

ALUMNI NETWORK AND CAREER GUIDANCE

A PROJECT REPORT

Submitted by

MOHAMMED ABDUL RAHMAN KALWATHI (412517104073)
VIJAI S (412517104151)

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

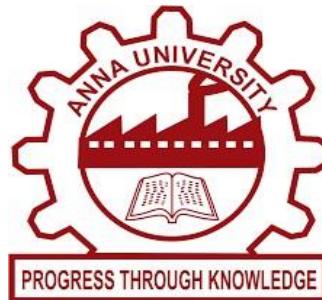
COMPUTER SCIENCE AND ENGINEERING



**SRI SAIRAM ENGINEERING COLLEGE(AUTONOMOUS),
SAI LEO NAGAR, CHENNAI-44**

ANNA UNIVERSITY: CHENNAI 600 025

MAY 2021



ANNA UNIVERSITY: CHENNAI 600025

BONAFIED CERTIFICATE

Certified that this project report "**ALUMNI NETWORK AND CAREER GUIDANCE**" is the bonafide work of "**J. MOHAMMED ABDUL RAHMAN KALWATHI AND VIJAI S**" who carried out the project work under my supervision.



SIGNATURE

Dr. B. Latha
HEAD OF DEPARTMENT,

Department of Computer
Science and Engineering
Sri Sai Ram Engineering College
(Autonomous) Chennai-600044



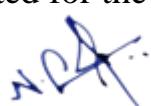
SIGNATURE

Ms. Sangeerani Devi. A
SUPERVISOR,

Department of Computer
Science and Engineering
Sri Sai Ram Engineering college
(Autonomous) Chennai-600 044

Submitted for the **VIVA-VOICE** Examination held on **06-05-2021**

(IN)



NER

(EXTERNAL EXAMINER)

S. S. Raji

ACKNOWLEDGEMENT

We would like to express our profound gratitude to our beloved and respected founder, (Late) **Thiru.MJF. Ln. LEO MUTHU**, Sri Sairam Engineering College for having provided plenty of resources.

We express our gratitude to our CEO **Mr. J. SAI PRAKASH** for his constant encouragement in completing this project.

We extend solemn thanks to our Principal **Dr.A. RAJENDRA PRASAD** for having given us his whole-hearted encouragement for completing this project.

We are indebted to our Head of the Department **Dr.B. LATHA**, for her support during the entire course of this project.

Our sincere thanks to our Project coordinator **Mr. N. GOPINATH**, Assistant professor for his kind support in bringing out this project successfully.

Our sincere thanks to our Supervisor **Mrs. A. SANDEEPANI DEVI**, Assistant Professor, for helping us in successful completion of this project.

Finally, we would like to thank all the staff members of the Department of Computer Science and Engineering, our parents, friends and all others who contributed directly and indirectly for the successful completion of our project.

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO
	LIST OF TABLE	v
	LIST OF FIGURES	viii
	ABSTRACT	ix
1	INTRODUCTION	10
	1.1 Overview	10
	1.1.1 Proposed Systems	10
	1.2 Domain Overview	
	1.2.1 Cloud computing	09
	1.2.2 Application Development	11
	1.3 Objective	12
2.	LITERATURE SURVEY	13
	2.1 Literature Survey	13
3.	SYSTEM ANALYSIS	15
	3.1 EXISTING SYSTEM	15
	3.1.1 Implementation of the Existing System	15
	3.2 PROPOSED SYSTEM	15
	3.2.1 Advantages of the System	16

4	REQUIREMENT SPECIFICATION	17
	4.1 Introduction	17
	4.2 Hardware Requirements	17
	4.3 Software Requirements	17
	4.3.1 Software Requirements Description	18
	4.3.1.1 Front End Technology	18
	4.3.1.2 Back End Technology	20
5	SYSTEM DESIGN	23
	5.1 Introduction	23
	5.2 Flow/Architectural Diagram	23
	5.3 Data Flow Diagram	24
	5.4 UML Diagrams	24
	5.4.1 Use Case Diagrams	25
	5.4.2 Class Diagrams	26
6	SYSTEM IMPLEMENTATION	27
	6.1 List of Modules	27
	6.2 Module Implementation	27
	6.2.1 Front End Development	27
	6.2.2 Creating Backend API	28
	6.2.3 Extracting data and integrate with front End	29
7	Output of the System	31
8	Testing	44
	8.1 Introduction	44
	8.2 Testing Strategies	44
	8.2.1 Whitebox Testing	45
	8.2.2 Blackbox Testing	45
	8.2.3 Unit Testing	45
	8.2.4 Functional Testing	45
	8.2.5 Integration Testing	46
	8.2.6 Validation Testing	46
	8.2.7 System Testing	46
	8.2.8 Structure Testing	46
9	CONCLUSION-FURTHER SCOPE	47
	APPENDIX-1	48
	APPENDIX-2	59
	APPENDIX-3	63
	APPENDIX-4	64
	REFERENCES	66

ABBREVIATION

MNC	Multinational Corporation
WAN	Wide Area Network
LAN	Local Area Network
VPN	Virtual Private Network
AWS	Amazon Web Services
GCP	Google Cloud Platform
API	Application Program Interface
GRE	Graduate Record Examinations
TOEFL	Test of English As a Foreign Language
IELTS	International English Language Test
CRM	Customer Relationship Management
IDE	Integrated Development Environment
GUI	Graphical User Interface
API	Application Program Interface
DOM	Document Object Model
JSX	Javascript XML
HTTP	Hypertext Transfer Protocol
REST	Representational State Transfer
JSON	JavaScript Object Notation

LIST OF FIGURES

- Figure 1 Flow Diagram of the project
- Figure 2 Dataflow Diagram of the project
- Figure 3 Use Case Diagram of the project
- Figure 4 Class diagram of the project
- Figure 5 Mongo DB platform
- Figure 6 Waiting for Database connection
- Figure 7 Database connection successful
- Figure 8 Landing Page
- Figure 9 Login Page
- Figure 10 Registration page
- Figure 11 Signup page for Alumni
- Figure 12 Signup page for Alumni with data
- Figure 13 Signup page for Student
- Figure 14 Signup page for Student with data
- Figure 15 Signup page for Staff
- Figure 16 Signup page for Staff with data
- Figure 17 Alumni data in JSON format in database
- Figure 18 Student data in JSON format in database
- Figure 19 Staff data in JSON format in database
- Figure 20 Higher studies and placement query page (After Student login)
- Figure 21 Placement and Company review page
- Figure 22 Higher Studies Review page
- Figure 23 MS in computer science Query page
- Figure 24 MBA Query page
- Figure 25 MTech in computer science Query page
- Figure 26 MS in computer Science Practice question paper and practice test
- Figure 27 MBA Practice question paper and practice test
- Figure 28 MTech in computer Science Practice question paper and practice test
- Figure 29 Placement review Page (After Alumni login)
- Figure 30 Rating for the company
- Figure 31 Getting feedback for the company about placement
- Figure 32 Thank you message
- Figure 33 Meet Up Page

ABSTRACT

Building a good professional network is the basis for anyone desiring to be successful in their career. The worth of any institution is measured by evaluating the success of students passed out from the institution. Communication between graduates and the institution can be very beneficial in many situations. Nowadays, there is no proper communication between the alumni and the students of the institution due to a lack of understanding and exposure. Students show less awareness of the technical world. They do lack the expectations of the MNC. Generally, students have a biased opinion in getting a job versus cracking a master degree. So they need a common platform to communicate.

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

The scope of the project is to create a platform between students and alumni of the institution which is helpful for the student for their career and also for their higher studies. Student can also come to know about what some MNC is expecting from them.

1.1.1 Proposed Systems

- Cloud computing (MONGODB ATLAS)
- Application Development

1.2 DOMAIN OVERVIEW

1.2.1. Cloud computing (MONGODB ATLAS):

The term Cloud refers to a **Network** or **Internet**. In other words, we can say that Cloud is something, which is present at remote location. Cloud can provide services over public and private networks, i.e., WAN, LAN or VPN. Applications such as e-mail, web conferencing, customer relationship management (CRM) execute on cloud.

The core of MongoDB Cloud is MongoDB Atlas, a fully managed cloud database for modern applications. Atlas is the best way to run MongoDB, the leading modern database. MongoDB's document model is the fastest way to innovate, bringing flexibility and ease of use to the database. Atlas is available on 70+ regions across AWS, GCP, and Azure. Best-in-class automation and proven practices guarantee availability, scalability, and compliance with the most demanding data security and privacy standards.

Atlas comes with MongoDB Atlas Search built in, making it easy to build fast, relevant, full-text search capabilities on top of your MongoDB data. No need to deploy a separate search platform: create search indexes directly in Atlas and use the MongoDB aggregation framework to build sophisticated queries.

1.2.2 Application Development:

Application development is the process of creating software applications that run on a mobile device or desktop, and a typical mobile application utilizes a network connection to work with remote computing resources. The programming and markup languages used for this kind of software development include HTML5, CSS, REACTJS, NODEJS and JAVASCRIPT. The application development in this project is carried out by Visual Studio.

Microsoft Visual Studio is an integrated development environment (IDE) from Microsoft. It is used to develop computer programs, as well as websites, web apps, web services and mobile apps. Visual Studio uses Microsoft software development platforms such as Windows API, Windows Forms, Windows Presentation Foundation, Windows Store and Microsoft Silverlight. It can produce both native code and managed code.

Visual Studio includes a code editor supporting IntelliSense as well as code refactoring. The integrated debugger works both as a source-level debugger and a machine-level debugger. Other built-in tools include a code profiler, designer for building GUI applications, web designer, class designer, and database schema designer. It accepts plug-ins that expand the functionality at almost every level—including adding support for source control systems and adding new toolsets like editors and visual designers for domain-specific languages or toolsets for other aspects of the software development lifecycle.

Visual Studio supports 36 different programming languages and allows the code editor and debugger to support (to varying degrees) nearly any programming language, provided a language-specific service exists. Built-in languages include C, C++, C++/CLI, Visual Basic .NET, JavaScript, TypeScript, XML, XSLT, HTML, and CSS. Support for other languages such as Python, Ruby, Node.js.

1.3 OBJECTIVE

The main objective of this project is to provide a handy product with application that helps students to connect the network between alumni and student of the institution Key objective maps to:

- students know the company hiring process by viewing the review given by alumni.
- For students who are doing the masters can know about test strategies, by viewing the review.
- open job opportunities, free courses are shared by staff.
- Easily communicate with alumni for future reference.

CHAPTER 2

LITERATURE SURVEY

2.1 LITERATURE SURVEY:

Paper [1]: PCRS: Personalized Career-Path Recommender System for Engineering Students

This paper is proposed by manar qamhieh, haya sammaneh, mona nabil demaidi in the year of 2020. This paper proposes a Personalized Career-path Recommender System (PCRS) which help high school students choose best candidate engineering discipline such as; Computer Engineering, Electrical Engineering and Civil Engineering. They also collecting student personal information, personality type and Building a fuzzy recommender system to provide students with personalized and user-specific ranking of engineering disciplines.

Advantage: Helps students choose their field based on their academic performance
Disadvantage: There is no involvement of human help for mentoring the students

Paper [2]: University Based Job Recommender & Alumni System

This paper is proposed by Akeem Olowolayemo, Kamaleiah Harun, Teddy Mantoro in the year 2018. The study proposed a university-based job recommender and alumni system that is intended to involve universities in aiding and supporting their alumni, especially fresh graduates in securing jobs. The system also provides the universities the strategies to maintain continuous bond with their alumni and constantly able to track their job progressions and achievements.

Advantage: Help university in gathering graduate's employability and helps graduates in finding job.

Disadvantage: No direct communication between students and Alumni.

Paper [3]: Customized Alumni Portal implementing Natural Language Processing Techniques

This paper is proposed by Ashmy Achu Shinu, Keerthi V, Keerthi V, Muhammed Shaheer, Namitha S Nair, Ansamma John in the year of 2019. This paper has discussed the implementation of an alumni portal with GDPR compliance incorporating Natural Language Processing. The key modules event and calendar module, opportunities module, mentoring and collaboration module and a discussion forum module, are built onto which Natural Language Processing (NLP) features are added.

Advantage Helps students choose their field based on their academic performance.

Disadvantage They are only mentoring the students

Paper [4]: OPTIMIZED ALUMNI DATA MANAGEMENT SYSTEM

This paper is proposed by Amirthavalli P, Bhuvaneshwari H, Divya G, Sorna Shanthi D in the year of 2017. This paper has discussed the receiving user information and enabling easy communication between the alumni and the institution, it provides more benefits to the administrator in fetching the user data than the existing applications. It mainly focuses on managing time efficiently and providing better service to the users, thereby managing time effectively and efficiently.

Advantage: Does not require maintenance effort from the alumni side.

Disadvantage: Less efficiency as latest technologies are not used.

CHAPTER 3

SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

The already existing system finds a solution by providing guidance and helps students choose engineering discipline and guide the students for career and also provides an optimized solution for collecting and managing the alumni data.

3.1.1 DISADVANTAGES OF EXISTING SYSTEM

This existing system has some drawbacks that has to be concentrated and could be used for reference of a new system.

- There is no involvement of human help for mentoring the students
- Less efficiency as latest technologies are not used
- No direct communication between students and Alumni.
- They are only mentoring the students

3.2 PROPOSED SYSTEM

The idea of the proposed system is to design a platform that connects alumni and students. It is a website-based application, initially, all the students and alumni have to give their information during the registration process. After that students and alumni information is kept in a separate profile. And, then alumni have to share their placement experience. Some of the alumni were going for higher studies, they want to share their (GRE, TOFEL, IELTS) examination experience. The feedback given by each alumnus will be displayed in the portal. Students can view feedback and they can ready for the placement or higher study preparation. Faculty can also create the account, the purpose of the faculty is to share information like a free course, job opportunity, placement related activity

3.2.1 ADVANTAGES OF THE PROPOSED SYSTEM

The advantages of the proposed system are:

- Students can know the cracking strategies for the company or exam by viewing the feedback given by alumni.
- Students can easily communicate with the alumni of the college
- Faculty can also share the information regarding the free course and placement.
- If any Alumni start their start-up, they can easily hire an employee from their college.
- Alumni information is stored in the cloud. So, we can retrieve their information at any time we want.

CHAPTER 4

REQUIREMENT SPECIFICATIONS

4.1 INTRODUCTION:

A Requirement Specification is a collection of the set of all requirements that are to be imposed on the design and verification of the product. The specification also contains other related information necessary for the design, verification, and maintenance of the product. The requirement specification of the project Software requirements irrespective of both functional and non-functional requirements.

4.2 HARDWARE REQUIREMENTS:

The hardware requirements may serve as the basis for a contract for the implementation of the system and should therefore be a complete and consistent specification of the whole system. They are used by software engineers as the starting point for the system design. It shows what the systems do and not how it should be implemented.

- RAM 8GB
- Dual-Core 2.8 GHz Processor and Above
- HDD 80 GB Hard Disk Space and Above

4.3 SOFTWARE REQUIREMENTS:

- HTML
- CSS
- BOOTSTRAP
- REACT JS
- NODE JS
- MONGODB ATLAS(CLOUD)
- VISUAL STUDIO

4.3.1. SOFTWARE REQUIREMENTS DESCRIPTION:

4.3.1.1. FRONT END TECHNOLOGY:

A. HTML:

HTML (Hyper Text Markup Language) is the most basic building block of the Web. It defines the meaning and structure of web content. Other technologies besides HTML are generally used to describe a web page's appearance/presentation (CSS) or functionality (JavaScript)."Hypertext" refers to links that connect web pages to one another, either within a single website or between websites. Links are a fundamental aspect of the Web. By uploading content to the Internet and linking it to pages created by other people, you become an active participant in the World Wide Web.

B. CSS:

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language such as HTML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript.

CSS is designed to enable the separation of presentation and content, including layout, colours, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple web pages to share formatting by specifying the relevant CSS in a separate .css file which reduces complexity and repetition in the structural content as well as enabling the .css file to be cached to improve the page load speed between the pages that share the file and its formatting.

C. BOOTSTRAP:

Bootstrap is a free and open-source CSS framework directed at responsive, mobile-first front-end web development. It contains CSS- and JavaScript-based design templates for typography, forms, buttons, navigation, and other interface components. Here are some additional reasons to use Bootstrap:

- Bootstrap's responsive CSS adjusts to phones, tablets, and desktops
- Mobile-first styles are part of the framework
- Bootstrap is compatible with all modern browsers (Chrome, Firefox, Internet Explorer, Safari, and Opera)

D. REACTJS:

React (also known as **React.js** or **ReactJS**) is an open-source, front end, JavaScript library for building user interfaces or UI components. It is maintained by Facebook and a community of individual developers and companies. React can be used as a base in the development of single-page or mobile applications. However, react is only concerned with state management and rendering that state to the DOM, so creating React applications usually requires the use of additional libraries for routing, as well as certain client-side functionality.¹

Advantages:

- Uses virtual DOM which is a JavaScript object. This will improve apps performance, since JavaScript virtual DOM is faster than the regular DOM.
- Can be used on client and server side as well as with other frameworks.
- Component and data patterns improve readability, which helps to maintain larger apps.

React Features:

1. JSX is JavaScript syntax extension. It isn't necessary use JSX in React development, but it is recommended.

2. React is all about components. You need to think of everything as a component. This will help you maintain the code when working on larger scale projects.

3. React implements one-way data flow which makes it easy to reason about your app. Flux is a pattern that helps keeping your data unidirectional

4.3.1.2 BACK END TECHONLOGY:

A. NODEJS:

Node.js is an open-source, cross-platform, back-end JavaScript runtime environment that runs on the V8 engine and executes JavaScript code outside a web browser. Node.js lets developers use JavaScript to write command line tools and for server-side scripting—running scripts server-side to produce dynamic web page content before the page is sent to the user's web browser. Consequently, Node.js represents a "JavaScript everywhere" paradigm, unifying web-application development around a single programming language, rather than different languages for server-side and client-side scripts. Node.js also provides a rich library of various JavaScript modules which simplifies the development of web applications using Node.js to a great extent.

