PLACEMENT DRIVE WEB APPLICATION

A PROJECT REPORT

Submitted by

L KOWSALYA

Reg. No: 21800020

Under the guidance of

Mr. R. Senthil Kumar MCA., ME., (Ph.D)

Assistant Professor(Sel. Gr)

in

DEPARTMENT OF COMPUTER APPLICATIONS

in partial fulfillment for the award of the degree

of

MASTER OF COMPUTER APPLICATIONS

PONDICHERRY UNIVERSITY

DEPARTMENT OF COMPUTER APPLICATIONS

RAJIV GANDHI COLLEGE OF ENGINEERING AND TECHNOLOGY

PONDICHERRY-607 402

FEBRUARY-2023

RAJIV GANDHI COLLEGE OF ENGINEERING AND TECHNOLOGY

PONDICHERRY-607 402

DEPARTMENT OF COMPUTER APPLICATIONS

BONAFIDE CERTIFICATE

Certified that this project report **"PLACEMENT DRIVE WEB APPLICATION"** is the bonafide work of "**L KOWSALYA" (Reg.No.21800020)** in partial fulfillment of Post Graduate degree of "**MASTER OF COMPUTER APPLICATIONS**" in Rajiv Gandhi College of Engineering and Technology during the year 2022- 2023, who carried out the project work under my supervision.

Staff In charge **Head of the Department**

Submitted for the University Examination held on ----------

Internal Examiner **External Examiner**

TABLE OF CONTENTS

CHAPTER NO	TITLE	PAGE NO
	ACKNOWLEDGEMENT	**I**
	ABSTRACT	**II**
	LIST OF FIGURES	**III**
1	**INTRODUCTION**	
	1.1 ABOUT THE PROJECT	1
2	**SYSTEM REQUIREMENTS**	
	2.1 HARDWARE REQUIREMENTS	2
	2.2 SOFTWARE REQUIREMENTS	2
	2.3 TECHNOLOGIES USED	3
	2.4 DATABASE	5
3	**ABOUT THE SOFTWARE**	
	3.1 WEB APPLICATION	8
	3.2 JAVA SCRIPT	9
	3.3 FIREBASE REAL TIME DATABASE	10
4	**PROBLEM DEFINITION AND PROPOSAL**	
	4.1 INTRODUCTION	11
	4.2 EXISTING SYSTEM AND ITS DRAWBACKS	11
	4.3 PROPOSED SYSTEM	12
	4.4 FEASIBLITY STUDY	12
	4.4.1 Operational Feasibility	13
	4.4.2 Technical Feasibility	13
	4.4.3 Economical Feasibility	13

5 SYSTEM DESIGN

5.1 INTRODUCTION 14
5.2 COMPONENT DESCRIPTION 14
5.3 MODULES DESCRIPTION 14
5.3.1 Admin 14
5.3.2 Student 15
5.3.3 Company 16
5.4 UML DIAGRAM 16
5.4.1 Use Case Diagram 16
5.4.2 State Chart 18
5.4.3 Sequence Diagram 20
5.5 DATA DICTIONARY 21

6 IMPLEMETATION

6.1 INTRODUCTION 24
6.2 FORMS USED 24
6.2.1 Student Side 24
6.2.2 Admin Side 24
6.2.3 Company Side 25

7 TESTING

7.1 INTRODUCTION 26
7.2 TESTING STRATEGIES 26
7.3 TEST PROCEDURE 27
7.3.1 System Testing 27
7.4 TEST CASE AND OUTPUT 27
7.4.1 Unit Testing 27
7.4.2 Functional Tests 28
7.4.3 Performance Test 28
7.4.4 Stress Test 28
7.4.5 Structured Test 29
7.5 SOFTWARE TESTING STRATEGIES 29
7.6 INTEGRATION TESTING 29

8 CONCLUSION AND FUTURE ENHANCEMENT

8.1 CONCLUSION 30

8.2 FUTURE ENHANCEMENT 30

APPENDIX

APENDIX I : SOURCE CODE 31

APENDIX II : SCREENSHOTS 53

REFERENCES 70

ACKNOWLEDGEMENT

With great gratitude I would like to thank **Shri. M. K. RAJAGOPALAN,** *Chairman,* Rajiv Gandhi College of Engineering and Technology, for producing highly efficient and potentially resourceful student community to meet their challenges and exploit opportunities in the new millennium.

My sincere thanks are due to **Dr. E. VIJAYAKRISHNA RAPAKA,** *Principal,* Rajiv Gandhi College of Engineering and Technology, for his invaluable and generously given support and encouragement to me throughout my course.

I wish to express my profound thanks to **Shri. V. BHASKARAN,** *Administrator,* Rajiv Gandhi College of Engineering and Technology, for his kind support and advice he had offered throughout my project work.

I mention my heartiest thanks to the respected **Head of the Department** of Computer Applications **Mr. R. SENTHIL KUMAR, M.C.A., M.E., (Ph.D).,** Assistant Professor(Sl.Gr) & HOD, Rajiv Gandhi College of Engineering and Technology . I am indeed highly grateful and immensely indebted for all his guidance in updating my progress and fine-tuning of this project.

I also wish to place on record my thanks to all my **teaching and non-teaching staffs** in the Department of Computer Applications, Rajiv Gandhi College of Engineering and Technology for the keen interest and guidance shown by them throughout the project.

KOWSALYA L

ABSTRACT

The project entitled **“PLACEMENT DRIVE WEB APPLICATION”** is an online platform designed to streamline and simplify the campus recruitment process. The application features a user-friendly interface that allows students to upload their resumes and apply for jobs with ease. This system can be used as an application for the Placement Officers in the college to manage the student information with regard to placement. And it also be used as an platform for Companies and College students for recruitment process and Job Opportunities. The placement drive web application is intended to enhance the efficiency and effectiveness of the campus recruitment process and facilitate better job placement for students.

The key feature of this Project is that it is one time Registration enabled. Our project provides the facility of maintaining the details of the Students. It reduces the manual work and consumes less paper work to reduce the time. This web based placement management system is proposed, where the student information in the college with regard to placement is managed efficiently. It intends to help fast in fast access procedures in placement related activities and ensures to maintain the details of the student. The placement cell allows the companies to view the student resumes in selective manner.

The Admin can also view feedback and queries from company and student side to take possible actions. Placement Drive web Application enables you to plan placements, train students in accordance with relevant market demands, organize placement drive, and allow companies to recruit students.

LIST OF FIGURES

S.NO	NAME OF THE DIAGRAM	PAGE NO
1	USE CASE DIAGRAM	17
2	STATE CHART DIAGRAM	18
3	SEQUENCE DIAGRAM	20

CHAPTER 1

INTRODUCTION

1.1 ABOUT THE PROJECT

The key feature of this Project is that it is one time Registration enabled. Our project provides the facility of maintaining the details of the Students. It reduces the manual work and consumes less paper work to reduce the time. This web based placement management system is proposed, where the students information in the college with regard to placement is managed efficiently. It intends to help fast in fast access procedures in placement related activities and ensures to maintain the details of the student. The placement cell allows the companies to view the student resumes in selective manner.

The platform chosen for this kind of system is web, reason being Web Application System has come up on a very large scale and is owned by almost every second person. Also, Web Application is a user friendly platform, thereby enabling ease of access for all the users. A number of applications made for the Web Application System is increasing on a large scale ever since its advent. Web is an open source mobile & computerised software environment.

A Web application (Web app) is an application program that is stored on a remote server and delivered over the Internet through a browser interface. Web services are Web apps by definition and many, although not all, websites contain Web apps. If we talk about the web application in general, a web application usually uses a combination of the server-side scripts such as Fire Base**,** for handling the information data storage and retrieval of the data.

Some of them also use the client-side scripts such as Java Script**,** HTML to represent the data/information in front of the users, and some of the web applications are also using both server-side and client-side at the same time.

CHAPTER 2

SYSTEM REQUIREMENTS

2.1 HARDWARE REQUIREMENT

- CPU type : RYZEN 5 Processor
- Clock speed : 3.0 GHz
- RAM : 8GB
- Hard disk capacity : 500 GB
- Monitor type : 14 Inch color monitor
- Input Device : Standard Keyboard and Mouse
- Output Device : Monitor
- Compact Disk : 1 GB

2.2 SOFTWARE REQUIREMENTS

- Operating System : Windows 10
- IDE : Vs Code
- Plug-in : Bootstrap 4.4
- Back-End : Fire Base (Cloud)
- Front-End : HTML, CSS, JavaScript

2.3 TECHNOLOGY

2.3.1 Front-end Design: HTML, CSS, Bootstrap

Hypertext Markup Language (HTML) is the standard markup language for creating web pages and web applications. With Cascading Style Sheets (CSS) and JavaScript, it forms a triad of cornerstone technologies for the World Wide Web. Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language like 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, colors, and fonts.

Bootstrap is a free and open-source front-end library for designing websites and web applications. It contains HTML- and CSS-based design templates for typography, forms, buttons, navigation and other interface components, as well as optional JavaScript extensions. Unlike many web frameworks, it concerns itself with front-end development only.

2.3.2 JavaScript

JavaScript is a dynamic computer programming language. It is lightweight and most commonly used as a part of web pages, whose implementations allow client-side script to interact with the user and make dynamic pages. It is an interpreted programming language with object-oriented capabilities.

JavaScript was first known as Live Script**,** but Netscape changed its name to JavaScript, possibly because of the excitement being generated by Java. JavaScript made its first appearance in Netscape 2.0 in 1995 with the name Live Script

The ECMA-262 Specification defined a standard version of the core JavaScript language.

- JavaScript is a lightweight, interpreted programming language.
- Designed for creating network-centric applications.
- Complementary to and integrated with Java.
- Complementary to and integrated with HTML.
- Open and cross-platform

Client-Side JavaScript

Client-side JavaScript is the most common form of the language. The script should be included in or referenced by an HTML document for the code to be interpreted by the browser.

It means that a web page need not be a static HTML, but can include programs that interact with the user, control the browser, and dynamically create HTML content.

The JavaScript client-side mechanism provides many advantages over traditional CGI server-side scripts. For example, you might use JavaScript to check if the user has entered a valid e-mail address in a form field.

The JavaScript code is executed when the user submits the form, and only if all the entries are valid, they would be submitted to the Web Server.

JavaScript can be used to trap user-initiated events such as button clicks, link navigation, and other actions that the user initiates explicitly or implicitly.

Advantages of JavaScript

The merits of using JavaScript are −

- **Less server interaction** − You can validate user input before sending the page off to the server. This saves server traffic, which means less load on your server.
- **Immediate feedback to the visitors** − They don't have to wait for a page reload to see if they have forgotten to enter something.
- **Increased interactivity** − You can create interfaces that react when the user hovers over them with a mouse or activates them via the keyboard.
- **Richer interfaces** − You can use JavaScript to include such items as drag-and-drop components and sliders to give a Rich Interface to your site visitors

Limitations of JavaScript

We cannot treat JavaScript as a full-fledged programming language. It lacks the following important features −

- Client-side JavaScript does not allow the reading or writing of files. This has been kept for security reason.
- JavaScript cannot be used for networking applications because there is no such support available.

- JavaScript doesn't have any multi-threading or multiprocessor capabilities.

Once again, JavaScript is a lightweight, interpreted programming language that allows you to build interactivity into otherwise static HTML pages.

JavaScript Development Tools

One of major strengths of JavaScript is that it does not require expensive development tools. You can start with a simple text editor such as Notepad. Since it is an interpreted language inside the context of a web browser, you don't even need to buy a compiler.

