nadar; Supriya Joshi, "Personalized College Recommender and Cutoff Predictor for Direct Second Year Engineering", IEEE Xplore: 2022