FEATURES OF NODE JS:

1. All APIs of Node.js library are asynchronous, that is, non-blocking. It essentially means a Node.js based server never waits for an API to return data. The server moves to the next API after calling it and a notification mechanism of Events of Node.js helps the server to get a response from the previous API call.
2. Being built on Google Chrome's V8 JavaScript Engine, Node.js library is very fast in code execution.
3. Node.js uses a single threaded model with event looping. Event mechanism helps the server to respond in a non-blocking way and makes the server highly scalable as opposed to traditional servers which create limited threads to handle requests. Node.js uses a single threaded program and the same program can provide service to a much larger number of requests than traditional servers like Apache HTTP Server.
4. Node.js applications never buffer any data. These applications simply output the data in chunks

B. MONGODB ATLAS:

MongoDB is a document-oriented database. It is a key feature of MongoDB. It offers a document-oriented storage. It is very simple you can program it easily. MongoDB stores data as documents, so it is known as document-oriented database. For example:

1. FirstName = "JOHN"
2. Address = "Chennai"
3. Spouse = [{NAME:"
BEN"}].
4. FirstName ="JOHN"

Storing data in this manner is called as document-oriented database. Mongo DB falls into a class of databases that calls Document Oriented Databases. There is also a broad category of database known as No SQL Databases.

FEATURES OF MONGODB:

- In MongoDB, you can search by field, range query and it also supports regular expression searches.
- You can index any field in a document
- MongoDB supports Master Slave replication.
- MongoDB can run over multiple servers. The data is duplicated to keep the system up and also keep its running condition in case of hardware failure.
- Provides high performance.
- Easy to administer in the case of failures
- It has an automatic load balancing configuration because of data placed in shards.

MONGODB ATLAS:

The core of MongoDB Cloud is MongoDB Atlas, a fully managed cloud database for modern applications. Atlas is the best way to run MongoDB, the leading modern database. MongoDB's document model is the fastest way to innovate, bringing flexibility and ease of use to the database. Atlas is available on 70+ regions across AWS, GCP, and Azure. Best-in-class automation and proven practices guarantee availability, scalability, and compliance with the most demanding data security and privacy standards.

Atlas comes with MongoDB Atlas Search built in, making it easy to build fast, relevant, full-text search capabilities on top of your MongoDB data. No need to deploy a separate search platform: create search indexes directly in Atlas and use the MongoDB aggregation framework to build sophisticated queries.

VISUAL STUDIO CODE:

Microsoft Visual Studio is an integrated development environment (IDE) from Microsoft. It is used to develop computer programs, as well as websites, web apps, web services and mobile apps. Visual Studio uses Microsoft software development platforms such as Windows API, Windows Forms, Windows Presentation Foundation, Windows Store and Microsoft Silverlight. It can produce both native code and managed code.

Visual Studio includes a code editor supporting IntelliSense as well as code refactoring. The integrated debugger works both as a source-level debugger and a machine-level debugger. Other built-in tools include a code profiler, designer for building GUI applications, web designer, class designer, and database schema designer. It accepts plug-ins that expand the functionality at almost every level—including adding support for source control systems and adding new toolsets like editors and visual designers for domain-specific languages or toolsets for other aspects of the software development lifecycle.

Visual Studio supports 36 different programming languages and allows the code editor and debugger to support (to varying degrees) nearly any programming language, provided a language-specific service exists. Built-in languages include C, C++, C++/CLI, Visual Basic .NET, JavaScript, TypeScript, XML, XSLT, HTML, and CSS. Support for other languages such as Python, Ruby, Node.js.

CHAPTER 5

SYSTEM DESIGN

5.1 INTRODUCTION:

Systems design is the process of defining elements of a system like modules, architecture, components and their interfaces and data for a system based on the specified requirements. It is the process of defining, developing and designing systems which satisfies the specific needs and requirements of a business or organization.

5.2 FLOW/ ARCHITECTURE DIAGRAM:

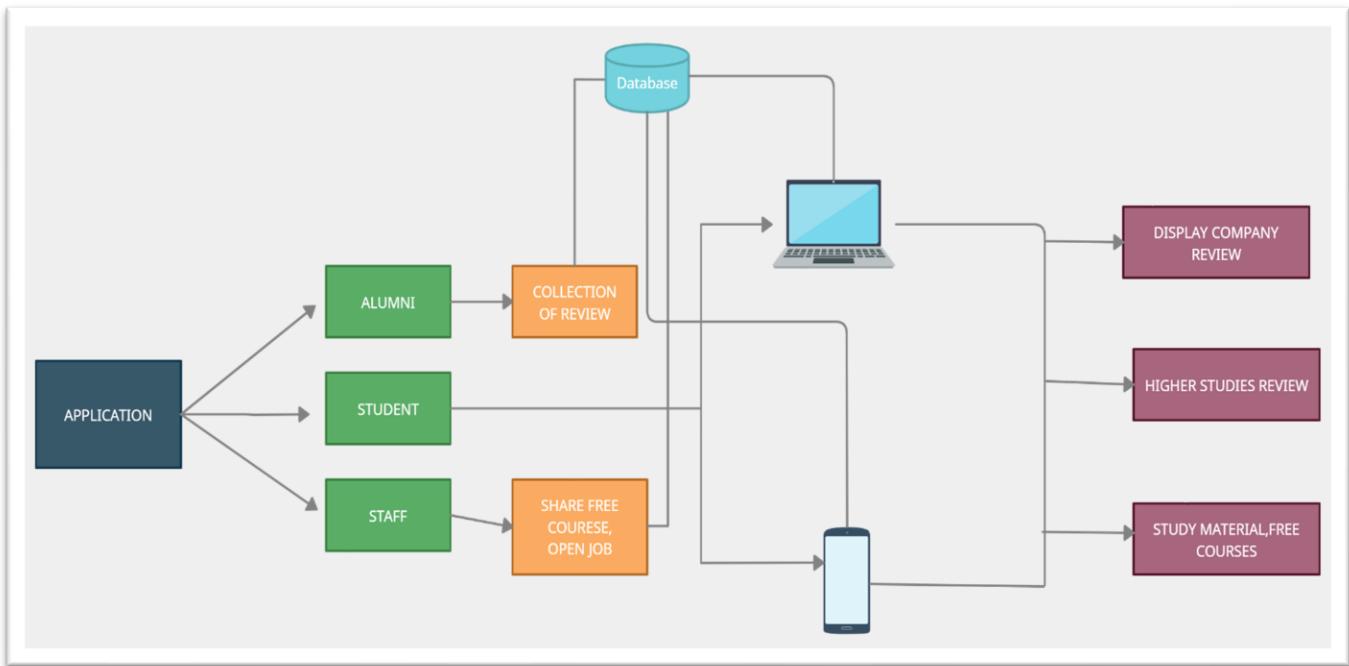


Figure 1: Flow Diagram

The above diagram states the overall architecture/ flow of the proposed project system. To explain briefly, the process goes like

- Initially, alumni, staff, and students have to create an account for them by giving their personal information.
- After that these data are stored in MongoDB atlas(database).
- And then, Alumni have to share their feedback about the company they have attended and the work experience.
- The staff has to share any open job opportunity and free available courses on any platform.
- finally, the information shared by Alumni and staff were displayed for students.

5.3. DATA FLOW DIAGRAM:

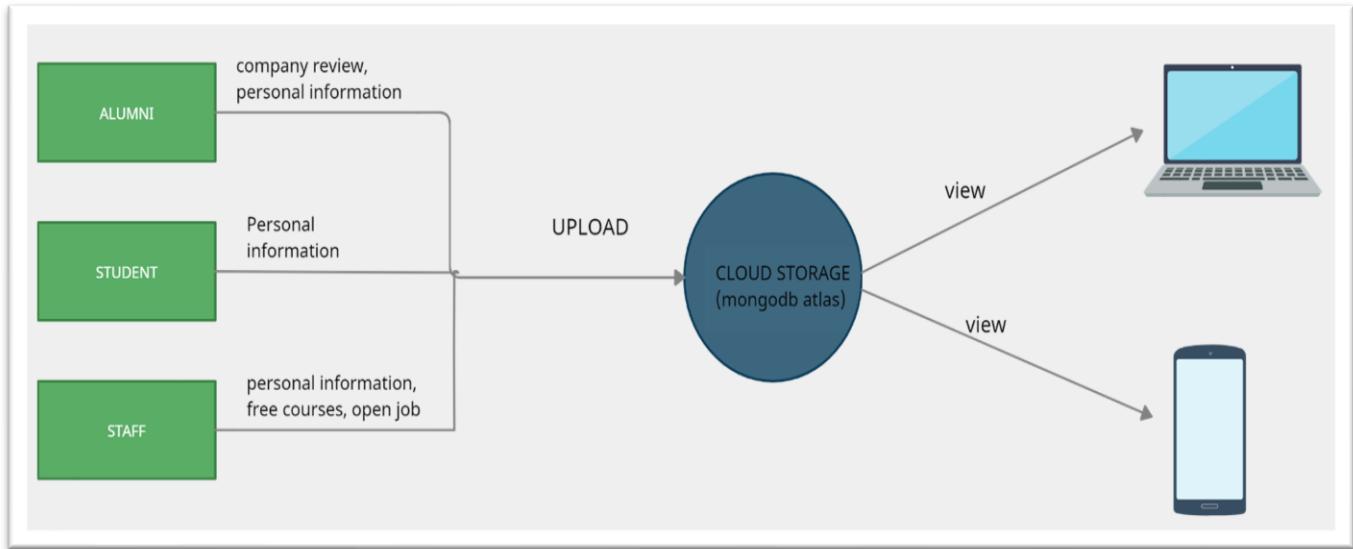


Figure 2: Dataflow Diagram

The above data flow diagram states the processing of collecting data from the Alumni, staff, Student like personal information, company review, free courses recommendations and open job. These data were collected from them and stored in the cloud for further analytics in the mobile application.

5.4 UML DIAGRAM:

UML is a standardized modelling language consisting of an integrated set of diagrams, developed to help system and software developers for specifying, visualizing, constructing, and documenting the artifacts of software systems, as well as for business modelling and other non-software systems. The UML represents a collection of best engineering practices that have proven successful in the modelling of large and complex systems. The UML is a very important part of developing object oriented software and the software development process. The UML uses mostly graphical notations to express the design of software project. The UML diagrams for the proposed project is given below.

5.4.1 USE CASE DIAGRAM:

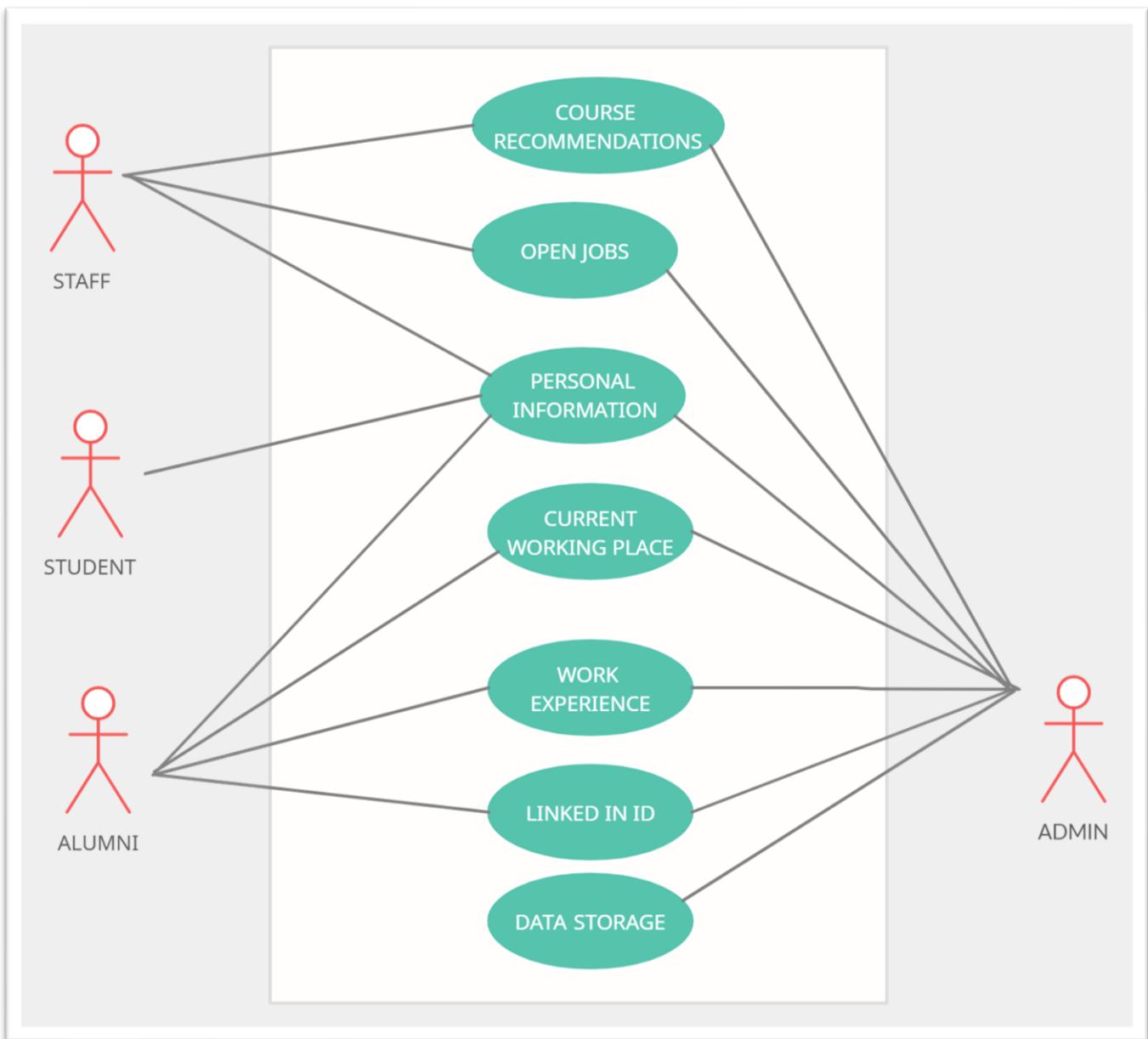


Figure: 3 UseCase Diagram

The use case diagrams represents the system functionality and requirements stated by the end users. In this project, the actors role is played by user. Both end user and mobile application can retrieve the other features shown.

5.4.2.CLASS DIAGRAM:

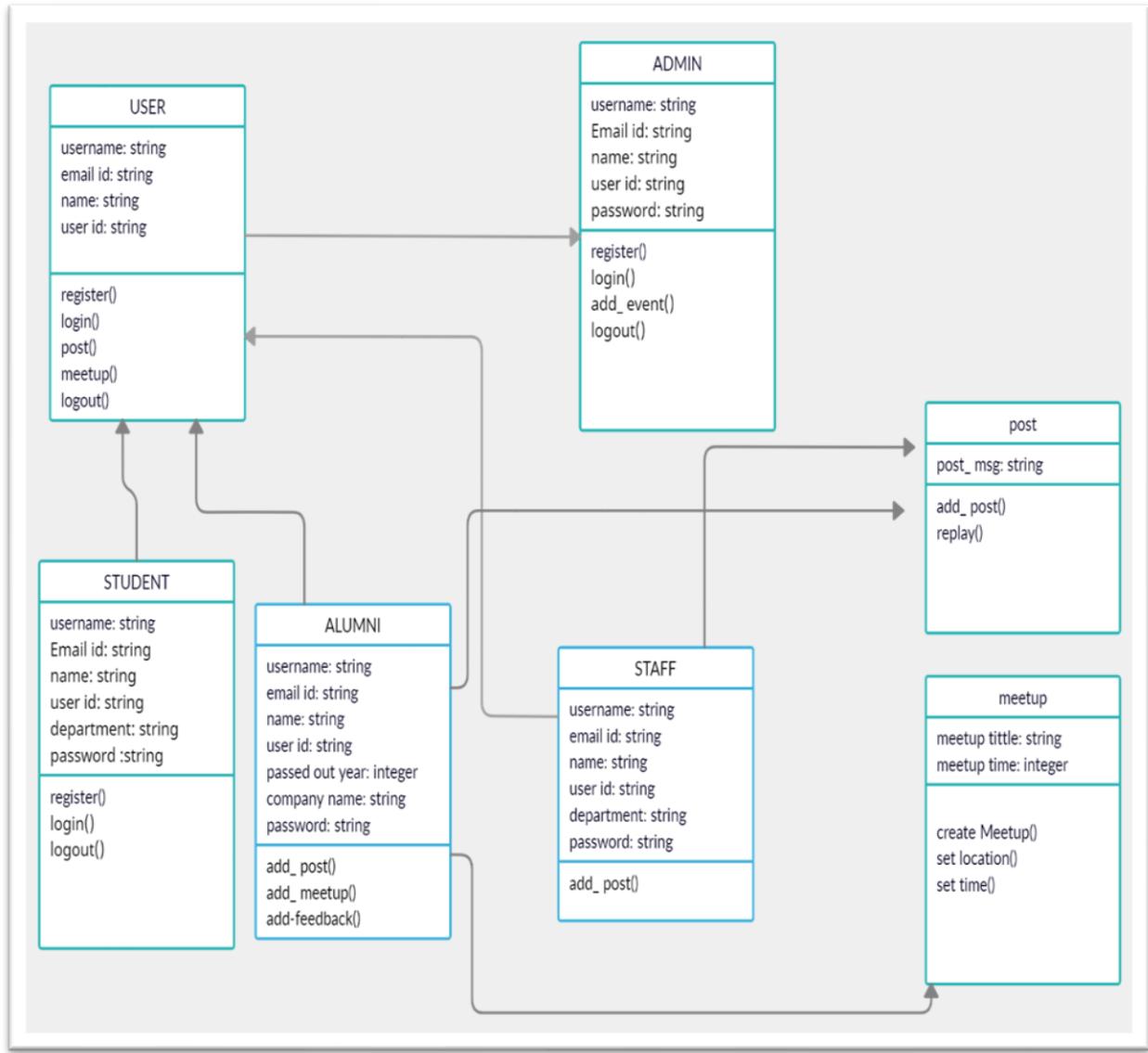


Figure 4: Class diagram

Generally, Class diagrams are used for general conceptual modeling of the structure of the application, and for detailed modeling translating the models into programming code. Class diagrams can also be used for data modeling.