To make our life simpler, various vendors have come up with very nice JavaScript editing tools. Some of them are listed here −

- **Microsoft FrontPage** − Microsoft has developed a popular HTML editor called FrontPage. FrontPage also provides web developers with a number of JavaScript tools to assist in the creation of interactive websites.
- **Macromedia Dreamweaver MX** − Macromedia Dreamweaver MX is a very popular HTML and JavaScript editor in the professional web development crowd. It provides several handy prebuilt JavaScript components, integrates well with databases, and conforms to new standards such as XHTML and XML.
- **Macromedia HomeSite 5** − HomeSite 5 is a well-liked HTML and JavaScript editor from Macromedia that can be used to manage personal websites effectively.

2.4 DATABASE:

2.4.1 Firebase

The Firebase Realtime Database is a cloud-hosted database in which data is stored as JSON. The data is synchronized in real-time to every connected client. All of our clients share one Realtime Database instances and automatically receive updates with the newest data, when we build cross-platform applications with our iOS, and JavaScript SDKs.

The Firebase Realtime Database is a NoSQL database from which we can store and sync the data between our users in real-time. It is a big JSON object which the

developers can manage in real-time. By using a single API, the Firebase database provides the application with the current value of the data and updates to that data. Real-time syncing makes it easy for our users to access their data from any device, be it web or mobile.

The Realtime database helps our users collaborate with one another. It ships with mobile and web SDKs, which allow us to build our app without the need for servers. When our users go offline, the Real-time Database SDKs use local cache on the device for serving and storing changes. The local data is automatically synchronized, when the device comes online. Firebase Authentication **aims to make building secure authentication systems easy, while improving the sign-in and onboarding experience for end users**. It provides an end-to-end identity solution, supporting email and password accounts, phone auth, and Google, Twitter, Facebook, and GitHub login, and more.

Key capabilities

A Real-time database is capable of providing all offline and online services. These capabilities include accessibility from the client device, scaling across multiple databases, and many more

Real-time

The Firebase Real-time database uses data synchronization instead of using HTTP requests. Any connected device receives the updates within milliseconds. It doesn't think about network code and provides collaborative and immersive experiences.

Offline

The Firebase Database SDK persists our data to disk, and for this reason, Firebase apps remain responsive even when offline. The client device receives the missed changes, once connectivity is re-established.

Accessible from client devices

There is no need for an application server to access the Firebase Real-time database. We can access it directly from a mobile device or web browser. Data validation and security are available through the Firebase Real-time Database Security Rules, expression-based rules executed when data is read or written.

Scaling across multiple databases

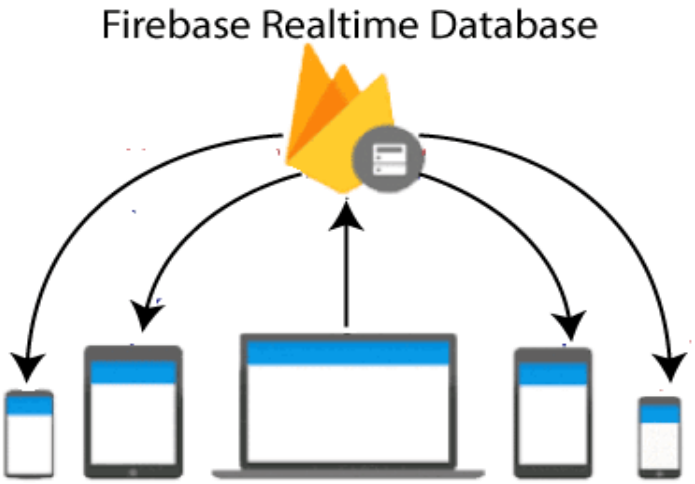
With the Firebase Real-time Database on Blaze Pricing Plan, we can support the data needs of our app by splitting our data across multiple database instances in a single Firebase project. Streamline authentication with Firebase authentication on our project and authenticate users in our database instances.

Other Alternatives

Apart from Firebase's real-time database, there are several alternatives that are used.

Cloud Firestore

Cloud Firestore is a scalable and flexible database used for server development, mobile, and web from Firebase and Google Cloud Platform.



Firebase Remote Config

It stores developer specified key-value pairs to change the behavior and appearance of our app without requiring users to download an update.

Firebase Hosting

It is used to hosts the HTML, CSS, and JavaScript of our website as well as other developer-provided assets like graphing, fonts, and icons.

Cloud Storage

It is used to store images, videos, and audio as well as other user-generated content.

CHAPTER 3

ABOUT THE SOFTWARE

3.1 WEB APPLICATION

It is a type of computer program that usually runs with the help of a web browser and also uses many web technologies to perform various tasks on the internet.

A web application can be developed for several uses, which can be used by anyone like it can be used as an individual or as a whole organization for several reasons.

If we talk about the web application in general, the client-side scripts such as **JAVA SCRIPT, HTML** to represent the data/information in front of the users, and some of the web applications are also using both **server-side** and **client-side** at the same time.

Apart from that web applications also allow its users to create documents, share them, or share the data/ information. By using the web application, users can collaborate on same projects by event when they are not available on the same geographical location.

THE FLOW OF THE WEB APPLICATION:

1. In general, a user sends a request to the web-server using web browsers such as **Google Chrome, Microsoft Edge, Firefox**, etc over the **internet**.
2. Then, the request is forwarded to the appropriate web **application server** by the **web-server**.
3. Web application server performs the requested operations/ tasks like **processing the database, querying the databases; produces** the result of the requested data.
4. The obtained result is sent to the web-server by the web application server along with the requested data/information or processed data.
5. The web server responds to the user with the requested or processed data/information and provides the result to the user's screen.

3.2 JAVASCRIPT:

JavaScript is a lightweight, cross-platform, and interpreted compiled programming language which is also known as the scripting language for webpages. It is well-known for the development of web pages, many non-browser environments also use it. JavaScript can be used for Client-side developments as well as Server-Side developments. Javascript is both imperative and declarative type of language. JavaScript contains a standard library of objects, like Array,Date, and Math, and a core set of language elements like operatio, **control structures**, and Statement.

- **Client-side:** It supplies objects to control a browser and its Document Object Model (DOM) Like if client-side extensions allow an application to place elements on an HTML form and respond to user events such as **mouse clicks**, **form input**, and **page navigation**.
- **Server-side:** It supplies objects relevant to running JavaScript on a server. Like if the server-side extensions allow an application to communicate with a database, and provide continuity of information from one invocation to another of the application, or perform file manipulations on a server. The useful framework which is the most famous these days is node.js
- **Imperative language –** In this type of language we are mostly concern about how it is to be done . It simply control the flow of computation . The procedural programming approach , object, oriented approach comes under this like async await we are thinking what it is to be done further after async call.
- **Declarative programming –** In this type of language we are concern about how it is to be done , basically here logical computation require . Here main goal is to describe the desired result without direct dictation on how to get it like arrow function do .

3.3 FIREBASE REAL TIME DATABASE:

During the past few decades, the relational database (RDBMS) SQL has been the primary model for database management. As an alternative model for database management, the non-relational "NoSQL" databases first used in 1998 by Carlo Strozzi has been gaining steam.

The need for processing unstructured data and the need for faster processing gave rise to NoSQL. The non-relational system is quicker, uses an ad-hoc approach for organizing data, and processes large amounts of different kinds of data. For large, unstructured data, NoSQL databases are better compared with relational databases due to their speed and flexibility.

The Firebase Realtime Database is a NoSQL cloud-hosted database. Data is stored as JSON and synchronized in Realtime to every connected client. Firebase Realtime Database is perfect when you build cross-platform apps. All clients share one Realtime Database instance and automatically receive updates with the newest data.

CHAPTER 4

PROBLEM DEFINITION AND PROPOSAL

4.1 INTRODUCTION

Students choose a specific college where the placement will be held, there is a need to maintain all these papers, causing large amount of space. It is manually done, changes of missing, difficult to handle the details of students.The last two decades have seen growing interest in the development of Web based platform. So, I Developed the web-based project.

4.2 EXISTING SYSTEM AND ITS DRAWBACKS:

The earlier system is not computerized. All transactions in the system are done manually maintaining records. To make this laborious job simple the clients have to computerize the system. The management and all the departments that have been carrying out this job using manually makes the job more complicated and tedious most of the times. So, the best way is computerize computerization of the current environment.

For example, in the earlier system placement officer has to collect student details for placements. Approving those student details takes lot of time. Placement officer and students have to consult each other directly if any information is needed. If any new company come for placements, placement officer and his staff has to search the student details and they have to find the eligible candidates for that particular company placement. Here searching for eligible candidates takes lots of time. And some times some candidates' details may be missed.

DRAWBACKS OF EXISTING SYSTEM:

- It takes so much time for a placement officer to collect students' details and approving the details provided by them.
- Poor communication between students and placement officer, so here intimating about new placements is a hard task.
- Students may not know about company details.
- Here also poor communication provides a problem.

4.3 PROPOSED SYSTEM

Proposed system is an online application that can be accessed throughout the organization and outside as well with proper login provided. Students logging should be able to upload their information in the form of a CV. All the users have some common services like updating details, searching for details, checking the details, querying to the administrator. Companies has to do the services like posting drives. They can upload results for selected candidates, search for student details.

ADVANTAGE OF THE PROPOSED SYSTEM:

- Placement officer can easily collect student' details, and approve the details provided by them.
- As it is an online application, communication with placement officer is easy to students and recruiters, so here intimating about new placements very easy task. Here recruiters can also search for the details provided by students on the basis of their percentage.
- Placement officer can send required materials used for placements preparation to students.
- With this option preparation for placements becomes easy.

4.4 FESIBLITY STUDY

The purpose of the feasibility study is not to solve the problem, but to determine the problem is worth solving. This helps to decide whether to proceed with the problem or not. It involves the analysis of the problem & collection of all relevant information relating to the product such as items that would be input to the system, processing required to carried those data, the output data required to be produced by the system as well the various constraints on the behavior of the system.

"Web based placement management system" had undergone the feasibility study so that the proposed system is possible for development deployment in our college.

Facts considered in the feasibility analysis were-

- Operational Feasibility
- Technical Feasibility
- Economical Feasibility

4.4.1 Operational Feasibility

This analysis involves how it will work when it is installed and the assessment of political and managerial environment in which it is implemented. People are inherently resistant to change and computers have been known to facilitate change. The new proposed system is very much useful to the users and therefore it will accept broad audience from around the world.

4.4.2 Technical Feasibility

Technical feasibility includes whether the technology is available in the market for development and its availability. The assessment of technical feasibility must be based on an outline design of system requirements in terms of input, output, files, programs and procedures.

This can be qualified in terms of volumes of data, trends, frequency of updating, cycles of activity etc., in order to give an introduction of technical system.

4.4.3 Economical Feasibility

This feasibility study present tangible and intangible benefits from the project by comparing the development and operational cost. The technique of cost benefit analysis is often used as a basis for assessing economic feasibility. This system needs some more initial investment than the existing system, but it can be justifiable that it will improve quality of service.

CHAPTER 5

SYSTEM DESIGN

5.1 INTRODUCTION

This chapter describes features, fragments, classes, architecture and the application itself by providing necessary information of major components. First, an overall information is given along with project's components and classes. Subsequently, the architecture details of the application is discussed**.** This Application start with instantiating Placement Officer. This is needed to maintain the student details and recruiting process. Detailed description regarding the interview is provided by the company in this section. Next, UI and user interaction handling sets up all necessary selections.

5.2 COMPONENT DESCRIPTION

In order to provide a detailed view concerning system mechanism, project can be grouped in three segments. Those are admin panel, company panel and student panel.

The admin panel is maintain all the information about students and allow student and company interact eachother. It has a separate dashbord is there the admin is work on it.The user panel is apply the drive whatever they want and attend the interview to get placed.