CHAPTER 6

SYSTEM IMPLEMENTATION

The implementation of the project is classified in different modules for reducing the complexity of the project and increasing the efficiency of the outcomes of the project.

6.1 LIST OF MODULES

The project has been classified into three modules. The list of the modules are:

- Frontend development
- Creating backend API
- Extracting data and integrate with front end

6.2 MODULE IMPLEMENTATION

6.2.1 FRONTEND DEVELOPMENT:

The entire project highly depends upon the software. The software set consists of HTML, CSS, REACTJS, BOOTSTRAP.

Our basic idea of development is to build a platform where alumni and students of a particular institution or college can easily communicate. Through this student can easily choose their career option or higher studies by getting guidance from the alumni. Staff can also present on this platform.

Initially, we have developed the login page, where the user has to give their personal information like username, name, password, department, year of study. If he\she is an alumnus then they have to give additional information like company name, LinkedIn id, year of experience. And this is called a landing page created by HTML, CSS, BOOTSTRAP.

After giving the information a unique account is created for each user. By using their username and password they can log in to their account. If a student has logged in then he\she is redirected to the placement and higher studies query page. In addition to that, there is a separate link for meetups.

After clicking the placement query page list of the company is displayed, for example, TCS, CTS, Accenture, etc. If he\she is clicking the higher studies page then a list of exams for MTech, MS, MBA is displayed, for example, GRE, TOEFL, etc.

6.2.2 CREATING BACKEND API:

The backend Application Programming Interfaces (APIs) are created using the following stack

- Node js, which is a Runtime environment for executing Js code, Outside of Browser.
- Express, which is a minimal and flexible Node.js web application framework that simplifies building out common web server tasks.
- Mongoose, which provides a straightforward, schema-based solution to model the application data will connect our backend to a MongoDB database

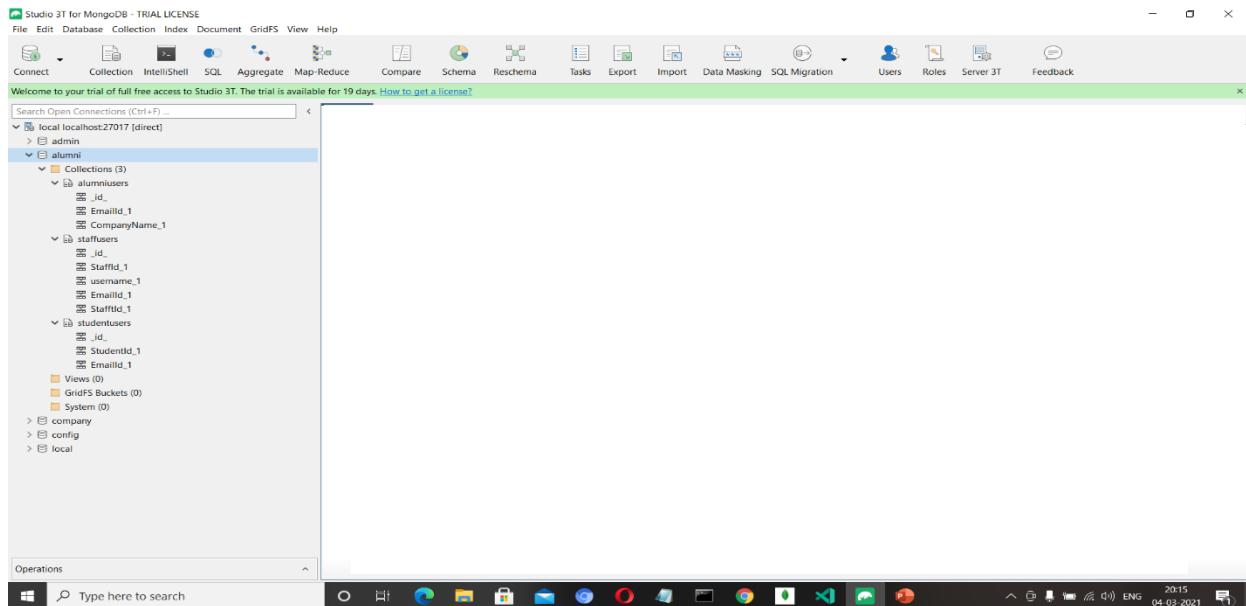


Figure5: Mongo DB platform

The CRUD operations such as create, read, update and delete functionality are implemented using REST APIs - Representational State Transfer. The data entered and retrieved by the users like Alumni, Students and Staff are handled by multiple types of calls, and return different data formats with the help of REST APIs. All these APIs are tested and developed with the help of an API client called Postman. The Signup and Login feature of the web page is created using Passport which is an authentication middleware of Node js.

6.2.3. EXTRACTING DATA AND INTEGRATE WITH FRONT END:

The data stored at the backend will be retrieved in JSON format. **JSON** stands for JavaScript Object Notation. It is a lightweight format for storing and transporting data and often used when data is sent from a server to a web page. It is an open standard file format and data interchange format that uses human-readable text to store and transmit data objects consisting of attribute-value pairs and arrays. It is a very common data format, with a diverse range of applications, one example being web applications that communicate with a server.

The data such as placement reviews, feedback entered by the alumni and details entered by the staff will be retrieved from the database by the concept of Fetch API which provides an interface for fetching resources. The `fetch()` method takes one mandatory argument, the path to the resource you want to fetch. The fetched data will be displayed by JavaScript logics performed with the help of templating engine called `ejs` which renders the webpage content at the front end.

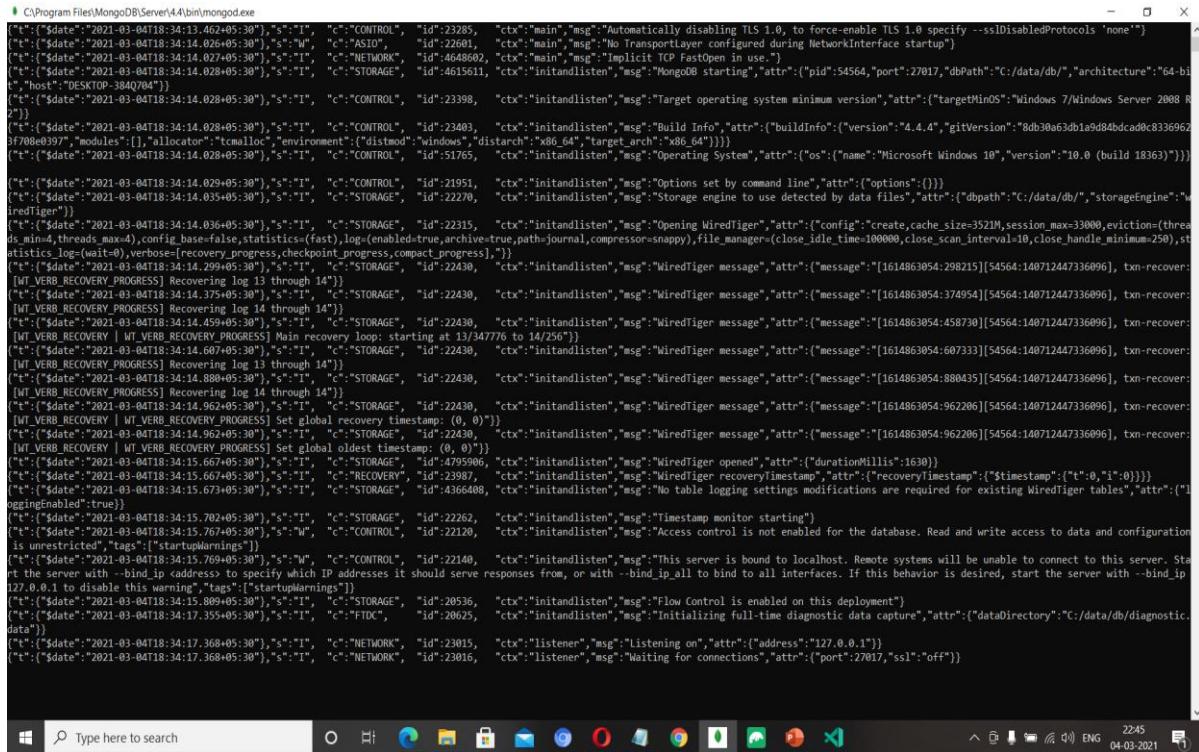


Figure 6: Waiting for Database connection

```
mongod
rt the server with --bind_ip <address> to specify which IP addresses it should serve responses from, or with --bind_ip_all to bind to all interfaces. If this behavior is desired, start the server with --bind_ip
127.0.0.1 to disable this warning", "tags": ["startupWarnings"]}
[{"t": "date", "v": "2021-03-14T22:31:14.513+05:30"}, "s": "1", "c": "STORAGE", "id": "20836", "ctx": "initandlisten", "msg": "Flow Control is enabled on this deployment"}
[{"t": "date", "v": "2021-03-14T22:31:14.878+05:30"}, "s": "1", "c": "FIDC", "id": "20625", "ctx": "initandlisten", "msg": "Initializing full-time diagnostic data capture", "attr": {"dataDirectory": "C:/data/db/diagnostic.log"}]
[{"t": "date", "v": "2021-03-14T22:31:14.883+05:30"}, "s": "1", "c": "NETWORK", "id": "23015", "ctx": "listener", "msg": "Listening on", "attr": {"address": "127.0.0.1"}}, {"t": "date", "v": "2021-03-14T22:31:14.885+05:30"}, "s": "1", "c": "NETWORK", "id": "23016", "ctx": "listener", "msg": "Waiting for connection", "attr": {"port": "20017", "s1": "off"}}
[{"t": "date", "v": "2021-03-14T22:31:23.814+05:30"}, "s": "1", "c": "NETWORK", "id": "22943", "ctx": "listener", "msg": "Connection accepted", "attr": {"remote": "127.0.0.1:50368", "connectionId": 1, "connectionCount": 1}}
[{"t": "date", "v": "2021-03-14T22:31:23.820+05:30"}, "s": "1", "c": "NETWORK", "id": "51800", "ctx": "conn1", "msg": "client metadata", "attr": {"remote": "127.0.0.1:50368", "client": "conn1", "doc": {"driver": {"name": "nodejs"}, "os": {"type": "Windows_NT", "version": "3.6.4"}, "version": "10.0.18363"}, "platform": "Node.js v14.15.5 LE (Unified)", "version": "3.6.4 (5.11.18)"}}
[{"t": "date", "v": "2021-03-14T22:31:34.334+05:30"}, "s": "1", "c": "NETWORK", "id": "22943", "ctx": "listener", "msg": "Connection accepted", "attr": {"remote": "127.0.0.1:50368", "connectionId": 2, "connectionCount": 2}}
[{"t": "date", "v": "2021-03-14T22:31:34.377+05:30"}, "s": "1", "c": "NETWORK", "id": "51800", "ctx": "conn2", "msg": "client metadata", "attr": {"remote": "127.0.0.1:50368", "client": "conn2", "doc": {"driver": {"name": "nodejs"}, "os": {"type": "Windows_NT", "version": "3.6.4"}, "version": "10.0.18363"}, "platform": "Node.js v14.15.5 LE (Unified)", "version": "3.6.4 (5.11.18)"}}
[{"t": "date", "v": "2021-03-14T22:32.22.346+05:30"}, "s": "1", "c": "NETWORK", "id": "22943", "ctx": "listener", "msg": "Connection accepted", "attr": {"remote": "127.0.0.1:50368", "connectionId": 3, "connectionCount": 3}}
[{"t": "date", "v": "2021-03-14T22:32.22.395+05:30"}, "s": "1", "c": "NETWORK", "id": "51800", "ctx": "conn3", "msg": "client metadata", "attr": {"remote": "127.0.0.1:50368", "client": "conn3", "doc": {"driver": {"name": "mongo-jav
a-driver"}, "os": {"type": "Windows", "version": "10.0"}, "version": "1.0.0"}, "platform": "Java/AdoptOpenJDK/11.0.18+80"}}
[{"t": "date", "v": "2021-03-14T22:32.22.419+05:30"}, "s": "1", "c": "NETWORK", "id": "22943", "ctx": "listener", "msg": "Connection accepted", "attr": {"remote": "127.0.0.1:50368", "connectionId": 4, "connectionCount": 4}}
[{"t": "date", "v": "2021-03-14T22:32.22.420+05:30"}, "s": "1", "c": "NETWORK", "id": "51800", "ctx": "conn4", "msg": "client metadata", "attr": {"remote": "127.0.0.1:50368", "client": "conn4", "doc": {"driver": {"name": "mongo-jav
a-driver"}, "os": {"type": "Windows", "version": "10.0"}, "version": "1.0.0"}, "platform": "Java/AdoptOpenJDK/11.0.18+80"}}
[{"t": "date", "v": "2021-03-14T22:32.26.047+05:30"}, "s": "1", "c": "COMMAND", "id": "51803", "ctx": "conn4", "msg": "Slow query", "attr": {"type": "command", "ns": "alumni.alumniusers", "command": "collStats", "sc
ale": 1.0, "db": "alumni", "lsid": {"id": "uuid:d6ad6cc0-db28-4215-94df-68aaeee544ef"}, "numYields": 0, "reslen": 43367, "locks": {"$ReplicationStateTransition": {"acquireCount": {"$w": 1}}, "Global": {"$acquireCount": {"$r": 1}}}, "Database": {"$acquireCount": {"$w": 1}}, "Collection": {"$acquireCount": {"$r": 1}}, "Metadata": {"$acquireCount": {"$r": 1}}, "Storage": {}, "protocol": "op_msg", "durationMills": 130}}
[{"t": "date", "v": "2021-03-14T22:36.15.624+05:30"}, "s": "1", "c": "WRITE", "id": "51803", "ctx": "LogicalSessionCacheRefresh", "msg": "Slow query", "attr": {"type": "update", "ns": "config.system.sessions", "command": "q"}, "db": "config", "lsid": {"id": "uuid:d6ad6cc0-db28-4215-94df-68aaeee544ef"}, "uid": "$binary:[base64-470E0jBHSa/IMz5KeUroPm0jWZG03SuFuS, subtype:0]"}, "user": {"$set": {"lastuse": "$SNOWFLAKE_TIMESTAMP"}, "multi": false, "upsert": true}, "planSummary": "IDACK", "keysExamined": 0, "docsExamined": 0, "hatched": 0, "modified": 0, "nModified": 0, "upsert": true, "keysInserted": 2, "numYields": 0, "locks": {"$ParallelBatchWriterMode": {"$acquireCount": {"$w": 1}}}, "ReplicationStateTransition": {"$acquireCount": {"$w": 1}}, "Global": {"$acquireCount": {"$w": 1}}, "Database": {"$acquireCount": {"$w": 1}}, "Collection": {"$acquireCount": {"$w": 1}}, "Metadata": {"$acquireCount": {"$w": 1}}, "Storage": {}, "protocol": "op_query", "durationMills": 2881}
[{"t": "date", "v": "2021-03-14T22:36.15.625+05:30"}, "s": "1", "c": "COMMAND", "id": "51803", "ctx": "LogicalSessionCacheRefresh", "msg": "Slow query", "attr": {"type": "command", "ns": "config.$cmd", "command": "update", "syste
mSessions": "ordered": false, "allowImplicitCollectionCreation": false, "writeConcern": {"$w": "majority", "timeout": 15000}, "db": "config"}, "numYields": 0, "reslen": 171, "locks": {"$ParallelBatchWriterMode": {"$acquireCount": {"$w": 1}}}, "Global": {"$acquireCount": {"$r": 1, "w": 1}}, "Database": {"$acquireCount": {"$w": 1}}, "Collection": {"$acquireCount": {"$w": 1}}, "Metadata": {"$acquireCount": {"$r": 1}}, "Storage": {}, "protocol": "op_msg", "durationMills": 2191}
[{"t": "date", "v": "2021-03-14T22:49:45.297+05:30"}, "s": "1", "c": "NETWORK", "id": "22944", "ctx": "conn2", "msg": "Connection ended", "attr": {"remote": "127.0.0.1:50368", "connectionId": 2, "connectionCount": 3}}
[{"t": "date", "v": "2021-03-14T22:49:45.641+05:30"}, "s": "1", "c": "NETWORK", "id": "4615610", "ctx": "conn1", "msg": "Failed to connect to socket connectivity", "attr": {"error": "The operation completed successfully."}}
[{"t": "date", "v": "2021-03-14T22:49:45.641+05:30"}, "s": "1", "c": "NETWORK", "id": "22944", "ctx": "conn2", "msg": "Interrupted operation as its client disconnected", "attr": {"opId": 140939}}
[{"t": "date", "v": "2021-03-14T22:49:45.691+05:30"}, "s": "1", "c": "NETWORK", "id": "22989", "ctx": "conn1", "msg": "Error sending response to client. Ending connection from remote", "attr": {"error": {"code": 6, "codeName": "HostUnreachable"}, "errmsg": "Connection reset by peer"}, "remote": "127.0.0.1:50368", "connectionId": 1}
[{"t": "date", "v": "2021-03-14T22:49:45.692+05:30"}, "s": "1", "c": "NETWORK", "id": "22944", "ctx": "conn1", "msg": "Connection ended", "attr": {"remote": "127.0.0.1:50368", "connectionId": 1, "connectionCount": 2}}
[{"t": "date", "v": "2021-03-14T22:52.01.782+05:30"}, "s": "1", "c": "NETWORK", "id": "22943", "ctx": "listener", "msg": "Connection accepted", "attr": {"remote": "127.0.0.1:50368", "connectionId": 5, "connectionCount": 3}}
[{"t": "date", "v": "2021-03-14T22:52.01.811+05:30"}, "s": "1", "c": "NETWORK", "id": "51800", "ctx": "conn3", "msg": "client metadata", "attr": {"remote": "127.0.0.1:50368", "client": "conn3", "doc": {"driver": {"name": "nodejs"}, "os": {"type": "Windows_NT", "version": "3.6.4"}, "version": "10.0.18363"}, "platform": "Node.js v14.15.5 LE (Unified)", "version": "3.6.4 (5.11.18)"}}
[{"t": "date", "v": "2021-03-14T22:52.01.847+05:30"}, "s": "1", "c": "NETWORK", "id": "22943", "ctx": "listener", "msg": "Connection accepted", "attr": {"remote": "127.0.0.1:50368", "connectionId": 6, "connectionCount": 4}}
[{"t": "date", "v": "2021-03-14T22:52.01.851+05:30"}, "s": "1", "c": "NETWORK", "id": "22943", "ctx": "listener", "msg": "Connection accepted", "attr": {"remote": "127.0.0.1:50368", "connectionId": 7, "connectionCount": 5}}
[{"t": "date", "v": "2021-03-14T22:52.01.855+05:30"}, "s": "1", "c": "NETWORK", "id": "51800", "ctx": "conn4", "msg": "client metadata", "attr": {"remote": "127.0.0.1:50368", "client": "conn4", "doc": {"driver": {"name": "nodejs"}, "os": {"type": "Windows_NT", "version": "3.6.4"}, "version": "10.0.18363"}, "platform": "Node.js v14.15.5 LE (Unified)", "version": "3.6.4 (5.11.18)"}}
[{"t": "date", "v": "2021-03-14T22:52.01.858+05:30"}, "s": "1", "c": "NETWORK", "id": "51800", "ctx": "conn1", "msg": "client metadata", "attr": {"remote": "127.0.0.1:50368", "client": "conn1", "doc": {"driver": {"name": "nodejs"}, "os": {"type": "Windows_NT", "version": "3.6.4"}, "version": "10.0.18363"}, "platform": "Node.js v14.15.5 LE (Unified)", "version": "3.6.4 (5.11.18)"}}
[{"t": "date", "v": "2021-03-14T22:52.01.861+05:30"}, "s": "1", "c": "NETWORK", "id": "22943", "ctx": "listener", "msg": "Connection accepted", "attr": {"remote": "127.0.0.1:50368", "connectionId": 8, "connectionCount": 6}}
[{"t": "date", "v": "2021-03-14T22:52.01.865+05:30"}, "s": "1", "c": "NETWORK", "id": "51800", "ctx": "conn2", "msg": "client metadata", "attr": {"remote": "127.0.0.1:50368", "client": "conn2", "doc": {"driver": {"name": "nodejs"}, "os": {"type": "Windows_NT", "version": "3.6.4"}, "version": "10.0.18363"}, "platform": "Node.js v14.15.5 LE (Unified)", "version": "3.6.4 (5.11.18)"}]
```

Figure 7: Database connection successful

CHAPTER 7

OUTPUT OF THE SYSTEM

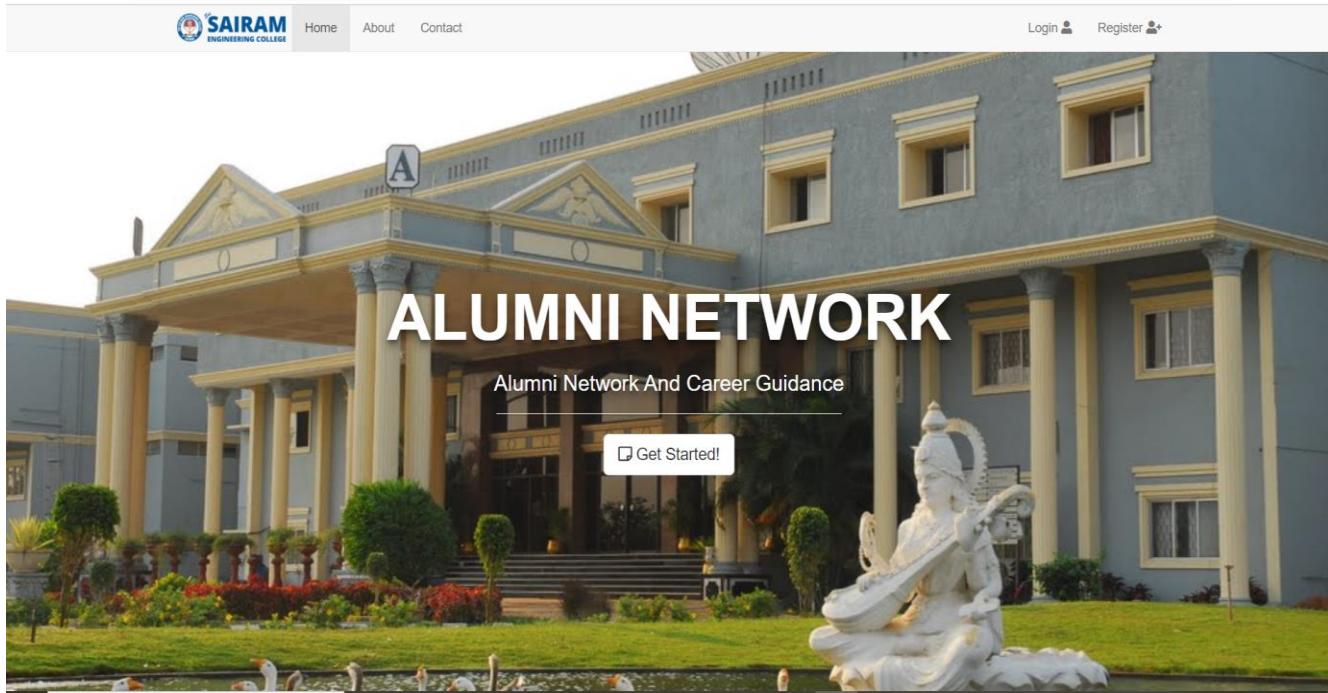


Figure 8: Landing Page

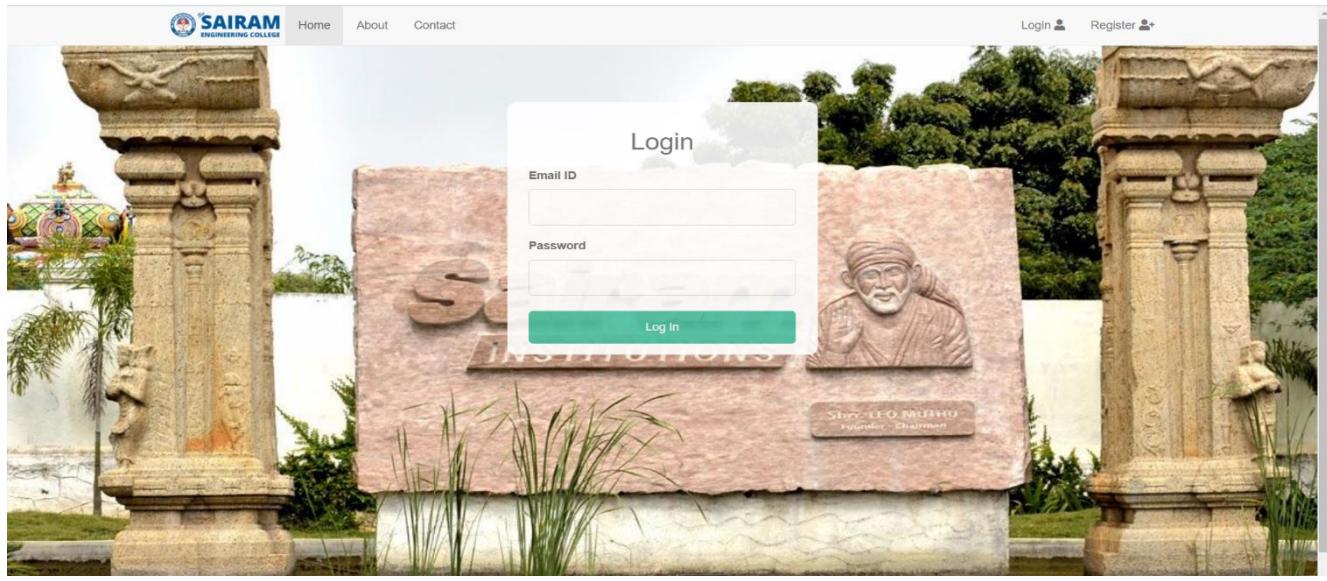


Figure 9: Login Page

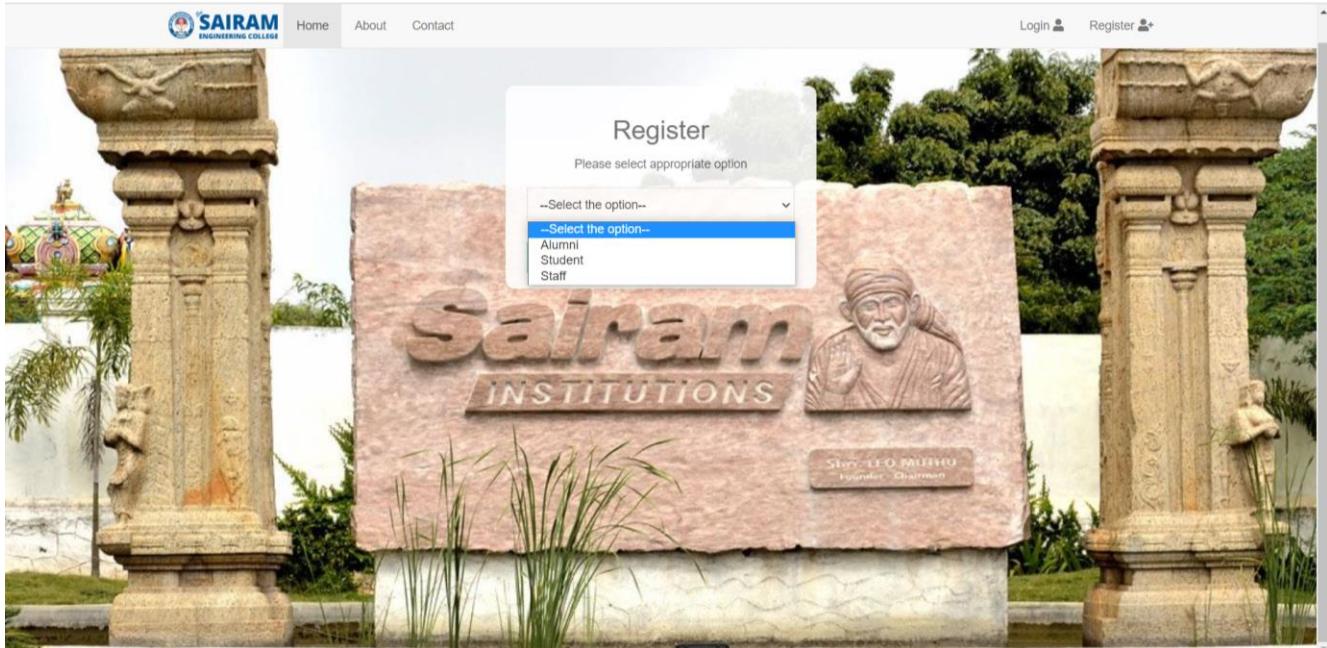


Figure 10: Registration page

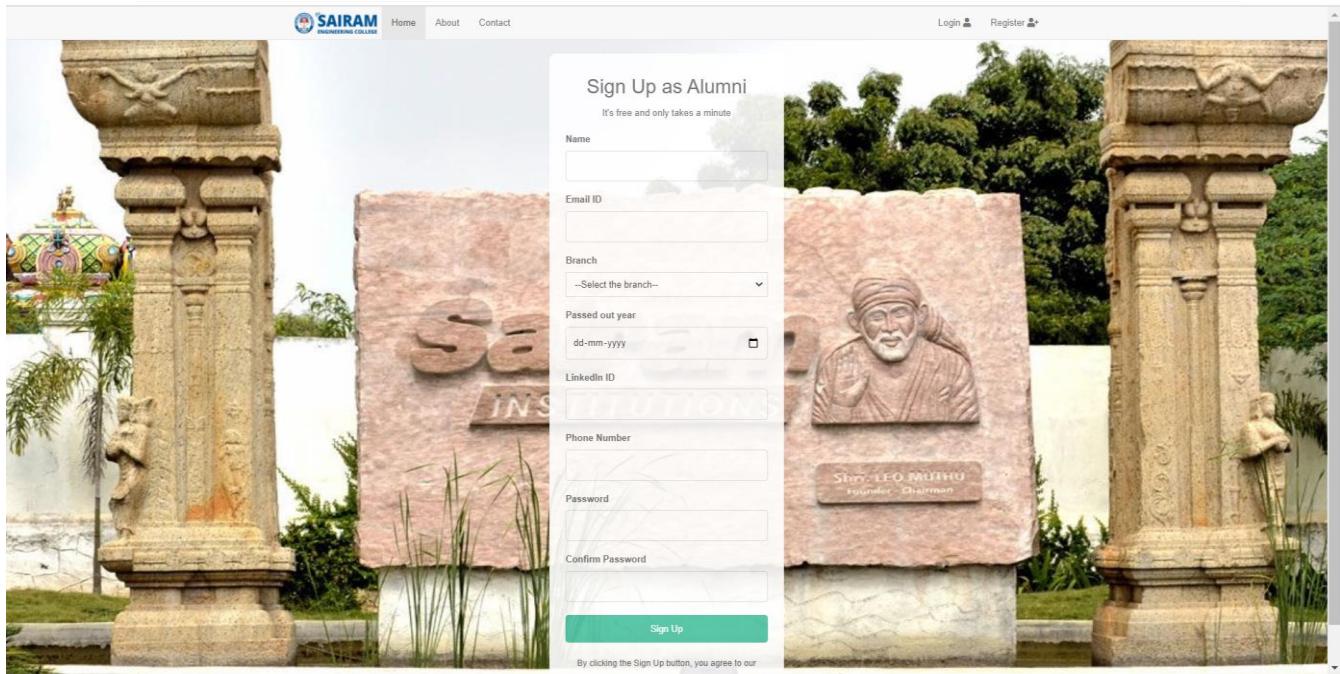


Figure 11: Signup page for Alumni

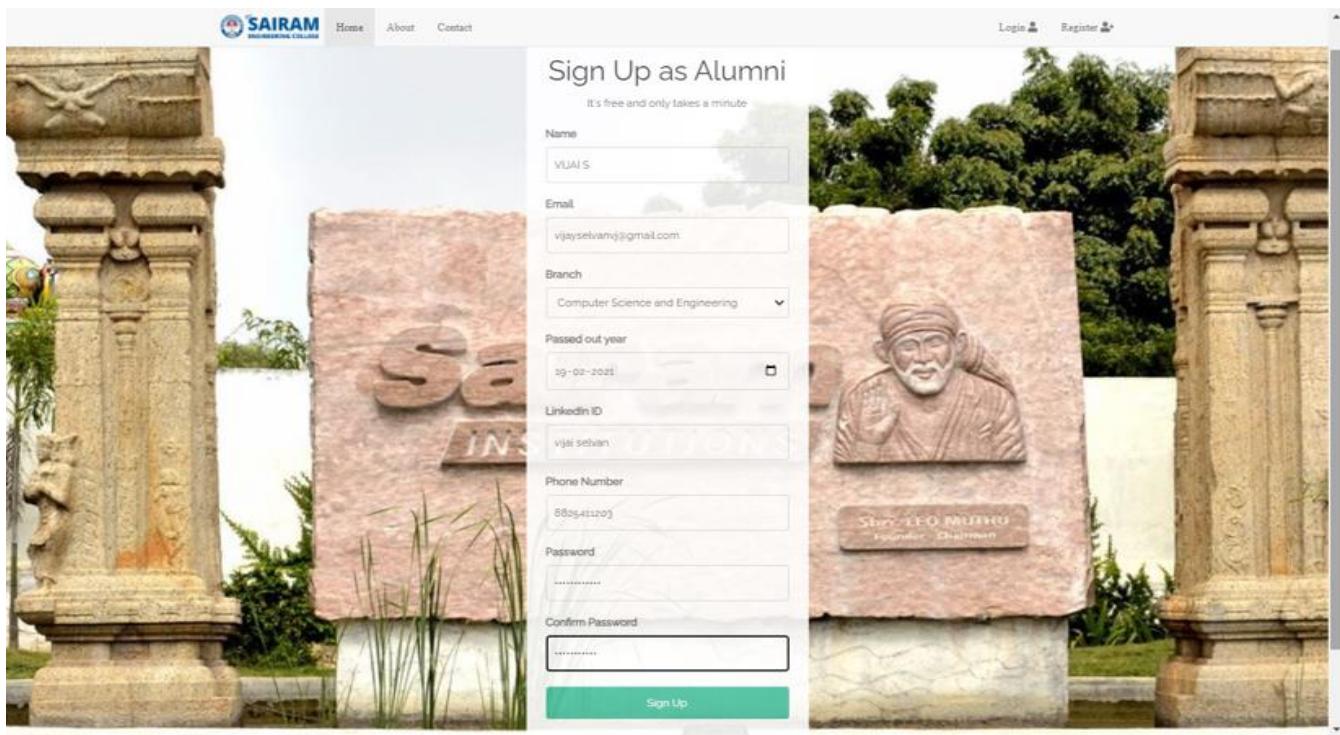


Figure 12: Signup page for Alumni with Data

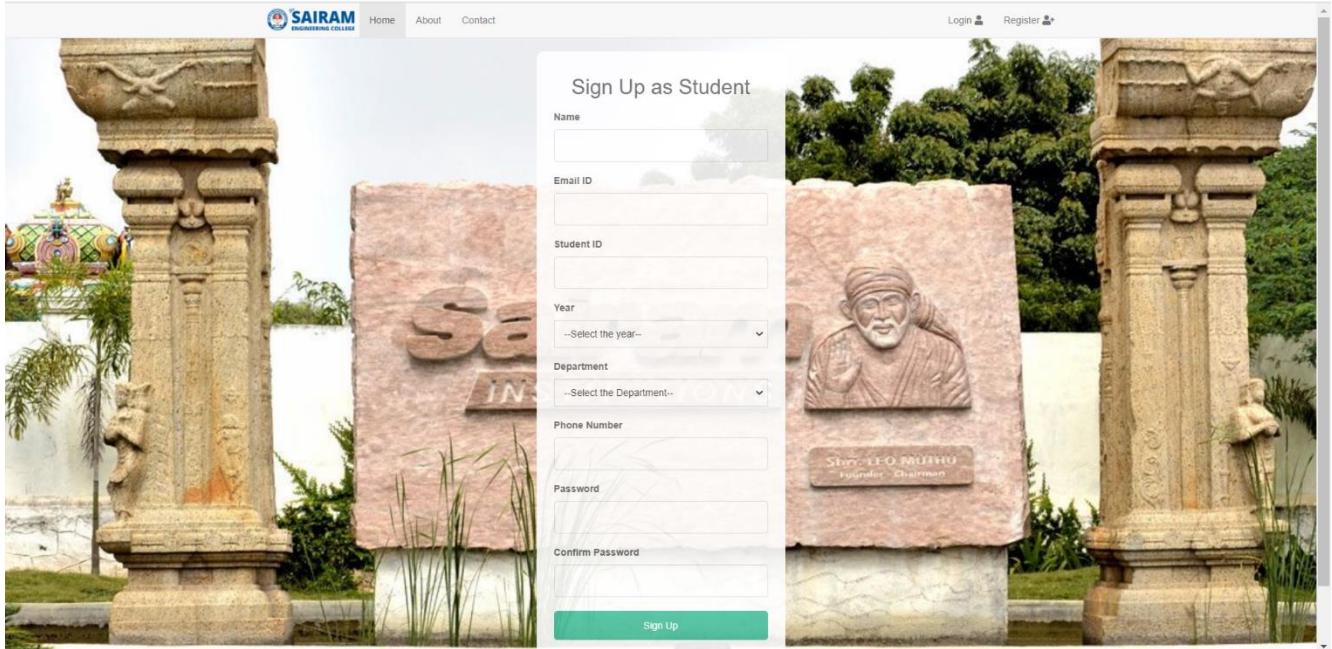


Figure 13: Signup page for Student

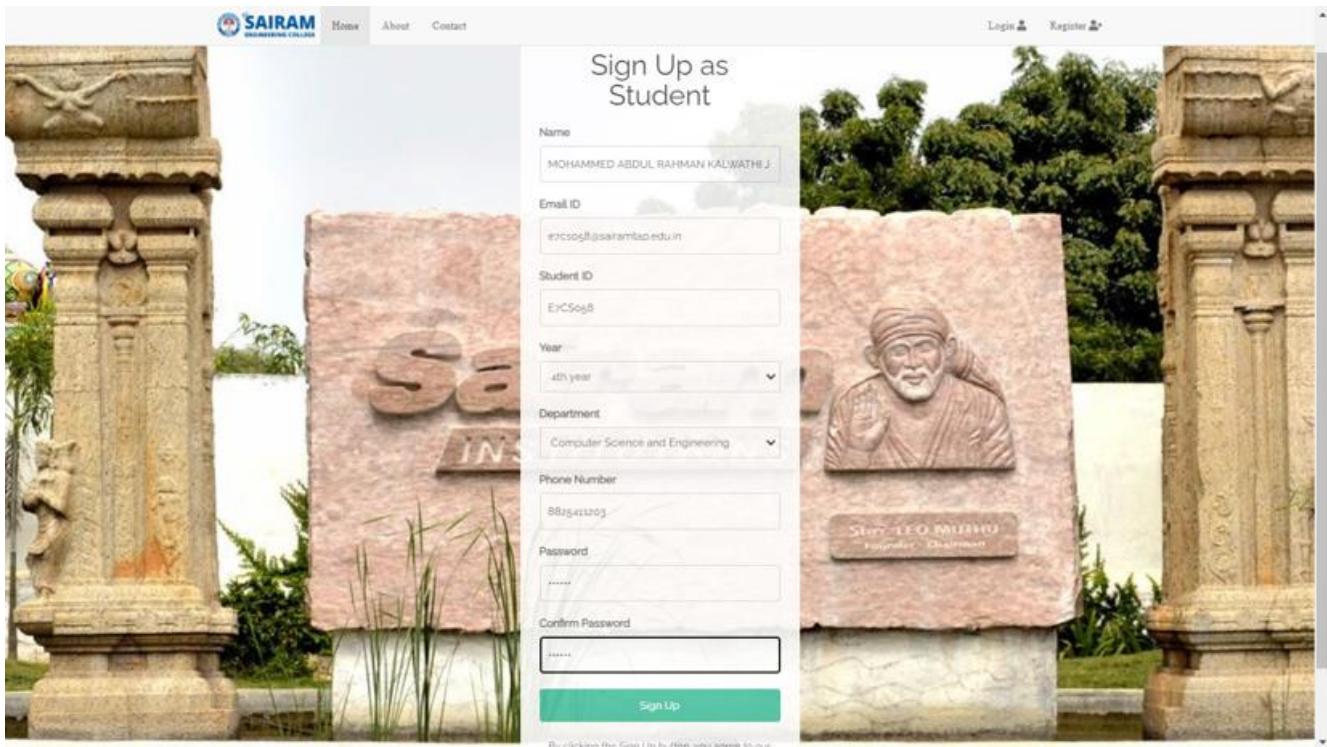


Figure 14: Signup page for Student with Data

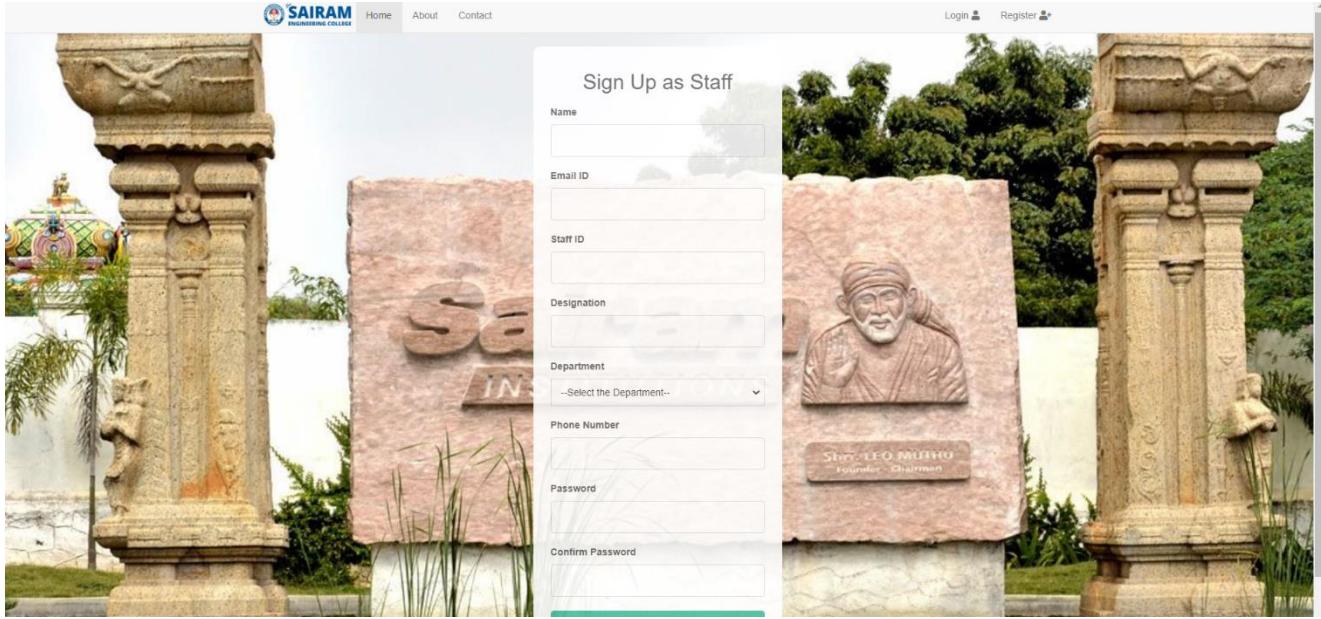


Figure 15: Figure: Signup page for Staff

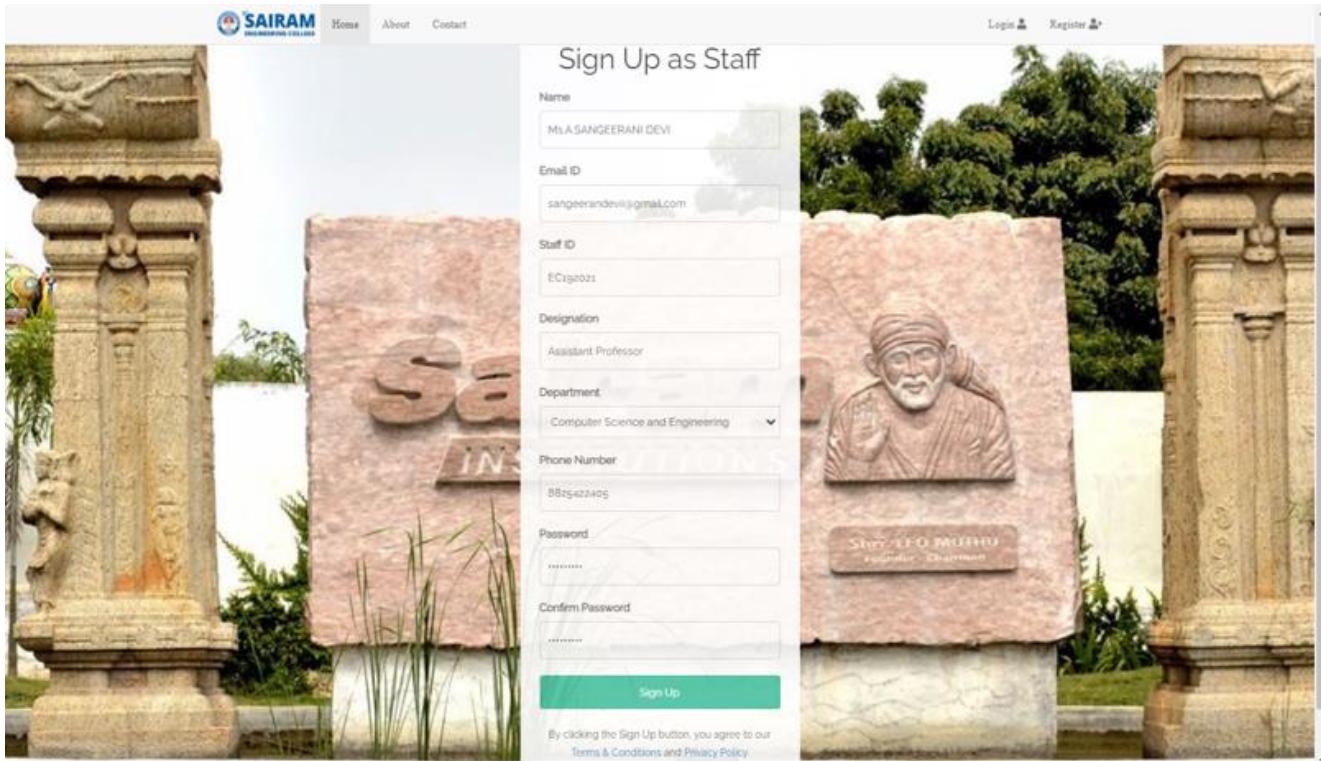


Figure 16: Signup page for Staff with Data

The screenshot shows the Studio 3T interface for MongoDB. The left sidebar displays the database structure under 'alumni'. The main pane shows a query result for the 'alumniusers' collection. The JSON output is as follows:

```

1 {
2   "_id" : ObjectId("6040f224f061805926ad56a6"),
3   "isAlumni" : true,
4   "username" : "vijayselvan123",
5   "Name" : "vijayselvan",
6   "CompanyName" : "cognizant",
7   "Designation" : "programmer analyst",
8   "PassedOutYear" : ISODate("2021-05-30T00:00:00.000+0000"),
9   "LinkedInID" : "vijay@selvan",
10  "PhoneNo" : "9944406957",
11  "EmailId" : "vijayselvanj@gmail.com",
12  "salt" : "af84526ecb9d5803b495e3e30069755029e1e0fa2a46d13a037c9c4a3ce8eald",
13  "hash" : "1a723611f7699c1ee87e41b178b4e25434968a8e7b004717212b285385ba09a95150f0c1fbfa22ef92cd83d7519c14e73352572fd6babdb011c8985bdb2f0d078e55911e",
14  "__v" : NumberInt(0)
15 }

```

The bottom status bar indicates '1 document selected'.

Figure 17: Alumni data in JSON format in database

The screenshot shows the Studio 3T interface for MongoDB. The left sidebar displays the database structure under 'alumni'. The main pane shows a query result for the 'studentusers' collection. The JSON output is as follows:

```

1 {
2   "_id" : ObjectId("6040f6afc11f0b062f4c5942"),
3   "isStudent" : true,
4   "username" : "abdul123",