5.3 MODULES DESCRIPTION

5.3.1 Admin DashBoard

In admin dash board they will login with this application if you create a new admin register the new mail id & password on it and maintain details of students

Connections

In Connections tab, admin can view all registered companies in the web portal. The details includes company name, email, contact no,HR-name. From this admin maintain the companies that tied up with their college for recruiting students.

Drives

In the drive tab, there are four sub tabs, Active drives, Results, Applications. From the active drives, admin can view the overall drives details including number of rounds, time, venue, package, position, job type, job role, location etc. posted for interview.

From the results tab, admin can maintain all the selected candidate details in the website and fetch the student details whenever needed.

From the application tab, admin can view the total applications for drives and those are identified with particular drive id. Admin can download the student resumes as well

.Students

In the students tab, admin can maintain all the registered candidates details in the website and fetch the students details whenever needed.

Results

Here admin can view all the details of placed students with department, year, selected on which company etc.

5.3.2 Student Dashboard

In student dash board they will login with this application if you create a new admin register the new mail id & password on it and apply for interview and get placed in companies.

Connections

In the connections tab, students can view the registered or tied up companies with their college for recruitment.

Active Drives

In the active drives tab, students can view the Drive post and drive post details includes job type, job role, location, package, interview venue, time, no of rounds and its details etc. and apply for that particular drive with resume.

Results

In the results tab, students can enter their name and get results if they got selected in the applied drive.

Profile

Here students can add academic details by their own and view the profile after updated.

5.3.3 Company Dashboard

In company dash board they will login with this application if you create a new admin register the new mail id & password on it and post drive for interview to recruit students.

Drive Post

In the post tab, companies can update upcoming drives for interview with required detailes.

Applications

In applications tab, companies can view the job applications applied by students and download the required student’s resume to know the details and validate for recruitment process.

Results

In the results tab, companies can view , update, add, edit the results of the recruitment process results with their drive id which helps to identify the particular drive post.

Profile

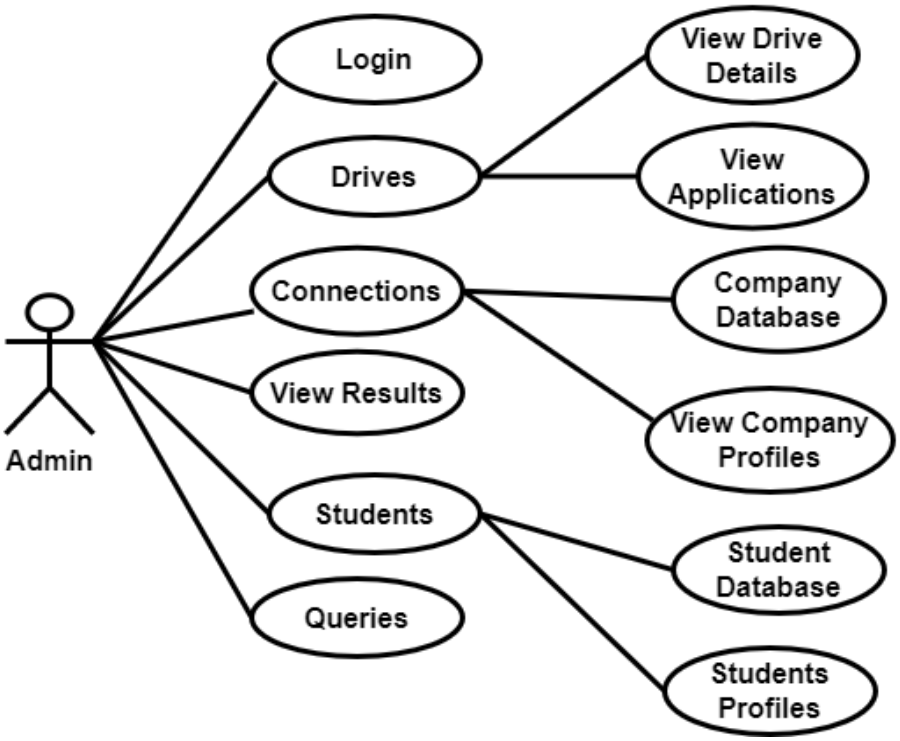
Here company co-ordinator can add the details of the company and history, founder, company goals and link their linkedin profile ID and website link which helps students and placement co-ordinator to know the company details in quick time.

5.4 UML Diagram

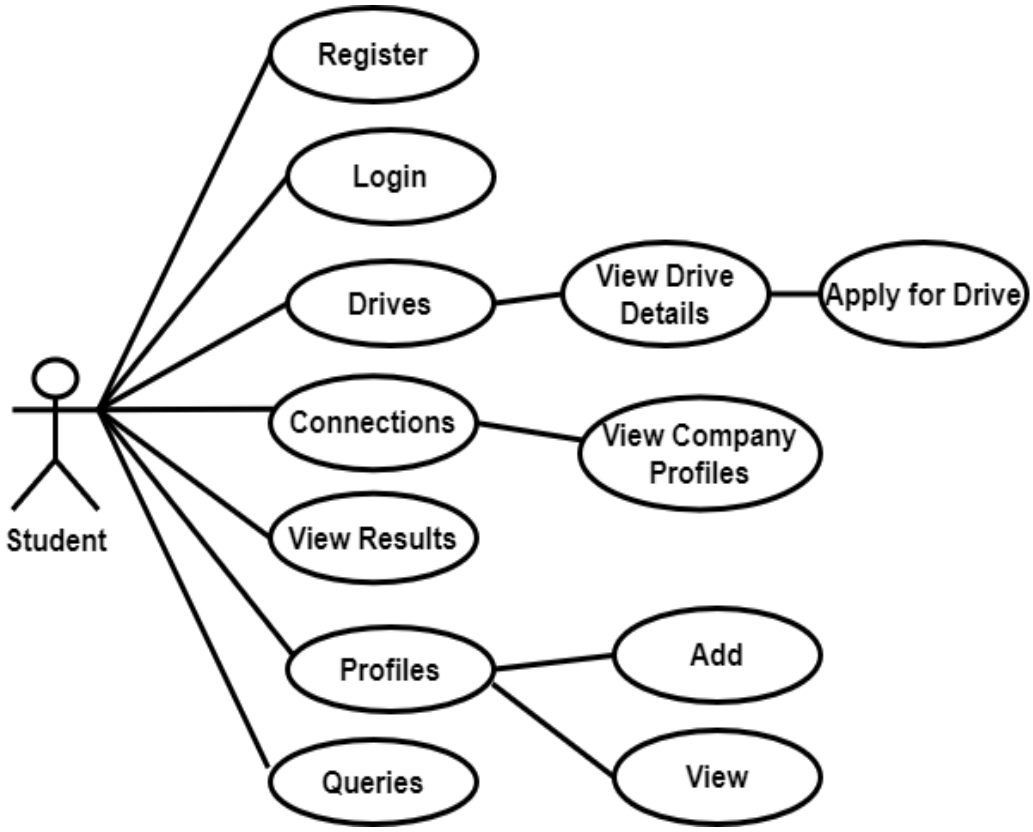
5.4.1 Use Case Diagram

A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved. A use case diagram can identify the different types of users of a system and the different use cases and will often be accompanied by other types of diagrams as well.

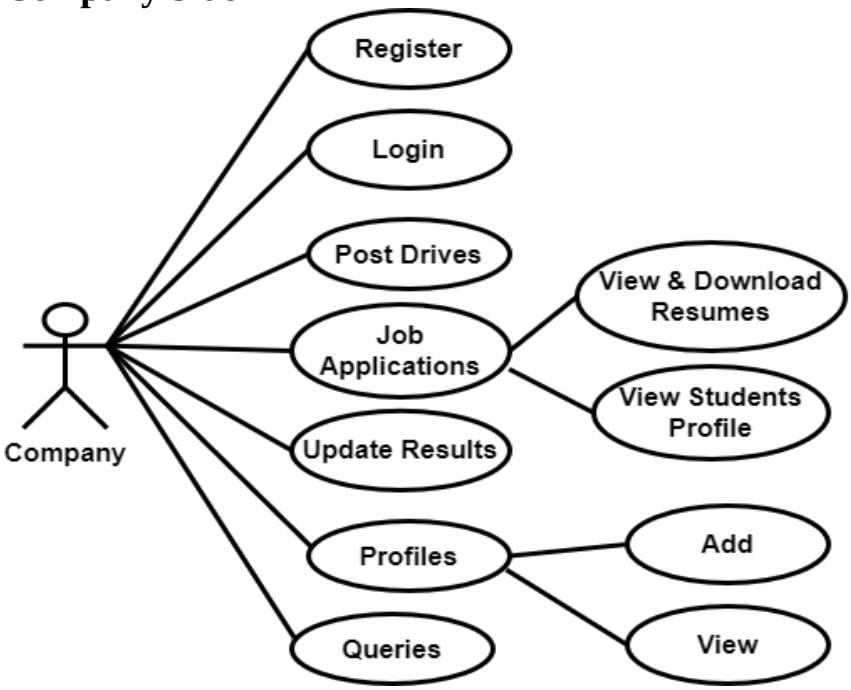
Admin Side



Student Side



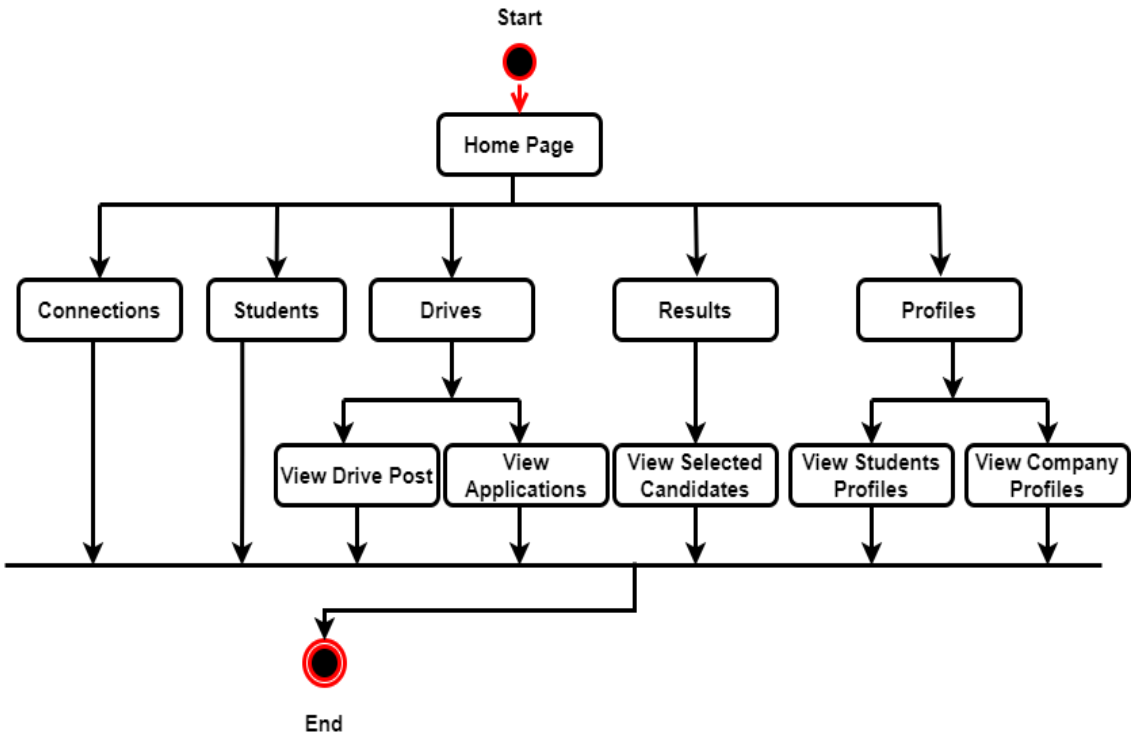
Company Side



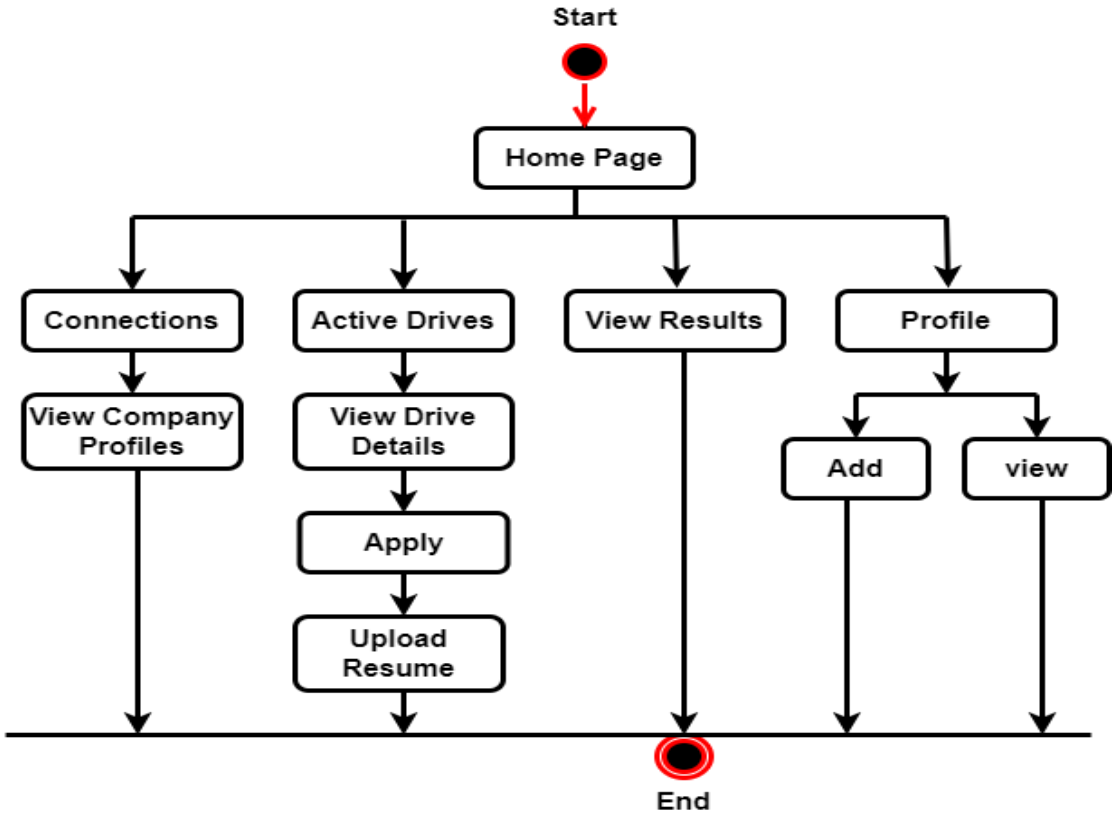
5.4.2 State Chart

A state diagram is a type of diagram used in computer science and related fields to describe the behavior of systems. State diagrams require that the system described is composed of a finite number of states; sometimes, this is indeed the case, while at other times this is a reasonable abstraction

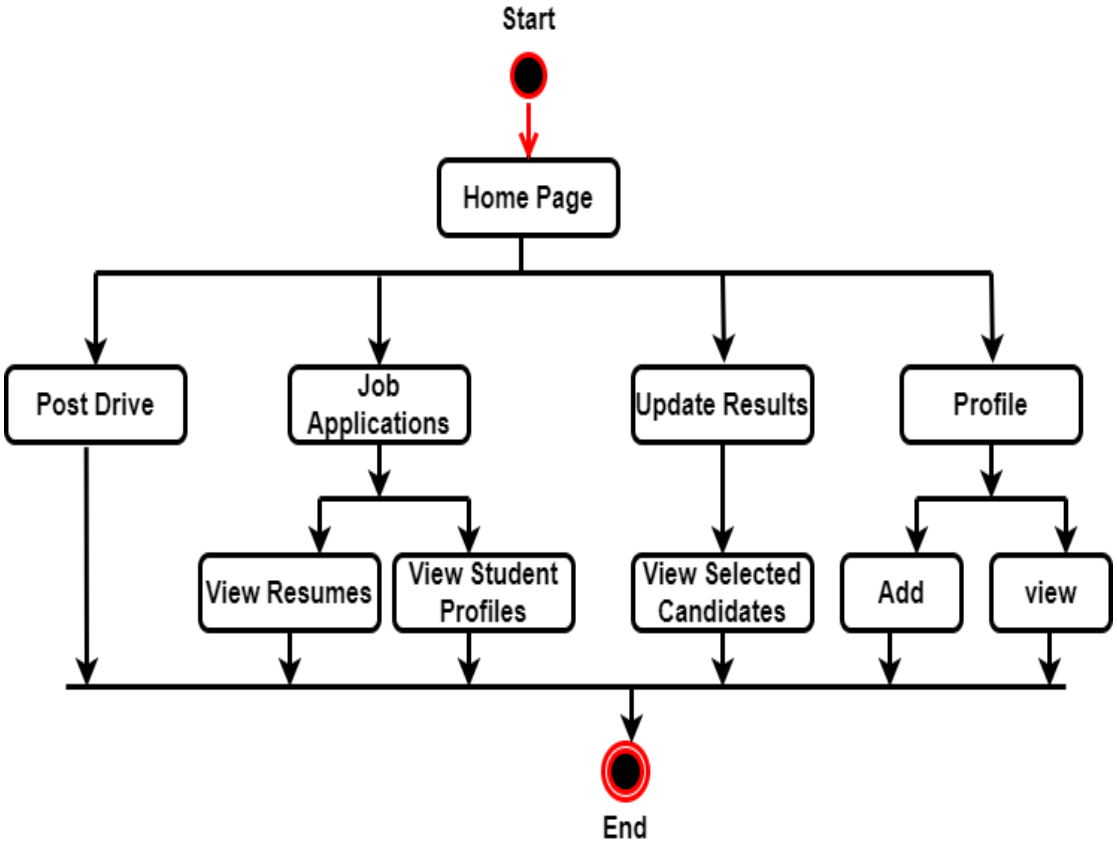
Admin side



Student Side

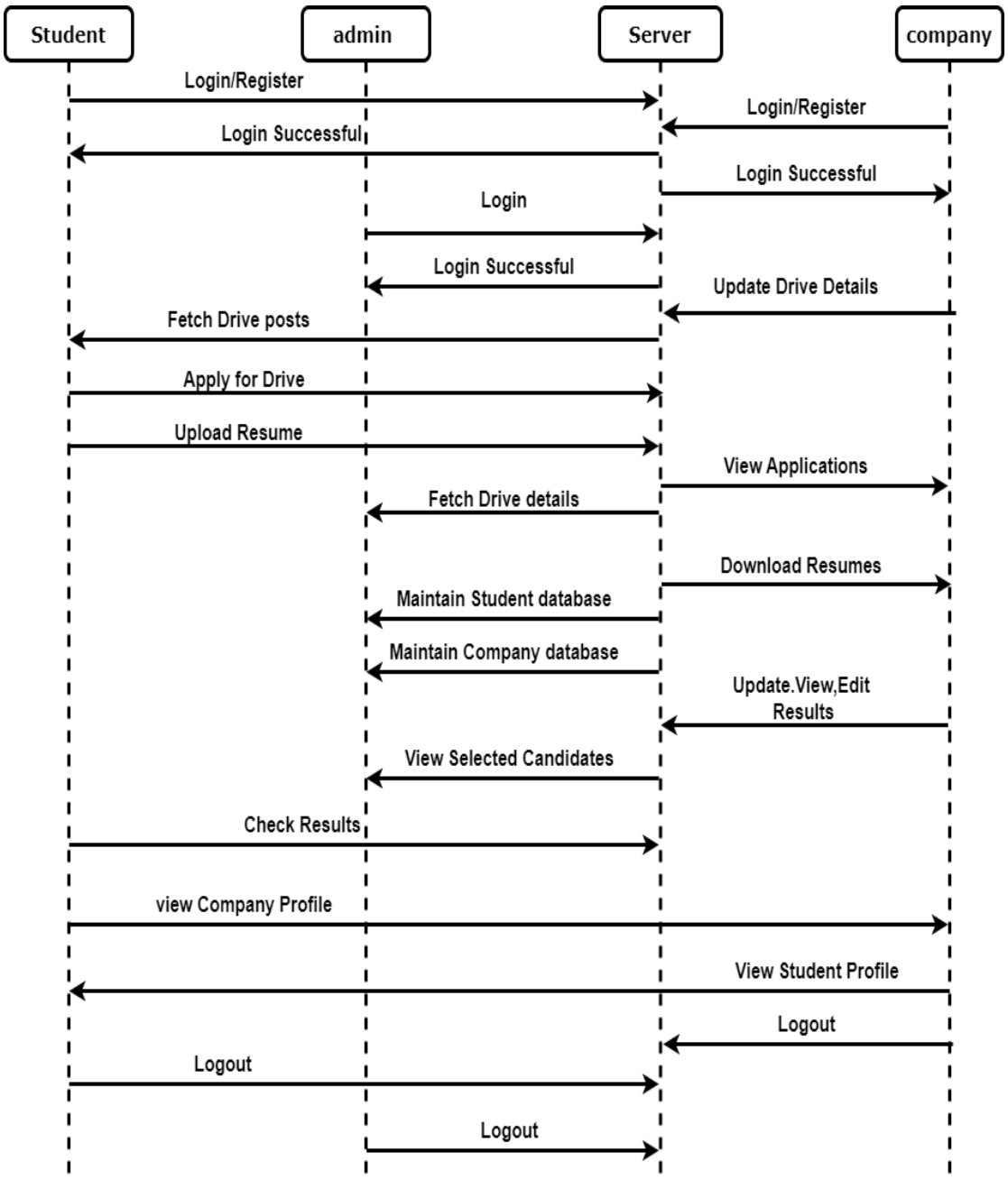


Company Side



5.4.3 SEQUENCE DIAGRAM

The sequence diagram represents the flow of messages in the system and is also termed as an event diagram. It helps in envisioning several dynamic scenarios. It portrays the communication between any two lifelines as a time-ordered sequence of events, such that these lifelines took part at the run time. In UML, the lifeline is represented by a vertical bar, whereas the message flow is represented by a vertical dotted line that extends across the bottom of the page. It incorporates the iterations as well as branching.