5   "Name" : "mohammed abdul rahman kalwathi.j",
6   "StudentId" : "e7cs058",
7   "Year" : "4",
8   "Department" : "CSE",
9   "PhoneNo" : "9944406957",
10  "EmailId" : "mkalwathi@gmail.com",
11  "salt" : "38abe245b4dcba44c12fcf1386496ade8ccaf74ec7ae1959048125c5676c08cc",
12  "hash" : "6443938f145881db4a7c5491d67715e1617ae37490d4b9c2e895275b3832af869744cef758f556a0332ec6f4a374ca6157f5636d1648bf0c38dd9d5477170e1f2c861c",
13  "__v" : NumberInt(0)
14 }

```

The bottom status bar indicates '1 document selected'.

Figure 18: Student data in JSON format in database

The screenshot shows the Studio 3T interface for MongoDB. The left sidebar displays the database structure with collections: admin, alumni, Collections (3), staffusers, and studentusers. The main pane shows a query builder for the 'staffusers' collection. The query is:

```

1 {
2     "_id" : ObjectId("6040f4c65c68c2b066b4f898"),
3     "isStaff" : true,
4     "username" : "sangeerani123",
5     "Name" : "sangeerani devi",
6     "StaffId" : "hg456",
7     "Designation" : "assistant professor",
8     "Department" : "CSE",
9     "PhoneNo" : "9944406957",
10    "EmailId" : "sangeerani.devi@gmail.com",
11    "salt" : "bf25da424da4944ffdc4539801fab29176314cf8b5be9bae53227b0701285a7d",
12    "hash" : "80e9f6f90fa848ffff84d17372cdff310ef40f3936ff4a4db1eb20e28fb855fa77098d2681927d47e0f09ba9848c189473f7a84d3c7969053f019ea47daa3a54992dc0d4",
13    "__v" : NumberInt(0)
14 }
15

```

The result pane shows one document selected. The bottom status bar indicates "1 document selected".

Figure 19: Student data in JSON format in database

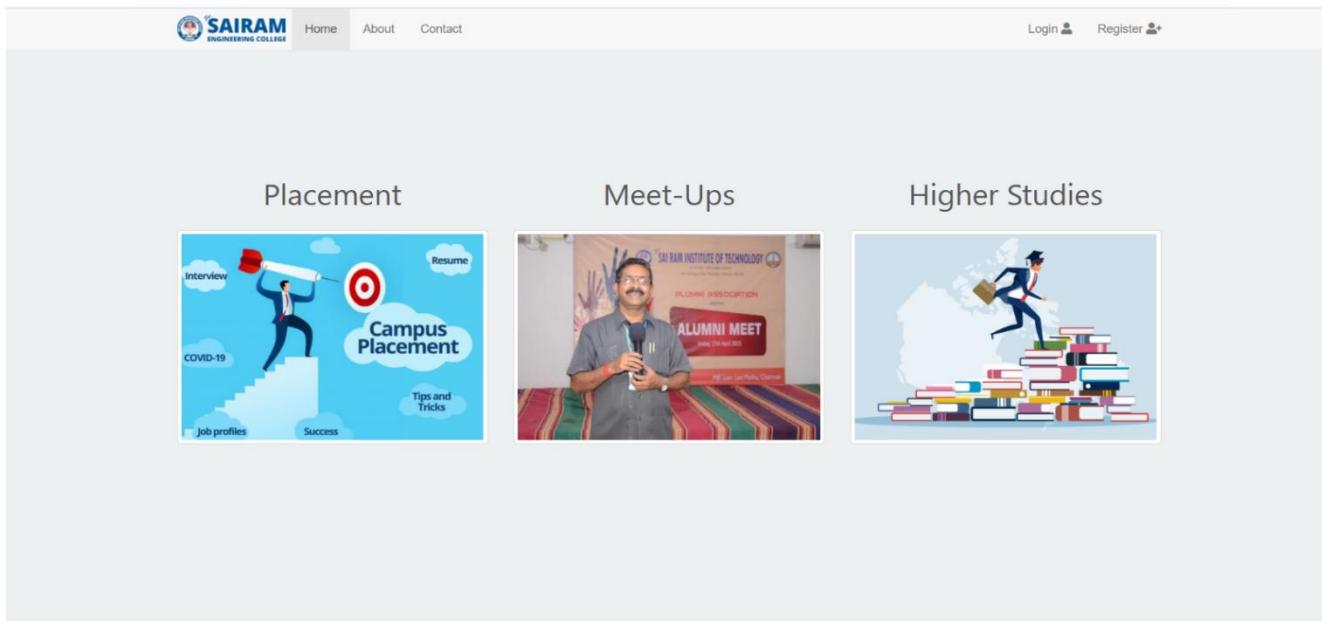


Figure 20: Higher studies and placement query page (After Student login)



Figure 21: Placement and Company review page

WELCOME MASTERS



MS in Computer Science

MS COMPUTER SCIENCE

The Master of Science in Computer Science (M.S. CS) program is a terminal degree program designed to prepare students for more highly productive careers in industry. Graduates receive the M.S. ... The program is designed for students who possess a bachelor's degree in computer science from an accredited institution..