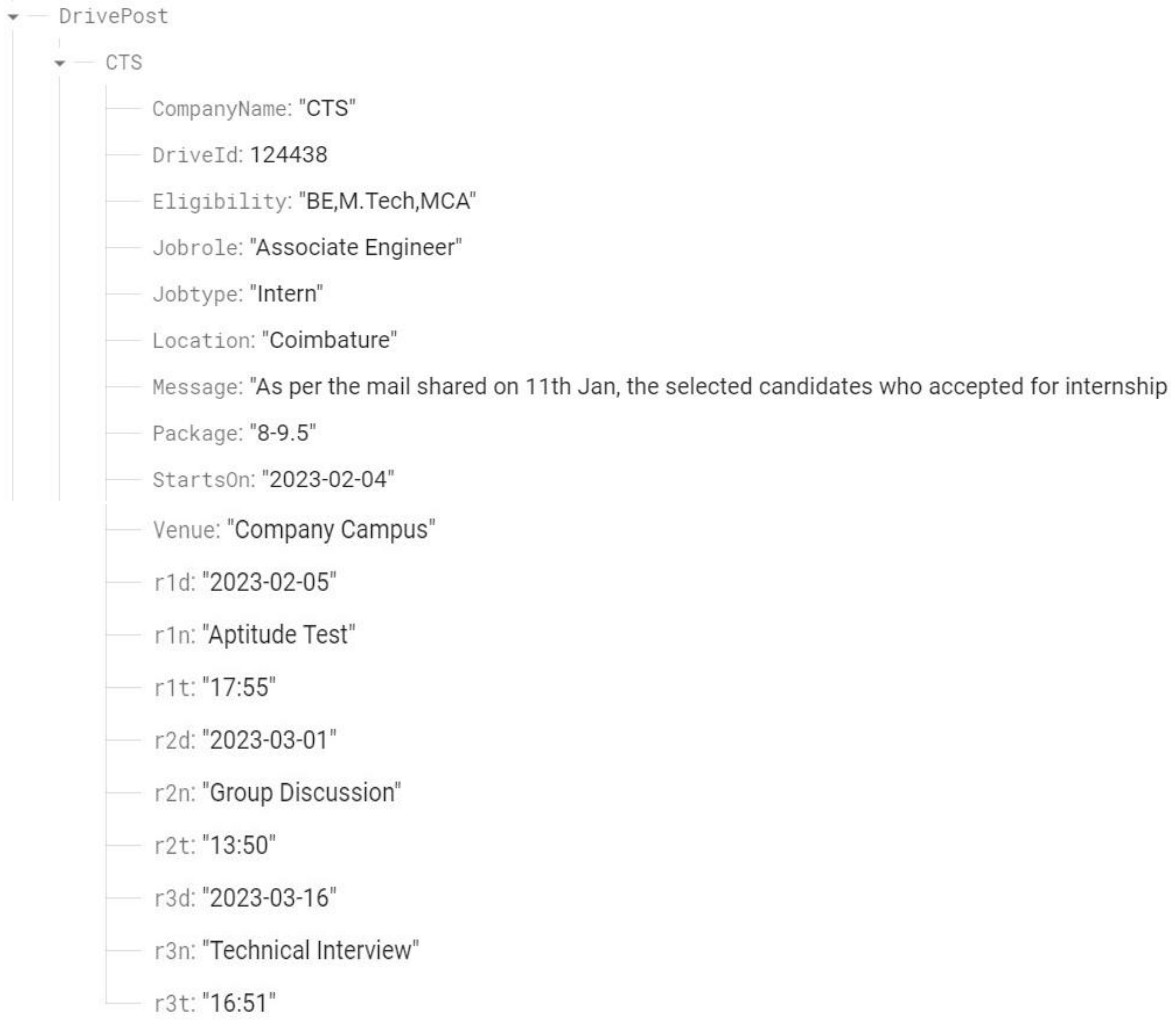


5.5 DATA DICTIONARY

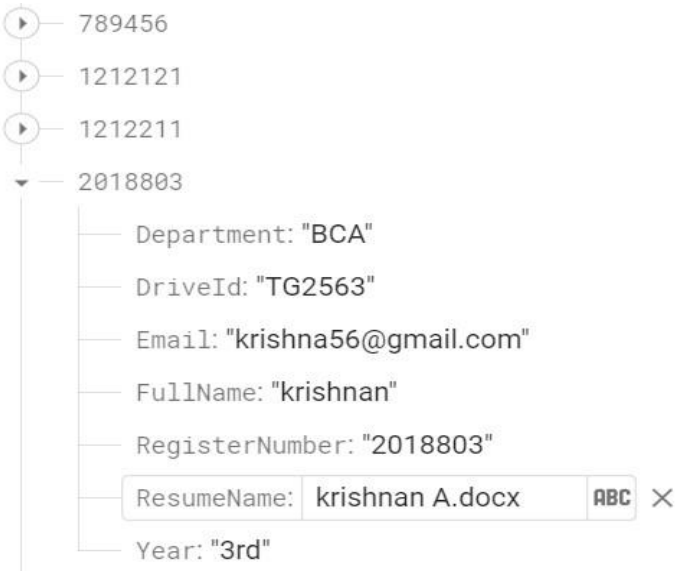
Student Details

Company Details

Drive Details



Job Applications Details



Company Profile Details

Student Profile Details

CHAPTER 6

IMPLEMENTATION

6.1 INTRODUCTION

Implementation is the execution of the plan, or implement the idea or model, or design to do something. Also, is a procedure that must be followed in the initial thinking of something actually happen. Therefore, the implementation includes all the processes included in getting a new software or hardware work correctly in their environment, including installation, configuration, running, testing, and make the necessary changes.

6.2 FORMS USED

6.2.1 Student Side

Student-home.html

This the student home page after logged in. Here there are three tabs connections, active drives and results. Each navigates to corresponding html file.

Connections.html

In this form, students can view the company details in the form of table and search the company as well.

Active Drives.html

In this form they can view the Drive post with required details for interview process which is posed by company like location, interview venue, time, selection rounds etc. And students apply for their interested drive.

Results.html

In this form the students can check the results with their name to know if they are selected or rejected in the Drive that they attended.

Profile.html

In this form students can add their own academic details and view as well and students can view the company's profile.

6.2.2 Admin Side

Admin-home.html

This is the admin home page after logged in. Here there are three tabs connections, drives, students, queries. Each navigates to corresponding html file.

Connections.html

In this form, admin can maintain the company details in the form of table and search the company as well.

Drives.html

In this form admin can view the Drive post with required details for interview process which is posed by company like location, interview venue, time, selection rounds etc. And view students applications.

Studentsdb.html

In this form, admin maintain the registered students, selected students for particular drive which is identified by drive id.

Results.html

In this form admin can view the selected students with Drive ID to know which company they got selected.

6.2.4 Company Side

Company-home.html

This is company coordinator home page after logged in. Here there are three tabs posts, applications, results, results. Each navigates to corresponding html file.

Drive-post.html

In this form, company coordinator update the drive details like position, package, location, selection rounds, interview venue, time etc.

Applications.html

In this form, company coordinator can view the students with resume those who are all applied for the particular drive.

Results.html

In this the results form, company publish interview results and also they can add new record, edit records etc. They update the selected candidate details here.

Profile.html

In this form company coordinator can add company profile with required details and view company profiles and also view the profiles of the applied candidates.

CHAPTER 7

TESTING

7.1 INTRODUCTION

Testing is a process, which reveals errors in the program. It is the major quality measure employed during software development. During software development. During testing, the program is executed with a set of test cases and the output of the program for the test cases is evaluated to determine if the program is performing as it is expected to perform.

7.2 TESTING STRATEGIES

The Test strategy document is a high-level document that outlines the testing technique used in the Software Development Life Cycle and confirms the test kinds or levels that will be performed on the product. One can't change the test strategy once it's been written, and it's been accepted by the Project Manager and development team.

In addition, the test strategy provides the following details, which are required while writing the test document:

- What technique must be used in addition to this?
- Which of the modules will be examined?
- What criteria apply for entry and exit?
- What kind of testing is necessary?

To put it another way, it's a document that explains the process of product evaluation. And the approaches can be developed using the following factors:

- Whether or not to automate
- From the standpoint of a resource

On the basis of the development design papers, we may write the test strategy.

The following documents are included in the development design document:

- Documents pertaining to the system design: These documents will mostly be used to build the test strategy.
- Design Documents: These are used to outline the software features that will be enabled in a future version.

7.3 TESTING PROCEDURE

System Testing

Involves in-house testing of the entire system before delivery to the user. Its aim is to satisfy the user the system meets all requirements of the client's specifications.

7.4 TEST CASE AND OUTPUT

7.4.1 Unit Testing

Unit Testing is done on individual modules as they are completed and become executable. It is confined only to the designer's requirements.

Each module can be tested using the following two Strategies:

Black Box Testing:

In this strategy some test cases are generated as input conditions that fully execute all functional requirements for the program. This testing has been uses to find errors in the following categories:

- Incorrect or missing functions
- Interface errors
- Errors in data structure or external database access
- Performance errors
- Initialization and termination errors.

In this testing only the output is checked for correctness. The logical flow of the data is not checked.

White Box testing:

In this the test cases are generated on the logic of each module by drawing flow graphs of that module and logical decisions are tested on all the cases. It has been uses to generate the test cases in the following cases:

- Guarantee that all independent paths have been executed.
- Execute all logical decisions on their true and false Sides.
- Execute all loops at their boundaries and within their operational bounds
- Execute internal data structures to ensure their validity

7.4.2 Functional Tests

It is a type of software testing which is used to verify the functionality of the software application, whether the function is working according to the requirement specification. In functional testing, each function tested by giving the value, determining the output, and verifying the actual output with the expected value. Functional testing performed as black-box testing which is presented to confirm that the functionality of an application or system behaves as we are expecting. It is done to verify the functionality of the application.

Functional testing also called as black-box testing, because it focuses on application specification rather than actual code. Tester has to test only the program rather than the system.

7.4.3 Performance Test

Performance testing is a non-functional Software Testing technique that determines how the stability, speed, scalability, and responsiveness of an application holds up under a given workload. It's a key step in ensuring software quality, but unfortunately, is often seen as an afterthought, in isolation, and to begin once functional testing is completed, and in most cases, after the code is ready to release.

The goals of Performance test include evaluating application output, processing speed, data transfer velocity, network bandwidth usage, maximum concurrent users, memory utilization, workload efficiency, and command response times.

7.4.4 Stress Test

Stress Testing is a software testing technique that determines the robustness of software by testing beyond the limits of normal operation. Stress testing is particularly important for critical software but is used for all types of software. Stress testing emphasizes robustness, availability, and error handling under a heavy load rather than what is correct behavior under normal situations. Stress testing is defined as a type of software testing that verifies the stability and reliability of the system. This test particularly determines the system on its robustness and error handling under extremely heavy load conditions. It even tests beyond the normal operating point and analyses how the system works under extreme conditions.

7.4.5 Structured Test

Structural testing is a type of Software testing which uses the internal design of the software for testing or in other words the software testing which is performed by the team which knows the development phase of the software, is known as structural testing.

Structural testing is basically related to the internal design and implementation of the software i.e., it involves the development team members in the testing team. It basically tests different aspects of the software according to its types. Structural testing is just the opposite of behavioral testing.

7.5 SOFTWARE TESTING STRATEGIES

Software Testing is a type of investigation to find out if there is any default or error present in the software so that the errors can be reduced or removed to increase the quality of the software and to check whether it fulfills the specifies requirements or not.

According to Glen Myers, software testing has the following objectives:

- The process of investigating and checking a program to find whether there is an error or not and does it fulfill the requirements or not is called testing.
- When the number of errors found during the testing is high, it indicates that the testing was good and is a sign of good test case.
- Finding an unknown error that wasn't discovered yet is a sign of a successful and a good test case.

The main objective of software testing is to design the tests in such a way that it systematically finds different types of errors without taking much time and effort so that less time is required for the development of the software.

7.6 INTEGRATION TESTING

Integration testing is systematic technique for constructing the program structure, while conducting test to recover errors associate with interfacing. In this case all the modules are combined and given the test data. The combined modules work successfully without any side effect on other programs and everything was found working fine.

CHAPTER 8

CONCLUSION AND FUTURE ENHANCEMENT

8.1 CONCLUSION

In conclusion, the Placement Drive Web Application is a valuable tool that can help streamline the campus placement process. By creating a centralized platform for job postings, resuse submissions, and communication between placement officer, students and company coordinator, the application cans save time and resources for both parties.

The development of such a project requires careful planning, design, and implementation, with a focus on user experience and security. It is important to consider the needs fo all stakeholders, including students, employers, and placement coordinators, and to incorporate feedback form them throughout the development process.

Overall, the placement drive web application can be a beneficial addition to any college's career services program, providing a modern and efficient way to connect students with potential employers and help them launch their careers.

8.2 FUTURE ENHANCEMENT

In future I will include some of the features are

- Integration with third party portals such as Indeed, Glassdoor, or Monster. This will allow the system to automatically post jobs openings to these portals and receive applications directly.
- Video interviews: The placement drive web application could be enhanced to include video interviews. This would allow recruiters to conduct initial interviews with candidates remotely, saving time and travel expenses. Additionally, candidates who live far from the recruiting location could still participate in the hiring process.

APPENDIXCES

APPENDIX I

SOURCE CODE

Welcome.js

```
<script>
const firebaseConfig = {
apiKey: "AIzaSyCKbzVMm-P0chJW22V9BC0CoCg5q1wE-dQ",
authDomain: "placementdrive-2c391.firebaseapp.com",
databaseURL: "https://placementdrive-2c391-default-rtdb.firebaseio.com",
projectId: "placementdrive-2c391",
storageBucket: "placementdrive-2c391.appspot.com",
messagingSenderId: "250498264872",
appId: "1:250498264872:web:ebd989e79954bd697f6a1d"};
//initialise firebase
firebase.initializeApp(ffirebaseconfig);
//reference your database
var  QueryFormDB = firebase.database().ref('QueryForm');
document.getElementById('QueryForm').addEventListener('submit',
QuerySubmitForm);
function QuerySubmitForm(e){
e.preventDefault();
var Fname = getElementByVal('Fname');
var PhNo = getElementByVal('PhNo');
var email = getElementByVal('email');
var dept = getElementByVal('dept');
var Ques = getElementByVal('Ques');
saveMessages(Fname, PhNo, email, dept, Ques);
document.querySelector(".alert").style.display = "block";
setTimeout(() => {
document.querySelector(".alert").style.display = "none";}, 3000);
document.getElementById("QueryForm").reset();}
const saveMessages = (Fname, PhNo, email, dept, Ques) => {
var newQueryForm = QueryFormDB.push();
```

```
newQueryForm.set({
Fname : Fname,
PhNo : PhNo,
email : email,
dept : dept,
Ques : Ques,});};
const getElementByVal = (id) => {
return document.getElementById(id).value;};
</script>
<script type="text/javascript">
function toggle(){
var header=document.getElementById("header")
header.classList.toggle('active')}
</script><script>
$(document).ready(function(){
$(window).scroll(function(){
if(this.scrollY > 20){
$('.navbar').addClass("sticky");
}else{
$('.navbar').removeClass("sticky");}
if(this.scrollY > 500){
$('.scroll-up-btn').addClass("show");
}else{
$('.scroll-up-btn').removeClass("show");}});
$('.scroll-up-btn').click(function(){
$('html').animate({scrollTop: 0});
$('html').css("scrollBehavior", "auto");});
$('.navbar .menu li a').click(function(){
$('html').css("scrollBehavior", "smooth");});
$('.menu-btn').click(function(){
$('.navbar .menu').toggleClass("active");
$('.menu-btn i').toggleClass("active");});
var typed = new Typed(".typing", {
```

```
strings: ["Get Placed...","Dream Job", "Dream Company"],
typeSpeed: 100,
backSpeed: 60,
loop: true});
var typed = new Typed(".typing-2", {
strings: ["WE HAVE THE TRUST OF THE BEST"],
typeSpeed: 100,
backSpeed: 60,
loop: true});
$('.carousel').owlCarousel({
margin: 20,
loop: true,
autoplay: true,
autoplayTimeOut: 2000,
autoplayHoverPause: true,
responsive: {0:{
items: 1,
nav: false},
600:{items: 2, nav: false},1000:{items: 3,nav: false
}}});});</script>
```

Admin-home.js

```
<script>
function load_connections() {
document.getElementById("htmlload").innerHTML='<object style="height:500px;
margin-top:60px; margin-left:70px; width:1000px" type="text/html"
data="adminconnections.html" ></object>';}
function load_drives() {
window.open('admindrive.html');}
function load_database() {
document.getElementById("htmlload").innerHTML='<object style="height:500px;
margin-top:60px; margin-left:70px; width:1000px" type="text/html"
data="StudentDetails.html" ></object>';}
function Query() {
```

```
document.getElementById("htmlload").innerHTML='<object style="height:500px;
margin-top:60px; margin-left:70px; width:1000px" type="text/html"
data="Query.html" ></object>';}
function load_notification() {
document.getElementById("htmlload").innerHTML='<object style="height:500px;
margin-top:60px; margin-left:70px; width:1000px" type="text/html"
data="adminnotification.html" ></object>';}
let userlink=document.getElementById('userlink');
let header=document.getElementById('hh');
let signoutlink=document.getElementById('signoutlink');
var currentUser=null;
function getUsername(){
let KeepLoggedIn=localStorage.getItem("KeepLoggedIn");
if(KeepLoggedIn =="yes"){
currentUser=JSON.parse(localStorage.getItem('user'));}
else{
currentUser=JSON.parse(sessionStorage.getItem('user'));}}
function Signout(){
sessionStorage.removeItem('user');
localStorage.removeItem('user');
localStorage.removeItem('KeepLoggedIn');
window.location="admin-login.html";}
window.onload =function(){
getUsername();
if(currentUser==null){
userlink.innerText="";
userlink.classList.replace("nav-link","btn");
userlink.classList.add("");
userlink.href="admin-login.html";
signoutlink.innerText="login";
signoutlink.classList.replace("nav-link","btn");
signoutlink.classList.add("btn-success");
signoutlink.href="admin-login.html";}
```

```
else{
userlink.innerText=currentUser.UserName;
// header.innerText="welcome"+currentUser.UserName;
userlink.classList.replace("btn","nav-link");
userlink.classList.remove("btn-primary");
userlink.href="#";
signoutlink.innerText="sign Out";
signoutlink.classList.replace("btn","nav-link");
signoutlink.classList.remove("btn-success");
signoutlink.href="javascript:Signout()";}}</script>
```

Student-home.js

```
<script>
function load_profile() {
document.getElementById("htmlload").innerHTML='<object style="height:500px;
margin-top:60px; margin-left:70px; width:1000px" type="text/html"
data="adminconnections.html" ></object>';}
function load_activedrives() {
document.getElementById("htmlload").innerHTML='<object style="height:640px;
margin-top:10px; margin-left:20px; width:1250px" type="text/html" data="st-
DrivePostRetrive.html" ></object>';}
function load_chatbox() {
document.getElementById("htmlload").innerHTML='<object style="height:580px;
margin:30px;width:1200px"type="text/html"data="StudentResults.html"
></object>';}
let userlink=document.getElementById('userlink');
let signoutlink=document.getElementById('signoutlink');
var currentUser=null;
function getUsername(){
let KeepLoggedIn=localStorage.getItem("KeepLoggedIn");
if(KeepLoggedIn =="yes"){
currentUser=JSON.parse(localStorage.getItem('user'));}
else{
currentUser=JSON.parse(sessionStorage.getItem('user'));}}
```

```
function Signout(){
sessionStorage.removeItem('user');
localStorage.removeItem('user');
localStorage.removeItem('KeepLoggedIn');
window.location="st-register.html";}
window.onload =function(){
getUsername();
if(currentUser==null){
userlink.innerText="Create New Account";
userlink.classList.replace("nav-link","btn");
userlink.classList.add("btn-primary");
userlink.href="st-register.html";
signoutlink.innerText="login";
signoutlink.classList.replace("nav-link","btn");
signoutlink.classList.add("btn-success");
signoutlink.href="st-login.html";}
else{
userlink.innerText=currentUser.FullName;
userlink.classList.replace("btn","nav-link");
userlink.classList.remove("btn-primary");
userlink.href="#";
signoutlink.innerText="sign Out";
signoutlink.classList.replace("btn","nav-link");
signoutlink.classList.remove("btn-success");
signoutlink.href="javascript:Signout()";}}
</script>
```

Company-home.js

```
<script>
function load_profile() {
document.getElementById("htmlload").innerHTML='<object style="height:500px;
margin-top:60px; margin-left:70px; width:1000px" type="text/html"
data="adminprofile.html" ></object>';}
function load_drives() {
```

```
window.open('cmdrive.html');}
function load_notification() {
document.getElementById("htmlload").innerHTML='<object style="height:500px;
margin-top:60px; margin-left:70px; width:1000px" type="text/html"
data="adminnotification.html" ></object>';}
let userlink=document.getElementById('userlink');
let header=document.getElementById('hh');
let signoutlink=document.getElementById('signoutlink');
var currentUser=null;
function getUsername(){
let KeepLoggedIn=localStorage.getItem("KeepLoggedIn");
if(KeepLoggedIn =="yes"){
currentUser=JSON.parse(localStorage.getItem('user'));}
else{
currentUser=JSON.parse(sessionStorage.getItem('user'));}}
function Signout(){
sessionStorage.removeItem('user');
localStorage.removeItem('user');
localStorage.removeItem('KeepLoggedIn');
window.location="cm-register.html";}
window.onload =function(){
getUsername();
if(currentUser==null){
userlink.innerText="Create New Account";
userlink.classList.replace("nav-link","btn");
userlink.classList.add("btn-primary");
userlink.href="cm-register.html";
signoutlink.innerText="login";
signoutlink.classList.replace("nav-link","btn");
signoutlink.classList.add("btn-success");
signoutlink.href="cm-login.html";}
else{
userlink.innerText=currentUser.CompanyName;
```

```
header.innerText="welcome"+currentUser.CompanyName;
userlink.classList.replace("btn","nav-link");
userlink.classList.remove("btn-primary");
userlink.href="#";
signoutlink.innerText="sign Out";
signoutlink.classList.replace("btn","nav-link");
signoutlink.classList.remove("btn-success");
signoutlink.href="javascript:Signout()";}}</script>
```

Drive Post.js

```
<script type="module">
import { initializeApp } from "https://www.gstatic.com/firebasejs/9.15.0/firebase-app.js";
const firebaseConfig = {
apiKey: "AIzaSyCf8TkfipanvrO0Z_FHT9cHv5LvR9ImFfM",
authDomain: "placementdrive-c2866.firebaseapp.com",
databaseURL: "https://placementdrive-c2866-default-rtdb.firebaseio.com",
projectId: "placementdrive-c2866",
storageBucket: "placementdrive-c2866.appspot.com",
messagingSenderId: "798924021032",
appId: "1:798924021032:web:0fcf61762dadde9c42841d"};
const app = initializeApp(firebaseConfig);
import{getDatabase,ref,child,onValue,get}
from "https://www.gstatic.com/firebasejs/9.15.0/firebase-database.js"
const db=getDatabase();
var Outerdiv = document.getElementById("box");
var ArrayOfDrivePost=[];
window.addEventListener('load',GetAllDrivePost);
function GetAllDrivePost(){
const dbRef = ref(db);
get(child(dbRef,"DrivePost"))
.then((snapshot)=>{
snapshot.forEach(Post => {
ArrayOfDrivePost.push(Post.val());});
```

```
AddAllPost();
})}
function AddAllPost(){
let i=0;
ArrayOfDrivePost.forEach(Post=>{
AddAPost(Post,i++);});
AssignEvents();}
function AddAPost(Post,index){
let html=`
<div id="RequestCard" id="scroll">
<label class="cn" id="CompanyName`+index+`">`+Post.CompanyName+`</label>
<label  class="jr" id="Jobrole `+index+`">`+Post.Jobrole+`</label>
<label  class="jt" id="Jobtype `+index+`">`+Post.Jobtype+`</label>
<label  class="loc" id="Location `+index+`">`+Post.Location+`</label>
<label  class="pack" id="Package `+index+`">`+Post.Package+` LPA</label>
<label  class="Eligible" id="E"`+index+`">Eligibility : `+Post.Eligibility+`</label>
<button class="viewdetails" id="detbtn-`+index+`">View Details</button></div>`
let newPost=document.createElement('div');
newPost.classList.add('RequestCard');
newPost.innerHTML=html;
Outerdiv.append(newPost);}
function GetPostIndex(id){
var indstart=id.indexOf('-')+1;
var indend=id.length;
return Number(id.substring(indstart,indend));}
function GoToPostDetails(event){
var index = GetPostIndex(event.target.id);
localStorage.Post = JSON.stringify(ArrayOfDrivePost[index]);
console.log(ArrayOfDrivePost[index]);
var newWindow = window.open();
newWindow.document.location.href = "PostDetails.html";}
function AssignEvents(){
var vdbtn = document.getElementsByClassName('viewdetails');
```

```
for(let i=0;i<vdbtn.length;i++){
vdbtn[i].addEventListener('click',GoToPostDetails);
}}</script>
```