[click on..](#)



MBA

A Master of Business Administration or MBA is a general graduate business degree that teaches students technical, managerial, and leadership skills. Earning an MBA gives you valuable business acumen, expands your professional network, and creates new opportunities. It's an ideal option to consider whether you want to accelerate your career trajectory or become an entrepreneur..

[click on..](#)



CSE
Computer Science & Engineering

M-Tech
Master of Technology

M.TECH COMPUTER SCIENCE

Career after M.Tech in Computer science Engineering Tech in Computer Science and Engineering. The candidates can find job opportunities in both private and public sector firms. The candidates can find lot of openings in software sector. The candidates can work in different areas in the software industry.

[click on..](#)

Figure 22: Higher Studies Review page

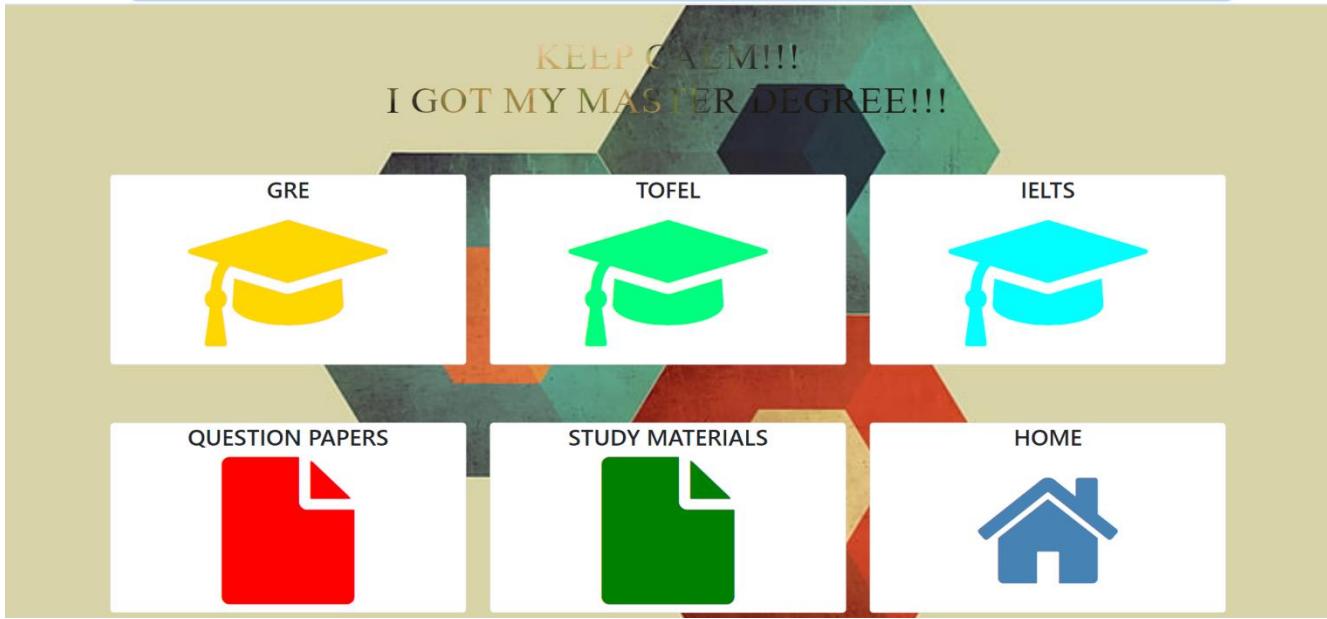


Figure 23: MS in computer science Query page



Figure 24: MBA Query page

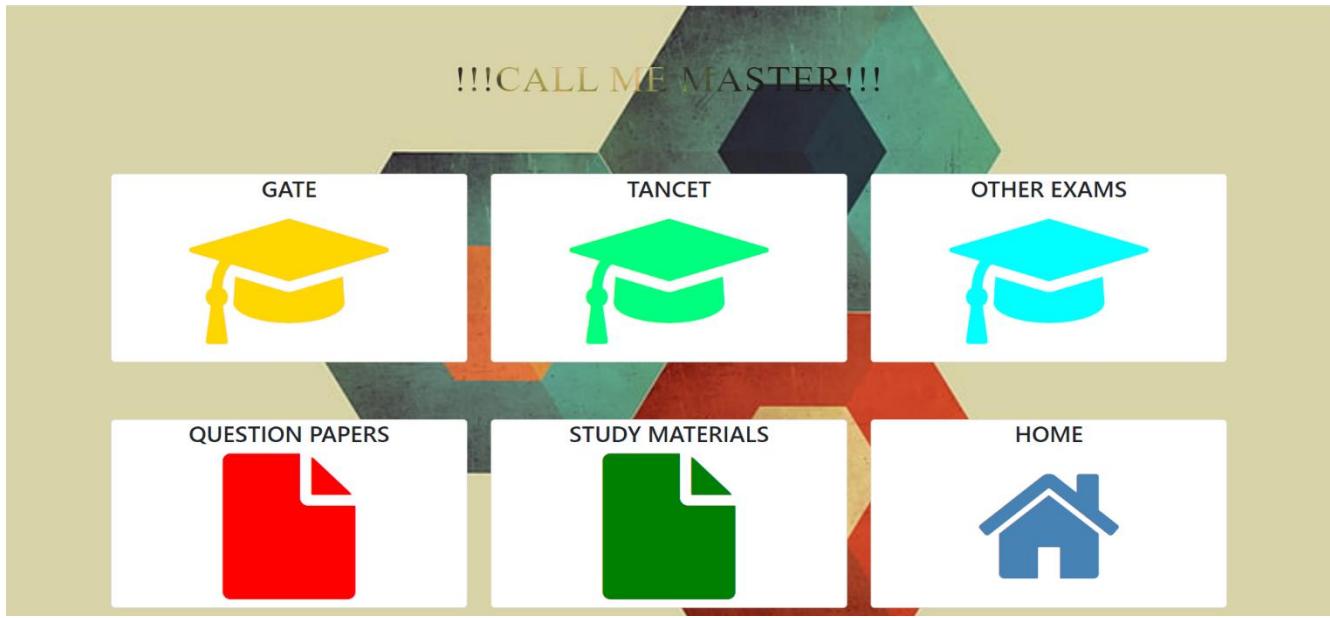


Figure 25: MTech in computer Science Query page

Downloads

Category	File Name	Action
GRE	GRE 2020 Q&A.pdf	
	GRE 2019 Q&A.pdf	
	GRE 2018 Q&A.pdf	
	GRE 2017 Q&A.pdf	
TOFEL	TOFEL sample paper2 Q&A.pdf	
	TOFEL speaking integrated Q&A.pdf	
	TOFEL ibt full length Test Q&A.pdf	
	TOFEL Listening Q&A.pdf	
	IELTS	IELTS general writing task Q&A.pdf
IELTS Reading practice paper Q&A.pdf		
IELTS Speaking Q&A.pdf		
IELTS Practice Test Q&A.pdf		
IELTS		IELTS Practice Test Q&A.pdf

Practice Test

GRE Practise Test **Start Test**

TOFEL Practise Test **Start Test**

IELTS Practise Test **Start Test**

Figure 26: MS in computer Science Practice question paper and practice test

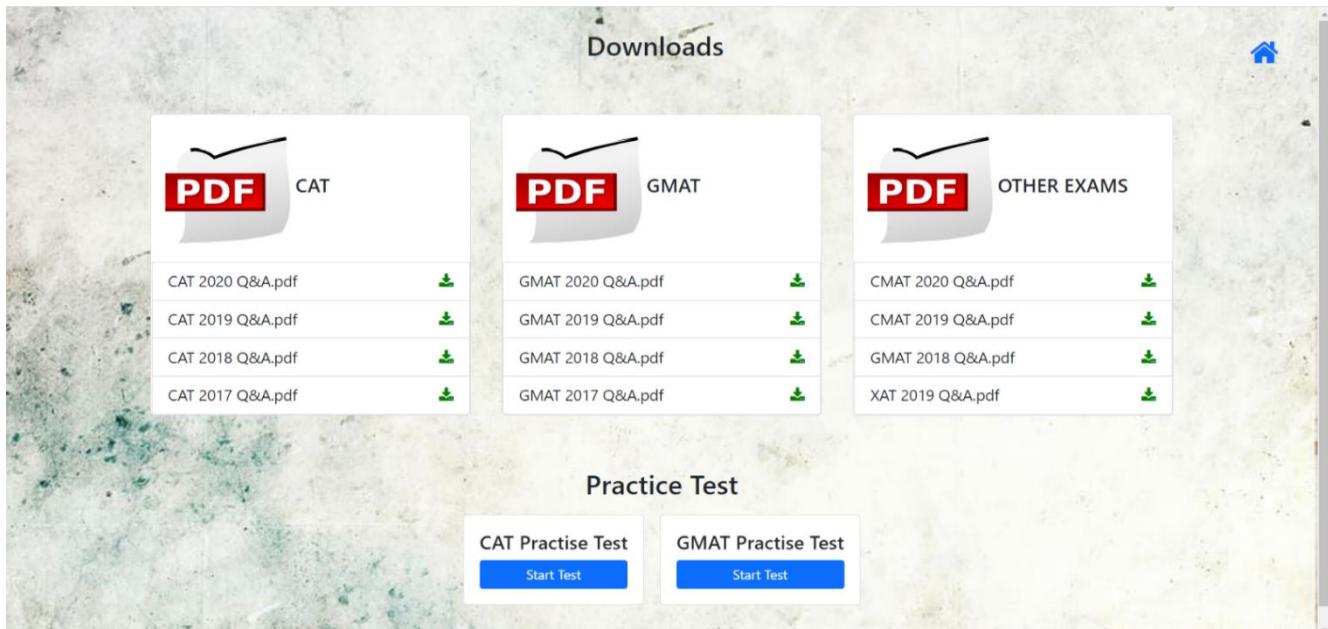


Figure 27: MBA Practice question paper and practice test

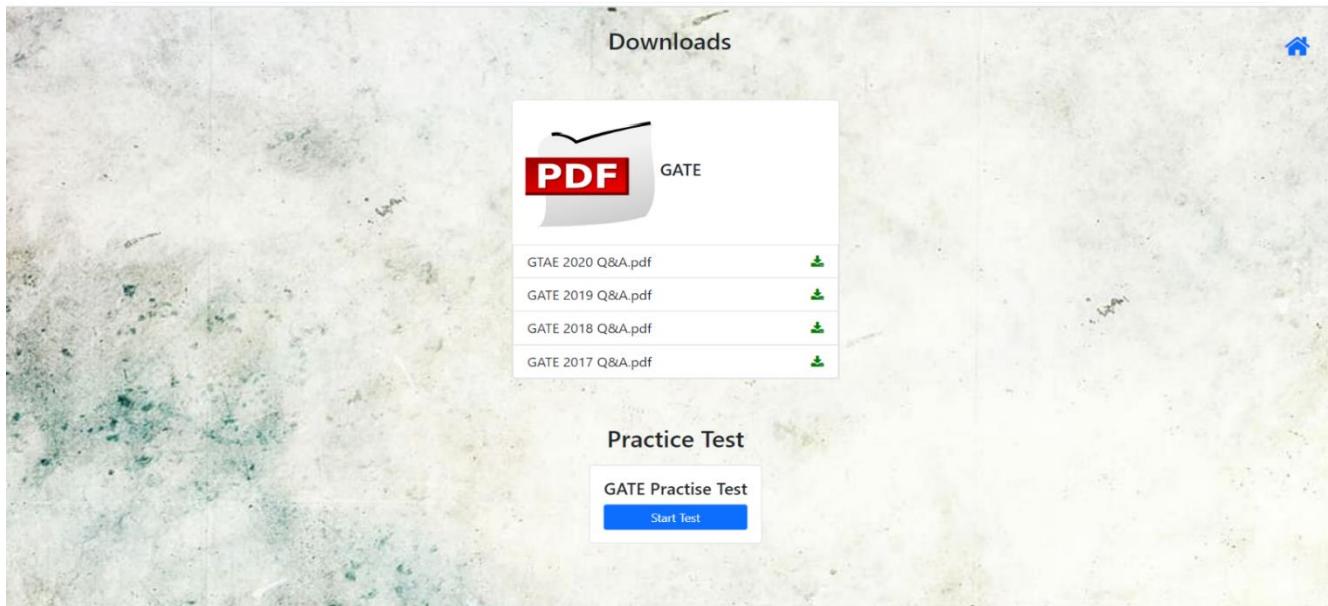


Figure 28: MTech in computer Science Practice question paper and practice test

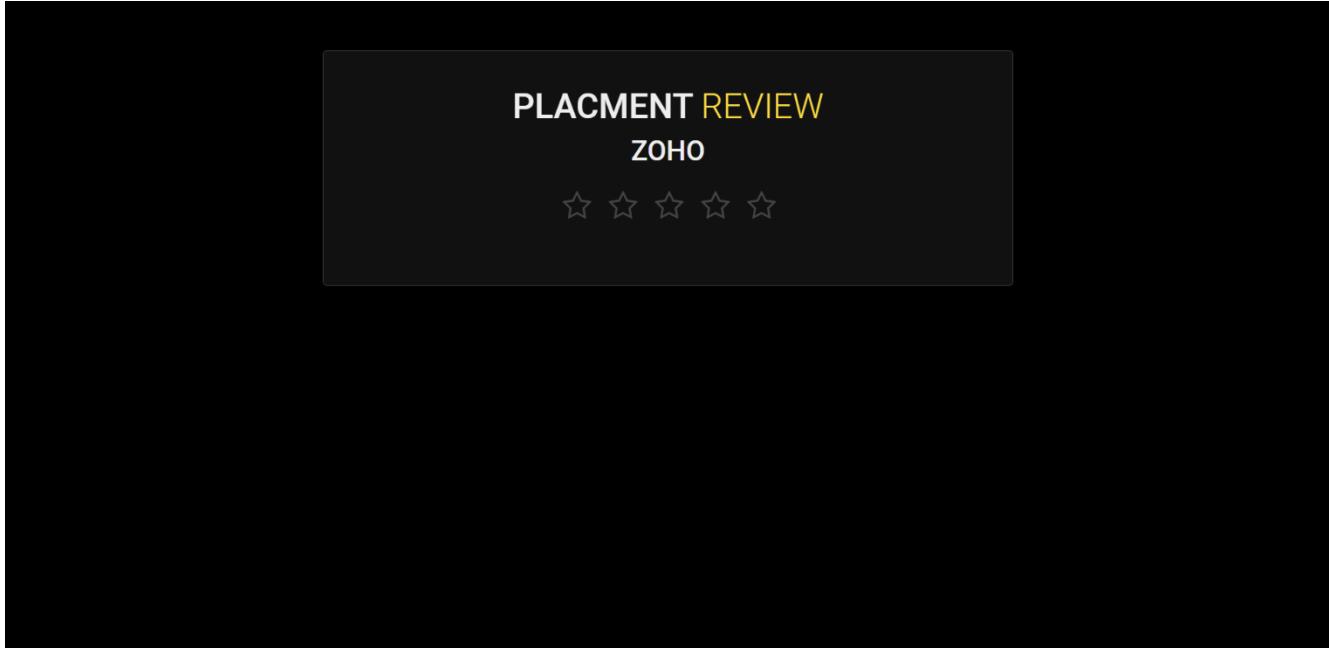


Figure 29: Placement review Page (After Alumni login)