Drive Details.js

```
<script type="module">
const createbtn=document.getElementById('applynow');
let Post=null;
window.onload=function(){
Post=localStorage.Post;
if(Post){
Post=JSON.parse(Post);
LoadPost();}}
function LoadPost(){
document.getElementById('cn').innerHTML= Post.CompanyName;
document.getElementById('DriveId').innerHTML= "Drive ID :";
document.getElementById('DriveId').innerHTML+= Post.DriveId;
document.getElementById('jr').innerHTML= Post.Jobrole;
document.getElementById('so').innerHTML= " Starts On ";
document.getElementById('so').innerHTML+= Post.StartsOn;
document.getElementById('v1').innerHTML+= Post.Venue;
document.getElementById('v2').innerHTML+= Post.Venue;
document.getElementById('v3').innerHTML+= Post.Venue;
document.getElementById('loc').innerHTML+= Post.Location;
document.getElementById('pack').innerHTML+= Post.Package;
document.getElementById('pack').innerHTML+=" LPA";
document.getElementById('e').innerHTML= "Eligibility : ";
document.getElementById('e').innerHTML+= Post.Eligibility;
document.getElementById('mes').innerHTML+="<br> ";
document.getElementById('mes').innerHTML+=Post.Message;
document.getElementById('r1t').innerHTML+= Post.r1t;
document.getElementById('r1t').innerHTML+= " AM ";
document.getElementById('r1d').innerHTML+= Post.r1d;
document.getElementById('r1n').innerHTML= Post.r1n;
```

```
document.getElementById('r2t').innerHTML+= Post.r2t;
document.getElementById('r2t').innerHTML+= " AM ";
document.getElementById('r2d').innerHTML+= Post.r2d;
document.getElementById('r2n').innerHTML= Post.r2n;
document.getElementById('r3t').innerHTML+= Post.r3t;
document.getElementById('r3t').innerHTML+= " AM ";
document.getElementById('r3d').innerHTML+= Post.r3d;
document.getElementById('r3n').innerHTML= Post.r3n;
document.getElementById('applynow').innerHTML="Apply Now";}
$(document).ready(function(){
$(".content .radio_content").hide();
$(".content .radio_content:first-child").show();
$(".radio_wrap").click(function(){
var current_raido = $(this).attr("data-radio");
$(".content .radio_content").hide();
$("."+current_raido).show();
})});
ApplyNow(){
window.open('JobApplicationForm.html');}
.addEventListener('click',ApplyNow);
</script>
```

Job Applications.js

```
<script type="module">
import { initializeApp } from "https://www.gstatic.com/firebasejs/9.17.0/firebase-app.js";
const firebaseConfig = {
apiKey: "AIzaSyCf8TkfipanvrO0Z_FHT9cHv5LvR9ImFfM",
authDomain: "placementdrive-c2866.firebaseapp.com",
databaseURL: "https://placementdrive-c2866-default-rtdb.firebaseio.com",
projectId: "placementdrive-c2866",
storageBucket: "placementdrive-c2866.appspot.com",
messagingSenderId: "798924021032",
appId: "1:798924021032:web:0fcf61762dadde9c42841d"};
```

```
const app = initializeApp(firebaseConfig);
import{getStorage,ref as sRef,uploadBytesResumable,getDownloadURL}
from"https://www.gstatic.com/firebasejs/9.17.0/firebase-storage.js"
import{getDatabase,ref,child,onValue,get}
from "https://www.gstatic.com/firebasejs/9.17.0/firebase-database.js"
const db=getDatabase();
var reader=new FileReader();
var sno=0;
var tbody=document.getElementById('tbody1');
var myimg=document.getElementById('myimg');
var DownBtn=document.getElementById('downbtn');
var namebox=document.getElementById('namebox');
var input = document.createElement('input');
input.type='file';
input.onchange = e =>{
files=e.target.files;
var extention = GetFileExt(files[0]);
var name= GetFileName(files[0]);
namebox.value=name;
extlab.innerHTML=extention;
reader.readAsDataURL(files[0]); }
reader.onload= function() {
myimg.src= reader.result;}
functionAddItemToTable(Name,RegisterNumber,Department,Year,Email,ResumeNa
me,DriveId,ResumeFileUrl){
let trow = document.createElement("tr");
let td1=document.createElement('td');
let td2=document.createElement('td');
let td3=document.createElement('td');
let td4=document.createElement('td');
let td5=document.createElement('td');
let td6=document.createElement('td');
let td7=document.createElement('td');
```

```
let td8=document.createElement('td');
let td9=document.createElement('td');
td1.innerHTML= ++sno;
td2.innerHTML= Name;
td3.innerHTML= RegisterNumber;
td4.innerHTML=Department;
td5.innerHTML= Year;
td6.innerHTML= Email;
td7.innerHTML= ResumeName;
td8.innerHTML= DriveId;
trow.appendChild(td1);
trow.appendChild(td2);
trow.appendChild(td3);
trow.appendChild(td4);
trow.appendChild(td5);
trow.appendChild(td6);
trow.appendChild(td7);
trow.appendChild(td8);
tbody.appendChild(trow);}
function AddAllItemsToTable(TheQuery){
sno=0;
tbody.innerHTML="";
TheQuery.forEach(element => {
AddItemToTable(element.FullName,element.DriveId, element.RegisterNumber,
element.Department, element.Year, element.Email,element.ResumeName);});}
function GetAllDataOnce(){
const dbRef=ref(db);
get(child(dbRef,"JobApplication"))
.then((snapshot)=>{
var Student=[];
snapshot.forEach(childSnapshot=> {
Student.push(childSnapshot.val());
});
```

```
AddAllItemsToTable(Student);});}
window.onload = GetAllDataOnce;
function GetUrlfromRealtimDB() {
var name=namebox.value;
var dbref=ref(db);
get(child(dbref,"ResumeLinks/"+name)).then((snapshot)=>{
if (snapshot.exists()) {
myimg.src=snapshot.val().ImgUrl;
}))}
DownBtn.onclick=GetUrlfromRealtimDB;
</script>
<script>
const searchfunction=()=>{
let filter=document.getElementById("SearchBar").value.toUpperCase();
let mytable =document.getElementById('tbody1');
let tr =mytable.getElementsByTagName('tr');
for(var i=0;i<tr.length;i++){
let td=tr[i].getElementsByTagName('td');
for(j=0;j<td.length;j++){
if(td[j]){
let textvalue=td[j].textContent || td[j].innerHTML;
if(textvalue.toUpperCase().indexOf(filter)>-1){
tr[i].style.display="";
break;}
else{
tr[i].style.display="none";
}}}}}
</script>
```

Application Details.js

```
<script type="module">
import { initializeApp } from "https://www.gstatic.com/firebasejs/9.17.0/firebase-app.js";
const firebaseConfig = {
```

```
apiKey: "AIzaSyCf8TkfipanvrO0Z_FHT9cHv5LvR9ImFfM",
authDomain: "placementdrive-c2866.firebaseapp.com",
databaseURL: "https://placementdrive-c2866-default-rtdb.firebaseio.com",
projectId: "placementdrive-c2866",
storageBucket: "placementdrive-c2866.appspot.com",
messagingSenderId: "798924021032",
appId: "1:798924021032:web:0fcf61762dadde9c42841d"};
const app = initializeApp(firebaseConfig);
import{getStorage,ref as sRef,uploadBytesResumable,getDownloadURL}
from"https://www.gstatic.com/firebasejs/9.17.0/firebase-storage.js"
import{getDatabase,ref,child,onValue,get}
from "https://www.gstatic.com/firebasejs/9.17.0/firebase-database.js"
const db=getDatabase();
const createbtn=document.getElementById('ViewProfile');
let Post=null;
window.onload=function(){
Post=localStorage.Post;
if(Post){
Post=JSON.parse(Post);
LoadPost();
GetUrlfromRealtimDB();}}
function LoadPost(){
document.getElementById('fn').innerHTML= Post.FullName;
document.getElementById('DriveId').innerHTML= "DriveID <br>";
document.getElementById('DriveId').innerHTML+= Post.DriveId;
document.getElementById('dept').innerHTML= "Department <br>";
document.getElementById('dept').innerHTML+= Post.Department;
document.getElementById('Campus').innerHTML= " Campus <br>";
document.getElementById('Campus').innerHTML+= "RGCET";
document.getElementById('loc').innerHTML= "RegisterNumber <br>";
document.getElementById('loc').innerHTML+= Post.RegisterNumber;
var reader=new FileReader();
var myimg=document.getElementById('myimg');
```

```
var DownBtn=document.getElementById('downbtn');
var input = document.createElement('input');
input.type='file';
input.onchange = e =>{
files=e.target.files;
var extention = GetFileExt(files[0]);
var name= GetFileName(files[0]);
extlab.innerHTML=extention;
reader.readAsDataURL(files[0]);}
reader.onload= function() {
myimg.src= reader.result;}
function GetUrlfromRealtimDB() {
var dbref=ref(db);
get(child(dbref,"ResumeLinks/"+Post.ResumeName)).then((snapshot)=>{
if (snapshot.exists()) {
myimg.src=snapshot.val().ImgUrl;
}})}
$(document).ready(function(){
$(".content .radio_content").hide();
$(".content .radio_content:first-child").show();
$(".radio_wrap").click(function(){
var current_raido = $(this).attr("data-radio");
$(".content .radio_content").hide();
$("."+current_raido).show();})});
function ViewProfile(){
window.open('JobApplicationForm.html');
}</script>
```