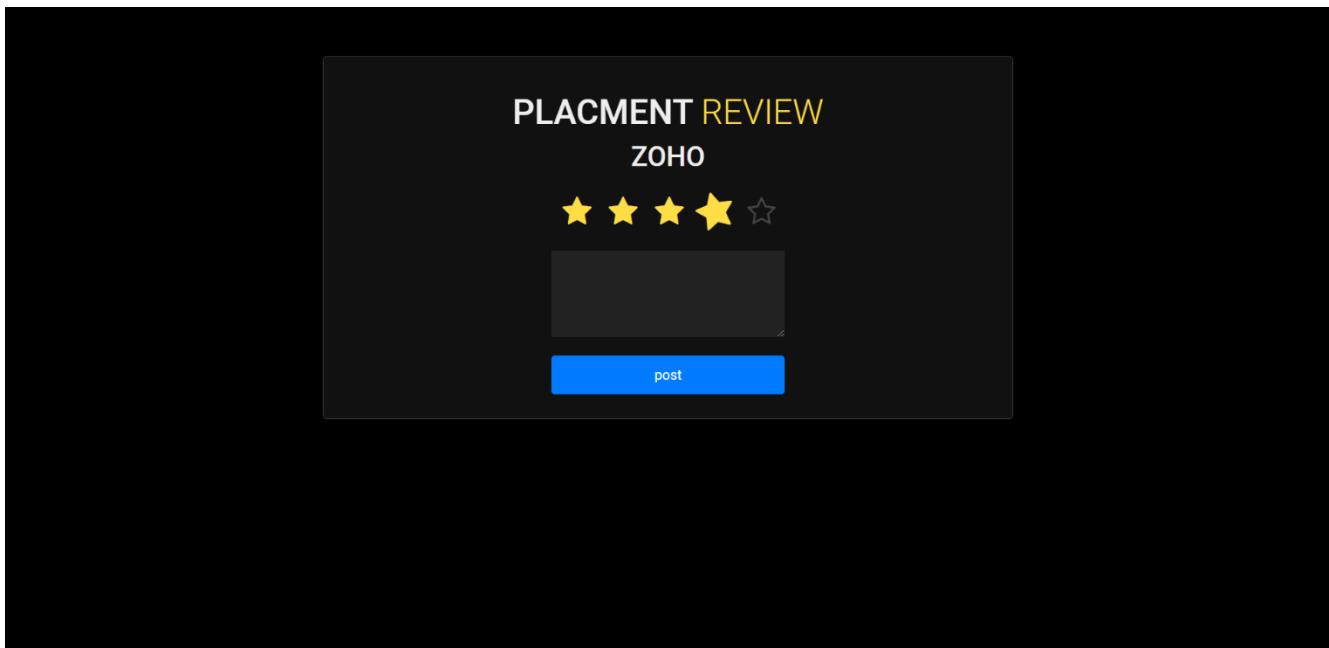


Figure 30: Rating for the company

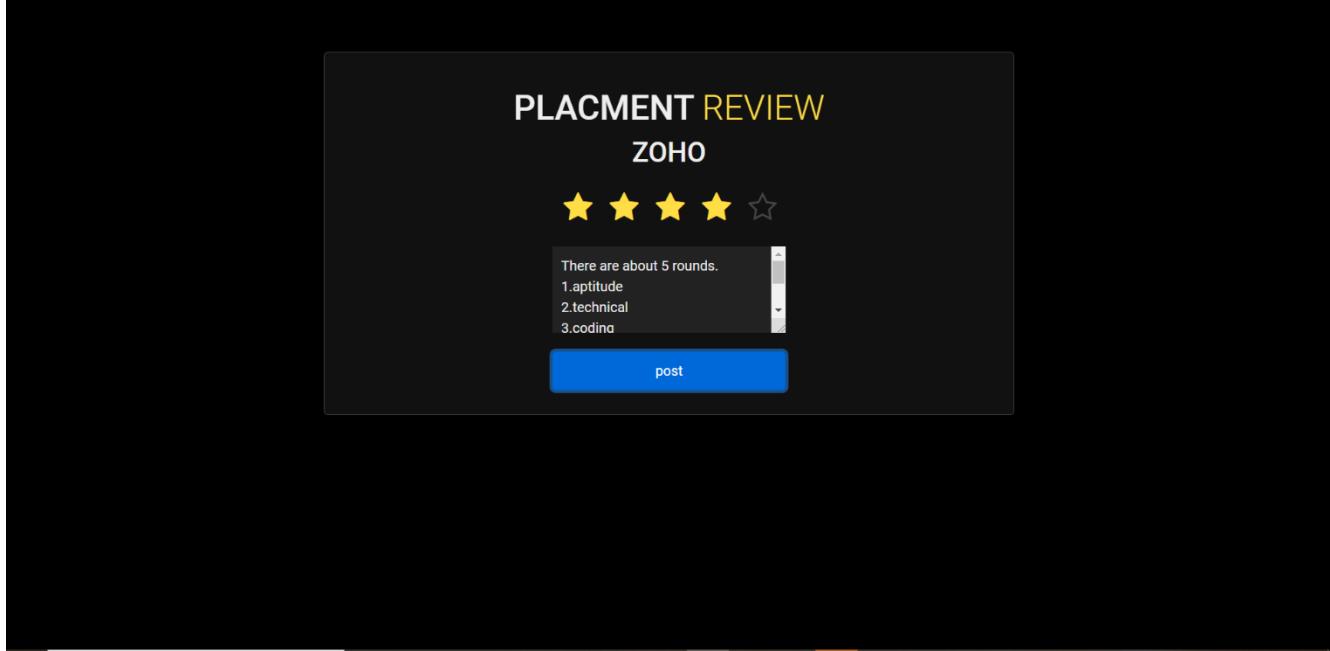


Figure 31: Getting feedback for the company about placement

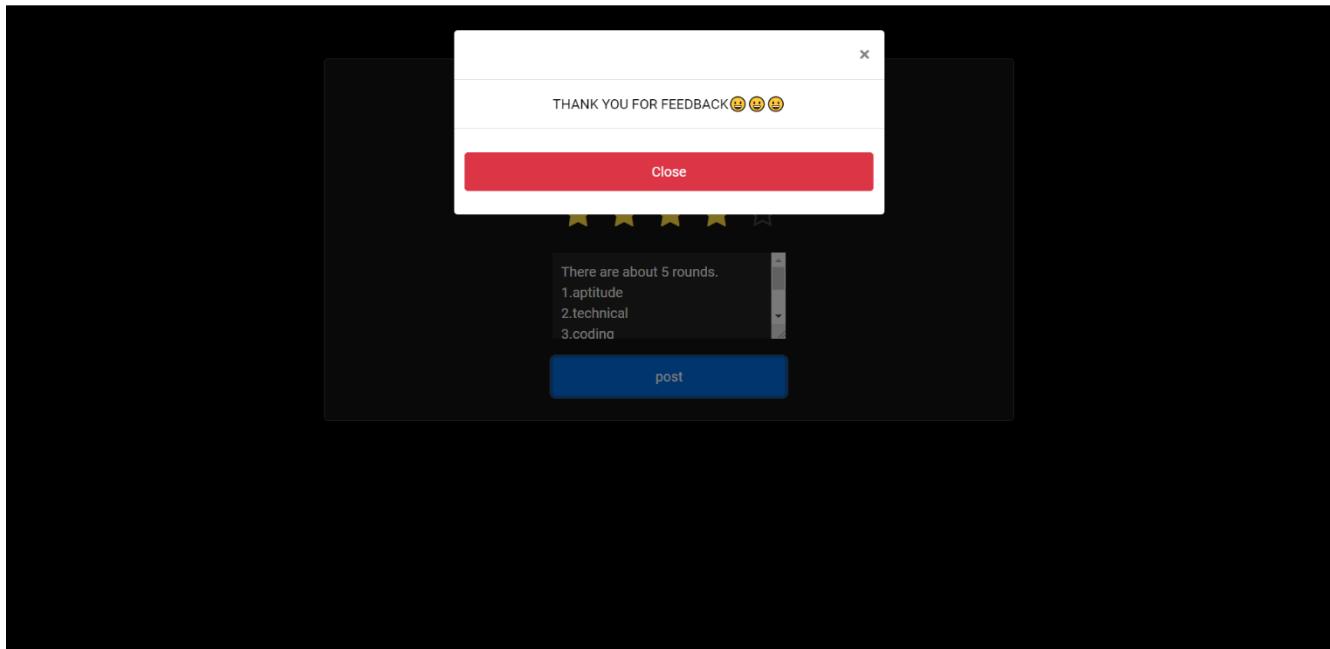


Figure 32: Thank you message

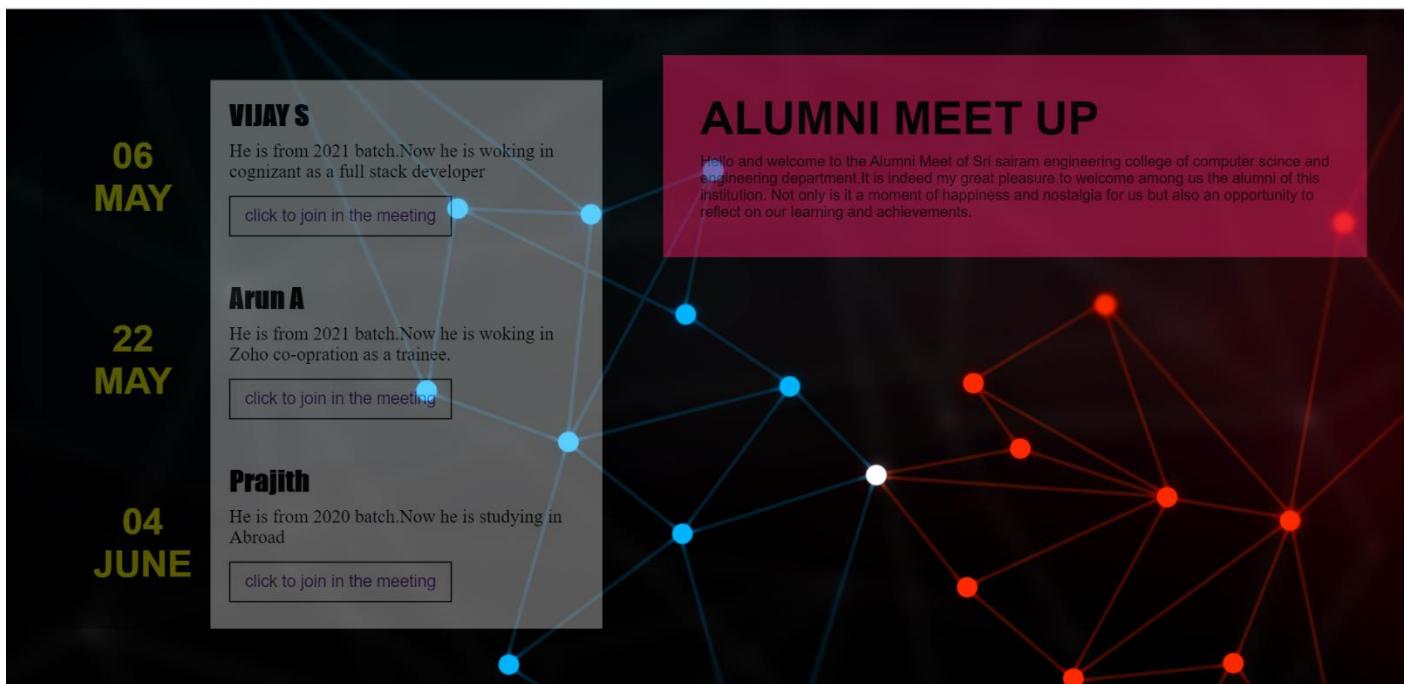


Figure 33: Meet up page

CHAPTER 8

TESTING

8.1 INTRODUCTION:

Testing is generally carried out to ensure the correct and verified model of the project. It plays a key role in validating the working model. Testing is the process of evaluating a system or its component(s) with the intent to find whether it satisfies the specified requirements or not.

Testing is executing a system in order to identify any gaps, errors, or missing requirements in contrary to the actual requirements.

8.2 TESTING STRATEGIES

- Whitebox Testing
- Blackbox Testing
- Unit Testing
- Functional Testing
- Intergration Testing
- Validation Testing
- System Testing
- Structure Testing

The brief description of the above metrics are discussed below.

8.2.1 WHITE BOX TESTING

White Box testing is also known as glass box testing. This type of testing, tests the internal structure of the program. This can be applied at the unit, integration and system levels of testing. Mostly, it is used in the unit level of the software testing process. Sometimes it may not reveal defects in areas which have not been implemented. It has its own advantages and its own disadvantages. The advantage is that knowing the programming language code and familiarizing with them may prove vital and help in identifying the errors quickly and at times may help in avoiding them at the earliest.

8.2.2 BLACK BOX TESTING

It is amongst the two methods of mostly used testing methods. These tests the main functionality of the program. It can be applied to every level of testing such as Unit, Integration, System and Acceptance levels of testing. Exhaustive input testing is required to find all errors. For doing this type of testing knowing the internal code and how it works is not needed but what it is supposed to do is known by the person who is performing the test. The test cases are developed based on the specific requirements according to the goals. There are Boundary Value Analysis, Class Partitioning, Cause Effect Graph etc.,

8.2.3 UNIT TESTING:

Unit testing is also known as Module Testing which focuses on verification efforts on the module. The module is tested separately and this is carried out at the programming stage itself. Unit test comprises of the set of tests performed by an individual programmer before integration of the unit into the system. This will help to test each and every single part or we can say as each and every module completely. These may even be small parts of the code and test cases will be developed which are independent of each other.

8.2.4 FUNCTIONAL TESTING:

Functional testing is mainly used as a Quality Assurance process. This is a very simple process where each function is provided with an appropriate input and is verified against an expected output and with boundary values. This would help in ensuring that the output is as acquired according to the expectations and would help assuring the quality. The various functions developed using Java are separately tested for their proper working by executing them as separate files.

8.2.5 INTEGRATION TESTING:

It is a systematic technique for constructing the program structure while at the same time conducting tests to uncover errors associated with in the interface. Integration of all the components to form the entire system and an overall testing is executed. Integration testing is any type of software testing that seeks to verify the interfaces between components against a software design. Software components may be integrated in an iterative way or all together as in a on the whole approach. Normally the former is considered a better practice since it allows interface issues to be located more quickly and fixed. It is mainly done based on by taking into account of the number of modules used, how many number of interfaces maybe required to integrate them, which had to be combined and clustering process.

8.2.6 VALIDATION TESTING:

Validation test succeeds when the software functions in a manner that can be reasonably expected by the client. Software validation is achieved through a series of black box testing which confirms to the requirements. The software is validated based on the series of tests that it passes through according to the condition posed by the customer. Mostly the customer main requirements would be to make every process as simple as possible and to reduce the complexity of the usage of the final product. Taking all these conditions into mind the validation testing is done and the various test cases are design.

8.2.7 SYSTEM TESTING:

System testing of software or hardware is testing conducted on complete, integrated system to evaluate the system's compliance with its specified requirements. Once all of the modules have been completely developed and both unit testing and integration testing is done on the various parts of the modules later the system testing is done so as to ensure that the requirements are fulfilled properly. All it basically does is it performs tests to find the discrepancies between the system and its original objective, current specifications and system documentation. If any discrepancy is to be found the respective errors will be rectified and again system testing will be performed to make sure the rectification does not introduce a new error into the system

8.2.8 STRUCTURE TESTING:

It is concerned with exercising the internal logic of a program and traversing certain execution paths. Structure testing takes into account of all process that works internally to make the entire system to work properly. The basic structure and the background codes and fragments that help in upholding the system are used.

This goes layer by layer as in till reaching the core process. The various layers or levels depends on the type of the project and the domain it comes under. This is considered to be a part of the White Box Testing Process. This is done so as to make sure all the internal processes work properly. If They don't then the probability that the entire process may collapse is a possibility and this causes a grave danger to the project leading to failure.

CHAPTER 9

CONCLUSION

The system is developed for mainly sharpening the communication between Students and Alumni and also providing career guidance in this digital era. We see most of the institutions are encouraging students to actively engage in Alumni meetings and meetups but we felt that a good platform with all the facilities to communicate with Alumni and listen to their feedback to follow their success steps could ease their activities in upskilling their career. Hope that we fulfilled the mentioned parameters. We will actively listen to the feedback and future enhancement will also be made based on the user's suggestions.

APPENDIX-1

//front end code created using Embedded javascript templating

```
var express = require('express');
var router = express.Router();
var StudentUser = require('../models/studentUser');
var AlumniUser = require('../models/alumniUser');
var StaffUser = require('../models/staffUser');
var passport = require('passport');
var middleware = require('../middleware');
var async = require('async');
var nodemailer = require('nodemailer');
var crypto = require('crypto');

router.get('/', function (req, res) {
  res.render('landing');
});

router.get('/login', function (req, res) {
  res.render('login');
});

router

router.post('/login', function (req, res, next) {
  const obj = JSON.parse(JSON.stringify(req.body));
  // console.log(obj);
  StudentUser.findOne({ username: obj.username }, function (err, user) {
    if (user && user.isStudent === true) {
      passport.authenticate('studentLocal', function (err, user, info) {
        if (err) {
          console.log(err);
          return next(err);
        }
        if (!user) {
          req.flash('error', 'Invalid Username or password');
          return res.redirect('/login');
        }
        req.logIn(user, function (err) {
          if (err) {
            return next(err);
          }
          var redirectTo = req.session.redirectTo
```

```

        ? req.session.redirectTo
        : '/select';
    delete req.session.redirectTo;
    req.flash('success', 'Welcome back ' + req.user.username);
    res.redirect(redirectTo);
    });
  })(req, res, next);
}
});

AlumniUser.findOne({ username: obj.username }, function (err, user) {
  if (user && user.isAlumni === true) {
    passport.authenticate('alumniLocal', function (err, user, info) {
      if (err) {
        console.log(err);
        return next(err);
      }
      if (!user) {
        req.flash('error', 'Invalid Username or password');
        return res.redirect('/login');
      }
      req.logIn(user, function (err) {
        if (err) {
          return next(err);
        }
        var redirectTo = req.session.redirectTo
        ? req.session.redirectTo
        : '/alumniChoice';
        delete req.session.redirectTo;
        req.flash('success', 'Welcome back ' + req.user.username);
        res.redirect(redirectTo);
      });
    })(req, res, next);
  }
});

StaffUser.findOne({ username: obj.username }, function (err, user) {
  if (user && user.isStaff === true) {
    passport.authenticate('staffLocal', function (err, user, info) {
      if (err) {
        console.log(err);
        return next(err);
      }
      if (!user) {
        req.flash('error', 'Invalid Username or password');
        return res.redirect('/login');
      }
      req.logIn(user, function (err) {
        if (err) {

```

```

        return next(err);
    }
    var redirectTo = req.session.redirectTo
        ? req.session.redirectTo
        : '/';
    delete req.session.redirectTo;
    req.flash('success', 'Welcome back ' + req.user.username);
    res.redirect(redirectTo);
});
})(req, res, next);
}
});
});

router.get('/register', function (req, res) {
    res.render('register');
});

router.post('/register', function (req, res) {
    const obj = JSON.parse(JSON.stringify(req.body));
    if (obj.category === 'select') {
        req.flash('error', 'Please select the option');
        res.redirect('/register');
    }
    if (obj.category === 'Alumni') {
        res.redirect('/alumniSignUp');
    }
    if (obj.category === 'Student') {
        res.redirect('studentSignUp');
    }
    if (obj.category === 'Staff') {
        res.redirect('staffSignUp');
    }
});
});

router.get('/alumniSignUp', function (req, res) {
    res.render('alumniSignUp');
});

router.get('/studentSignUp', function (req, res) {
    res.render('studentSignUp');
});

router.get('/staffSignUp', function (req, res) {
    res.render('staffSignUp');
});
});

```

```
// register logic
router.post('/studentSignUp', function (req, res) {
  const obj = JSON.parse(JSON.stringify(req.body));
  if (obj.password.length < 6) {
    req.flash('error', 'password length must be greater than 6 Characters');
    res.redirect('/studentSignUp');
  } else if (obj.password !== obj.confirmPassword) {
    req.flash('error', 'Password mismatch');
    res.redirect('/studentSignUp');
  } else {
    newStudentUser = new StudentUser({
      username: obj.username.trim(),
      Name: obj.name.trim(),
      StudentId: obj.studentId.trim(),
      Year: obj.year.trim(),
      Department: obj.dept.trim(),
      PhoneNo: obj.phoneno.trim(),
      isStudent: true,
    });
    StudentUser.findOne({ email: obj.email }, function (err, user) {
      if (!user) {
        newStudentUser.EmailId = obj.email.trim();
        async.waterfall([
          function (done) {
            StudentUser.register(
              newStudentUser,
              obj.password,
              function (err, user) {
                if (err) {
                  console.log(err);
                  req.flash('error', err.message);
                  return res.redirect('/studentSignUp');
                }
                passport.authenticate('local')(
                  req,
                  res,
                  function () {
                    req.flash(
                      'success',
                      'Successfully registered as ' +
                      newStudentUser.username
                    );
                    res.redirect('/login');
                  }
                );
              }
            );
          }
        ]);
      }
    });
  }
});
```

```

        },
        function (user, done) {
            var smtpTransport = nodemailer.createTransport({
                service: 'Gmail',
                auth: {
                    user: 'vijaiselvan2000@gmail.com',
                    pass: process.env.GMAILPW,
                },
            });
            var mailOptions = {
                to: user.email,
                from: 'vijaiselvan2000@mail.com',
                subject:
                    'WELCOME TO ALUMNI NETWORK AND CAREER GUIDANCE PORTAL'
            ,
                text:
                    'Hello,\n\n' +
                    'Thankyou ' +
                    user.username +
                    ' for signing in\n\n' +
                    'You are ready access our web portal ' +
                    'http://' +
                    req.headers.host,
            };
            smtpTransport.sendMail(mailOptions, function (err) {
                done(err);
            });
        },
    ],
]);
} else if (err || user) {
    req.flash('error', 'EmailID already exists');
    return res.redirect('/studentSignUp');
}
});
}

// newStudentUser = new StudentUser({
//     username: obj.username.trim(),
//     Name: obj.name.trim(),
//     EmailId: obj.email.trim(),
//     StudentId: obj.studentId.trim(),
//     Year: obj.year.trim(),
//     Department: obj.dept.trim(),
//     PhoneNo: obj.phoneno.trim(),
//     isStudent: true
// });

```

```

// StudentUser.register(newStudentUser, obj.password, function (err, user) {
//     if (err) {
//         console.log(err);
//         req.flash("error", err.message);
//         return res.redirect('/studentSignUp');
//     }
//     passport.authenticate('local')(req, res, function () {
//         req.flash("success", "Successfully registered as " + newStudentUser.username);
//         res.redirect('/login');
//     })
// })
// =====
});

router.post('/alumniSignUp', function (req, res) {
    const obj = JSON.parse(JSON.stringify(req.body));
    if (obj.password.length < 6) {
        req.flash('error', 'password length must be greater than 6 Characters');
        res.redirect('/alumniSignUp');
    } else if (obj.password !== obj.confirmPassword) {
        req.flash('error', 'Password mismatch');
        res.redirect('/alumniSignUp');
    } else {
        newAlumniUser = new AlumniUser({
            username: obj.username.trim(),
            Name: obj.name.trim(),
            CompanyName: obj.company.trim(),
            Designation: obj.designation.trim(),
            PassedOutYear: obj.passout.trim(),
            LinkedInID: obj.linkedIn.trim(),
            PhoneNo: obj.phoneno.trim(),
            isAlumni: true,
        });
        AlumniUser.findOne({ email: obj.email }, function (err, user) {
            if (!user) {
                newAlumniUser.EmailId = obj.email.trim();
                async.waterfall([
                    function (done) {
                        AlumniUser.register(
                            newAlumniUser,
                            obj.password,
                            function (err, user) {
                                if (err) {
                                    console.log(err);
                                    req.flash('error', err.message);
                                    return res.redirect('/alumniSignUp');
                                }
                            }
                        )
                    }
                ], done);
            }
        })
    }
})

```

```

        }
        passport.authenticate('local')(

            req,
            res,
            function () {
                req.flash(
                    'success',
                    'Successfully registered as ' +
                    newAlumniUser.username
                );
                res.redirect('/login');
            }
        );
    },
    function (user, done) {
        var smtpTransport = nodemailer.createTransport({
            service: 'Gmail',
            auth: {
                user: 'vijaiselvan2000@gmail.com',
                pass: process.env.GMAILPW,
            },
        });
        var mailOptions = {
            to: user.email,
            from: 'vijaiselvan2000@mail.com',
            subject:
                'WELCOME TO ALUMNI NETWORK AND CAREER GUIDANCE PORTAL'
        ,
            text:
                'Hello,\n\n' +
                'Thankyou ' +
                user.username +
                ' for signing in\n\n' +
                'You are ready access our web portal ' +
                'http://' +
                req.headers.host,
        };
        smtpTransport.sendMail(mailOptions, function (err) {
            done(err);
        });
    },
]);
} else if (err || user) {
    req.flash('error', 'EmailID already exists');
    return res.redirect('/alumniSignUp');
}

```

```

        }
    });

// newAlumniUser = new AlumniUser({
//     username: obj.username.trim(),
//     Name: obj.name.trim(),
//     CompanyName: obj.company.trim(),
//     Designation: obj.designation.trim(),
//     PassedOutYear: obj.passout.trim(),
//     LinkedInID: obj.linkedIn.trim(),
//     PhoneNo: obj.phoneno.trim(),
//     isAlumni: true
// });

// AlumniUser.register(newAlumniUser, obj.password, function (err, user) {
//     if (err) {
//         console.log(err);
//         req.flash("error", err.message);
//         return res.redirect('/alumniSignUp');
//     }
//     passport.authenticate('local')(req, res, function () {
//         req.flash("success", "Successfully registered as " + newAlumniUser.user
name);
//         res.redirect('/login');
//     })
// })
// =====
});

router.post('/staffSignUp', function (req, res) {
const obj = JSON.parse(JSON.stringify(req.body));
if (obj.password.length < 6) {
    req.flash('error', 'password length must be greater than 6 Characters');
    res.redirect('/staffSignUp');
} else if (obj.password !== obj.confirmPassword) {
    req.flash('error', 'Password mismatch');
    res.redirect('/staffSignUp');
} else {
    newStaffUser = new StaffUser({
        username: obj.username.trim(),
        Name: obj.name.trim(),
        StaffId: obj.staffId.trim(),
        Designation: obj.designation.trim(),
        Department: obj.dept.trim(),
        PhoneNo: obj.phoneno.trim(),
        isStaff: true,
    });
}
});
```

```

});  

StaffUser.findOne({ email: obj.email }, function (err, user) {  

  if (!user) {  

    newStaffUser.EmailId = obj.email.trim();  

    async.waterfall([  

      function (done) {  

        StaffUser.register(  

          newStaffUser,  

          obj.password,  

          function (err, user) {  

            if (err) {  

              console.log(err);  

              req.flash('error', err.message);  

              return res.redirect('/staffSignUp');  

            }  

            passport.authenticate('local')(  

              req,  

              res,  

              function () {  

                req.flash(  

                  'success',  

                  'Successfully registered as ' +  

                  newStaffUser.username  

                );  

                res.redirect('/login');  

              }  

            );  

          }  

        );  

      },  

      function (user, done) {  

        var smtpTransport = nodemailer.createTransport({  

          service: 'Gmail',  

          auth: {  

            user: 'vijaiselvan2000@gmail.com',  

            pass: process.env.GMAILPW,  

          },  

        });  

        var mailOptions = {  

          to: user.email,  

          from: 'vijaiselvan2000@mail.com',  

          subject:  

            'WELCOME TO ALUMNI NETWORK AND CAREER GUIDANCE PORTAL'  

          ,  

          text:  

            'Hello,\n\n' +  

            'Thankyou ' +

```

```

                user.username +
                ' for signing in\n\n' +
                'You are ready access our web portal ' +
                'http://' +
                req.headers.host,
            );
            smtpTransport.sendMail(mailOptions, function (err) {
                done(err);
            });
        },
    ]);
} else if (err || user) {
    req.flash('error', 'EmailID already exists');
    return res.redirect('/staffSignUp');
}
});

// newAlumniUser = new AlumniUser({
//     username: obj.username.trim(),
//     Name: obj.name.trim(),
//     CompanyName: obj.company.trim(),
//     Designation: obj.designation.trim(),
//     PassedOutYear: obj.passout.trim(),
//     LinkedInID: obj.linkedIn.trim(),
//     PhoneNo: obj.phoneno.trim(),
//     isAlumni: true
// });

// AlumniUser.register(newAlumniUser, obj.password, function (err, user) {
//     if (err) {
//         console.log(err);
//         req.flash("error", err.message);
//         return res.redirect('/alumniSignUp');
//     }
//     passport.authenticate('local')(req, res, function () {
//         req.flash("success", "Successfully registered as " + newAlumniUser.username);
//         res.redirect('/login');
//     })
// })
// =====
});

//logout
router.get('/logout', function (req, res) {
    req.flash('success', 'Successfully logged out');
}
)

```

```
req.logout();
res.redirect('/login');
});

router.get('/select', middleware.isStudentLoggedin, function (req, res) {
  res.render('select');
});

router.get('/alumniChoice', middleware.isAlumniLoggedin, function (req, res) {
  res.render('alumniChoice');
});

// USER PROFILE
router.get('/users/:id', function (req, res) {
  StudentUser.findById(req.params.id, function (err, foundUser) {
    if (err) {
      req.flash('error', 'Something went wrong.');
      res.redirect('back');
    } else {
      res.render('users/show', { user: foundUser });
    }
  })
})

// Companies
router.get('/placement', middleware.isStudentLoggedin, function (req, res) {
  res.render('placement/companies');
});

module.exports = router;
```

APPENDIX-2

//CSS STYLE SHEET

```
body {  
    /* background:url(https://sairam.edu.in/wp-content/uploads/2018/06/banner-1.jpg) no-repeat center center fixed; */  
    /* background: url(https://mapio.net/images-p/32420671.jpg) no-repeat center center fixed; */  
    background: url(https://mapio.net/images-p/63154540.jpg) no-repeat center center fixed;  
    -webkit-background-size: cover;  
    -moz-background-size: cover;  
    -o-background-size: cover;  
    background-size: cover;  
}  
  
body,  
html {  
    height: 100%;  
    width: 100%;  
    color: #fff;  
}  
  
h1 {  
    font-weight: 700;  
    font-size: 5em;  
}  
  
.center {  
    padding-top: 25%;  
    text-align: center;  
    text-shadow: 0px 4px 3px rgba(0, 0, 0, 0.2), 0px 6px 10px rgba(0, 0, 0, 0.4),  
    0px 8px 12px rgba(0, 0, 0, 0.1);  
}  
  
hr {  
    width: 400px;  
    border-top: 1px solid #f8f8f8;  
    border-bottom: 1px solid rgba(0, 0, 0, 0.2);  
    max-width: 90%;  
}  
  
/* Login page */  
.saiBg {  
    background: url(https://sairam.edu.in/wp-content/uploads/2014/04/banner41-1.jpg)
```

```
    no-repeat center center fixed;
background-size: cover;
line-height: 1.8;
}

#container {
  margin: 30px auto;
  padding: 20px;
  max-width: 400px;
}

.form-wrap {
  border-radius: 10px;
  margin-top: 75px;
  background-color: #fff;
  padding: 15px 25px;
  color: #333;
  opacity: 0.9;
}

.form-wrap h2,
.form-wrap p {
  text-align: center;
  color: #666;
}

.form-wrap .form-group {
  margin-top: 15px;
}

.form-wrap .form-group label {
  display: block;
  color: #666;
}

.form-wrap .form-group input {
  width: 100%;
  padding: 10px;
  border: 1px solid #ddd;
  border-radius: 5px;
}

.form-wrap .btn {
  display: block;
  width: 100%;
  padding: 10px;
  margin-top: 20px;
}
```

```
        color: #fff;
        background-color: #49c1a2;
        cursor: pointer;
        text-align: center;
        border-radius: 5px;
    }

.form-wrap .btn:hover {
    background-color: #37a08e;
}

.form-wrap .bottom-text {
    font-size: 13px;
    margin-top: 20px;
}

footer {
    text-align: center;
}

footer a {
    color: #49c1a2;
}

/* register */
.form-wrap .form-group select {
    width: 100%;
    padding: 10px;
    border: 1px #ddd solid;
}

.category {
    display: block;
    margin: 0 auto;
}

.signUp {
    margin-top: 20px;
}

/* Select page */
.select-page {
    color: #555;
    padding-top: 200px;
    background: #ecf0f1;
}
```

```
.image1 {
    background-image: url(https://www.myamcat.com/blog/wp-content/uploads/2020/08/Campus-Placement.png);
}

.image2 {
    background-image: url(https://sairamit.edu.in/wp-content/uploads/2018/08/Alumni20156-600x400.jpg);
    max-width: 100%;
    height: auto;
}

.image3 {
    background-image: url(https://leverageedu.com/blog/wp-content/uploads/2019/11/Higher-Studies-in-Canada.png);
    background-size: auto;
    width: 300px;
    height: auto;
}

.image {
    width: 100%;
    height: 250px;
    background-size: cover;
    background-position: center;
}

.jumbotron {
    color: #2c3e50;
    background: #ecf0f1;
}

.navbar {
    margin-bottom: 100px;
}

.txt {
    font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', Roboto, Oxygen,
    Ubuntu, Cantarell, 'Open Sans', 'Helvetica Neue', sans-serif;
    text-align: center;
    font-size: 35px;
}

.thumbnail {
    margin-top: 20px;
}
```

APPENDIX-3

//DEPENDENCIES

```
{
  "name": "alumni",
  "version": "1.0.0",
  "description": "",
  "main": "app.js",
  "dependencies": {
    "async": "^3.2.0",
    "body-parser": "^1.19.0",
    "connect-flash": "^0.1.1",
    "dotenv": "^8.2.0",
    "ejs": "^3.1.6",
    "express": "^4.16.1",
    "express-session": "^1.17.1",
    "method-override": "^3.0.0",
    "mongoose": "^5.11.18",
    "nodemailer": "^6.4.6",
    "nodemon": "^2.0.7",
    "passport": "^0.4.1",
    "passport-local": "^1.0.0",
    "passport-local-mongoose": "^6.0.1"
  },
  "devDependencies": {},
  "scripts": {
    "test": "echo \\\"Error: no test specified\\\" && exit 1"
  },
  "author": "",
  "license": "ISC"
}
```

APPENDIX-4

//Nodejs and mongo DB setup for backend

```
var express = require('express'),
    app = express(),
    bodyParser = require('body-parser'),
    mongoose = require('mongoose'),
    passport = require('passport'),
    LocalStrategy = require('passport-local'),
    StudentUser = require('./models/studentUser'),
    AlumniUser = require('./models/alumniUser'),
    StaffUser=require('./models/staffUser'),
    methodOverride = require('method-override'),
    flash = require('connect-flash');

app.use(methodOverride('_method'));

mongoose.connect('mongodb://localhost/alumni', {
  useNewUrlParser: true,
  useUnifiedTopology: true,
  useFindAndModify: false,
  useCreateIndex: true,
});

var indexRoutes = require('./routes/index');

app.use(bodyParser.urlencoded({ extended: false }));
app.set('view engine', 'ejs');
app.use(express.static('public'));
app.use(flash());

//PASSPORT configuration
app.use(require('express-session')({
  secret: 'session',
  resave: false,
  saveUninitialized: false
}));

app.use(passport.initialize());
app.use(passport.session());

passport.use('studentLocal', new LocalStrategy(StudentUser.authenticate()));
```

```
passport.use('alumniLocal', new LocalStrategy(AlumniUser.authenticate()));
passport.use('staffLocal', new LocalStrategy(StaffUser.authenticate()));
passport.serializeUser(function(user, done) {
  done(null, user);
});

passport.deserializeUser(function(user, done) {
  if(user!=null)
    done(null,u
ser);
});

app.use(function (req, res, next) {
  res.locals.currentUser = req.user;
  res.locals.error = req.flash("error");
  res.locals.success = req.flash("success");
  next();
})

app.use(indexRoutes);

var port = 3000;
app.listen(port, function () {
  console.log('Alumni sever has started');
});
```

REFERENCES

1. *M. Qamhieh et al.: "PCRS for Engineering Students", vol.8, December 10, 2020.*
2. *Amirthavalli P et al, "OPTIMIZED ALUMNI DATA MANAGEMENT SYSTEM", Vol.6 Issue.3, March- 2017.*
3. *Akeem Olowolayemo, "University based Job Recommender & Alumni System", 2018*
4. *Ashmy Achu Shinu, "Customized Alumni Portal implementing Natural Language Processing Techniques", 2019*