Update results.js

```
<script type="module">
import { initializeApp } from "https://www.gstatic.com/firebasejs/9.17.0/firebase-app.js";
const firebaseConfig = {
apiKey: "AIzaSyCf8TkfipanvrO0Z_FHT9cHv5LvR9ImFfM",
```

```
authDomain: "placementdrive-c2866.firebaseapp.com",
databaseURL: "https://placementdrive-c2866-default-rtdb.firebaseio.com",
projectId: "placementdrive-c2866",
storageBucket: "placementdrive-c2866.appspot.com",
messagingSenderId: "798924021032",
appId: "1:798924021032:web:0fcf61762dadde9c42841d"
};
const app = initializeApp(firebaseConfig);
import { getDatabase, ref, set, child, get ,update,remove}
from "https://www.gstatic.com/firebasejs/9.17.0/firebase-database.js";
const db = getDatabase();
const Sno=document.getElementById('sno');
const Email =document.getElementById('Email');
const Name=document.getElementById('Name');
const RegisterNumber=document.getElementById('RegisterNumber');
const DriveID=document.getElementById('DriveID');
const Company=document.getAnimations('CompanyMod');
const Department=document.getElementById('Department');
const Year=document.getElementById('Year');
var sno=0;
var tbody=document.getElementById('tbody1');
function
AddItemToTable(Name,Email,RegisterNumber,DriveID,CompanyName,Department,
Year){
let trow = document.createElement("tr");
let td0=document.createElement('td');
let td1=document.createElement('td');
let td2=document.createElement('td');
let td3=document.createElement('td');
let td4=document.createElement('td');
let td5=document.createElement('td');
let td6=document.createElement('td');
let td7=document.createElement('td');
```

```
td0.innerHTML= ++sno;
td1.innerHTML= Name;
td2.innerHTML= Email;
td3.innerHTML=RegisterNumber;
td4.innerHTML= DriveID;
td5.innerHTML= CompanyName;
td6.innerHTML= Department;
td7.innerHTML= Year;
trow.appendChild(td0);
trow.appendChild(td1);
trow.appendChild(td2);
trow.appendChild(td3);
trow.appendChild(td4);
trow.appendChild(td5);
trow.appendChild(td6);
trow.appendChild(td7);
tbody.appendChild(trow);}
function AddAllItemsToTable(TheQuery){
sno=0;
tbody.innerHTML="";
TheQuery.forEach(element => {
AddItemToTable(element.Name, element.Email, element.RegisterNumber,
element.DriveID, element.CompanyName, element.Department, element.Year);});}
function GetAllDataOnce(){
const dbRef=ref(db);
get(child(dbRef,"Results"))
.then((snapshot)=>{
var Student=[];
snapshot.forEach(childSnapshot=> {
Student.push(childSnapshot.val());});
AddAllItemsToTable(Student);
});}
window.onload = GetAllDataOnce;
```

```
var NameMod=document.getElementById('NameMod');
var EmailMod=document.getElementById('EmailMod');
var RegMod=document.getElementById('RegMod');
var DriveIDMod=document.getElementById('DriveIDMod');
var CompanyMod=document.getElementById('CompanyMod');
var DeptMod=document.getElementById('DeptMod');
var YearMod=document.getElementById('YearMod');
var NameMod1=document.getElementById('NameMod1');
var EmailMod1=document.getElementById('EmailMod1');
var RegMod1=document.getElementById('RegMod1');
var DriveIDMod1=document.getElementById('DriveIDMod1');
var CompanyMod1=document.getElementById('CompanyMod1');
var DeptMod1=document.getElementById('DeptMod1');
var YearMod1=document.getElementById('YearMod1');
var insbtn=document.getElementById("insbtn");
function InsertData(){
set(ref(db, "Results/"+ NameMod.value),{
Email:EmailMod.value,
Name:NameMod.value,
RegisterNumber:RegMod.value,
DriveID:DriveIDMod.value,
CompanyName:CompanyMod.value,
Department:DeptMod.value,
Year:YearMod.value})
.then(()=>{
alert("Data Stored Successfully 👆");})
.catch((error)=>{
alert("Unsuccessful, Error □"+error);});}
insbtn.addEventListener("click",InsertData);
var selbtn=document.getElementById("Selbtn");
var updbtn=document.getElementById("Updbtn");
var delbtn=document.getElementById("Delbtn");
function SelectData(){
```

```
const dbref=ref(db);
get(child(dbref,"Results/"+ NameMod1.value)).then((snapshot)=>{
if(snapshot.exists()){
NameMod1.value=snapshot.val().Name;
EmailMod1.value=snapshot.val().Email;
RegMod1.value=snapshot.val().RegisterNumber;
DriveIDMod1.value=snapshot.val().DriveID;
CompanyMod1.value=snapshot.val().CompanyName;
DeptMod1.value=snapshot.val().Department;
YearMod1.value=snapshot.val().Year; }
else{
alert("No Data found □");}})
.catch((error)=>{
alert("Unsuccessful, Error □"+error);
});}
function UpdateData(){
update(ref(db, "Results/"+ NameMod1.value),{
Name: NameMod1.value,
Email: EmailMod1.value,
RegisterNumber: RegMod1.value,
DriveID: DriveIDMod1.value,
CompanyName: CompanyMod1.value,
Department: DeptMod1.value,
Year: YearMod1.value}).then(()=>{
alert("Data Updated Successfully 👆");})
.catch((error)=>{
alert("Unsuccessful, Error □"+error);});}
function DeleteData(){
remove(ref(db, "Results/"+ NameMod1.value))
.then(()=>{
alert("Data Deleted Successfully 👆");})
.catch((error)=>{
alert("Unsuccessful, Error □"+error);});}
```

```
selbtn.addEventListener("click",SelectData);
updbtn.addEventListener("click",UpdateData);
delbtn.addEventListener("click",DeleteData);
</script>
<script>
const searchfunction=()=>{
let filter=document.getElementById("SearchBar").value.toUpperCase();
let mytable =document.getElementById('tbody1');
let tr =mytable.getElementsByTagName('tr');
for(var i=0;i<tr.length;i++){
let td=tr[i].getElementsByTagName('td');
for(j=0;j<td.length;j++){
if(td[j]){
let textvalue=td[j].textContent || td[j].innerHTML;
if(textvalue.toUpperCase().indexOf(filter)>-1){
tr[i].style.display="";
break;}
else{
tr[i].style.display="none";}}}}}
</script>
```

View results.js

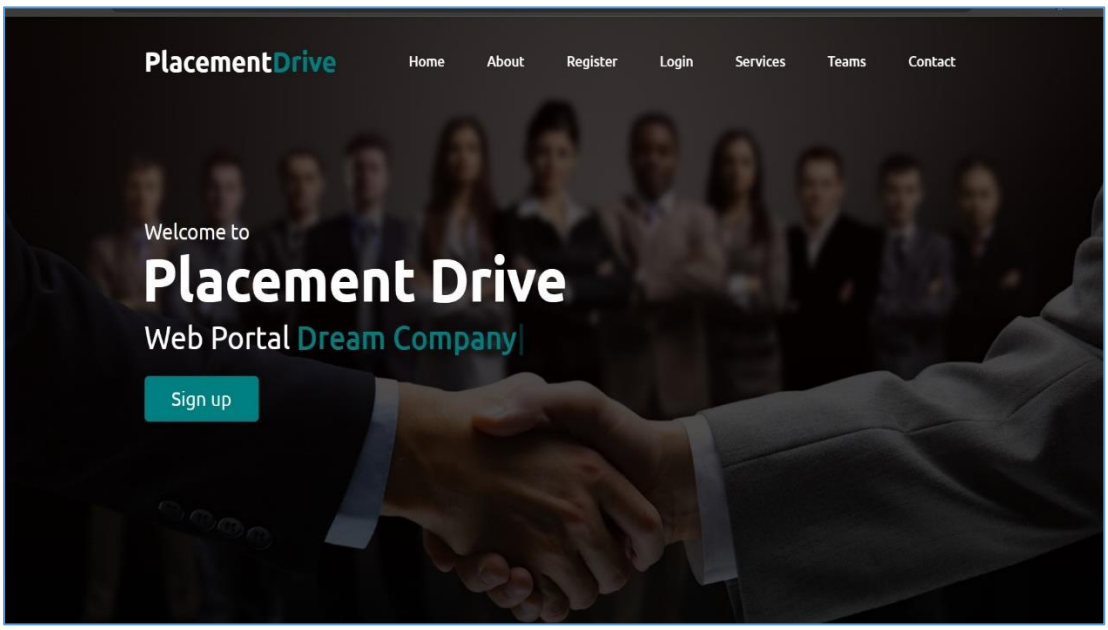
```
<script type="module">
import { initializeApp } from "https://www.gstatic.com/firebasejs/9.17.0/firebase-app.js";
const firebaseConfig = {
apiKey: "AIzaSyCf8TkfipanvrO0Z_FHT9cHv5LvR9ImFfM",
authDomain: "placementdrive-c2866.firebaseapp.com",
databaseURL: "https://placementdrive-c2866-default-rtdb.firebaseio.com",
projectId: "placementdrive-c2866",
storageBucket: "placementdrive-c2866.appspot.com",
messagingSenderId: "798924021032",
appId: "1:798924021032:web:0fcf61762dadde9c42841d"};
const app = initializeApp(firebaseConfig);
```

```
import { getDatabase, ref, set, child, get ,update,remove}
from "https://www.gstatic.com/firebasejs/9.17.0/firebase-database.js";
const db = getDatabase();
var NameMod1=document.getElementById('NameMod1');
var EmailMod1=document.getElementById('EmailMod1');
var RegMod1=document.getElementById('RegMod1');
var DriveIDMod1=document.getElementById('DriveIDMod1');
var CompanyMod1=document.getElementById('CompanyMod1');
var DeptMod1=document.getElementById('DeptMod1');
var YearMod1=document.getElementById('YearMod1');
var greet=document.getElementById('greet');
var selbtn=document.getElementById("Selbtn");
function SelectData(){
const dbref=ref(db);
get(child(dbref,"Results/"+ NameMod1.value)).then((snapshot)=>{
if(snapshot.exists()){
NameMod1.value=snapshot.val().Name;
NameMod2.value=snapshot.val().Name;
EmailMod1.value=snapshot.val().Email;
RegMod1.value=snapshot.val().RegisterNumber;
DriveIDMod1.value=snapshot.val().DriveID;
CompanyMod1.value=snapshot.val().CompanyName;
DeptMod1.value=snapshot.val().Department;
YearMod1.value=snapshot.val().Year;
greet.innerHTML=("Congrats! you are Selected in ");
greet.innerHTML+=snapshot.val().CompanyName;}
else{
alert("No Data found □");}})
.catch((error)=>
{alert("Unsuccessful, Error □"+error);});}
selbtn.addEventListener("click",SelectData);
</script>
```

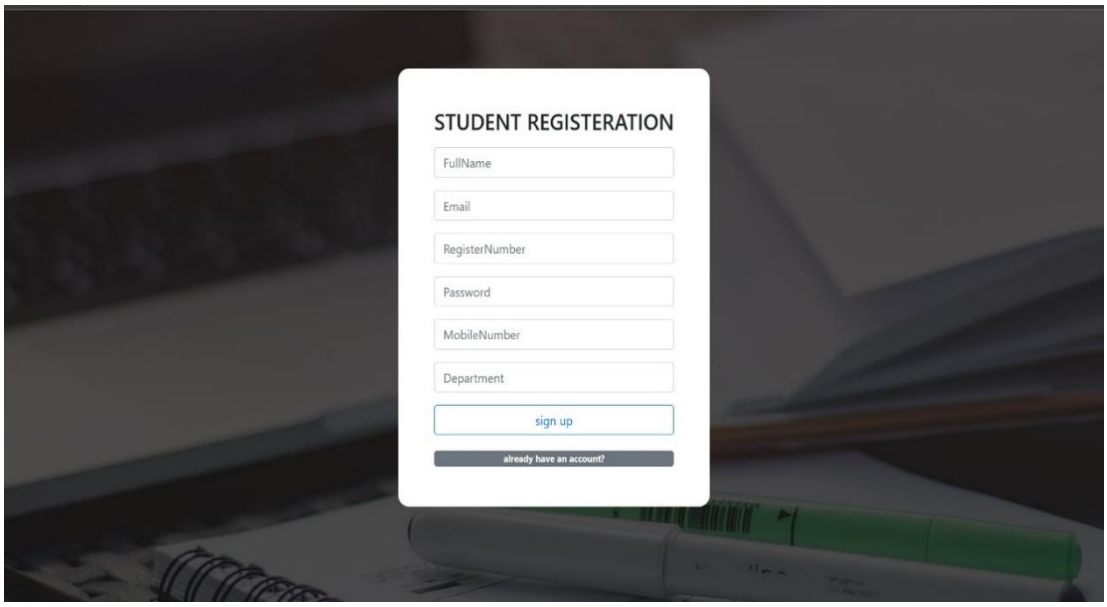
APPENDIX II

SCREENSHOTS

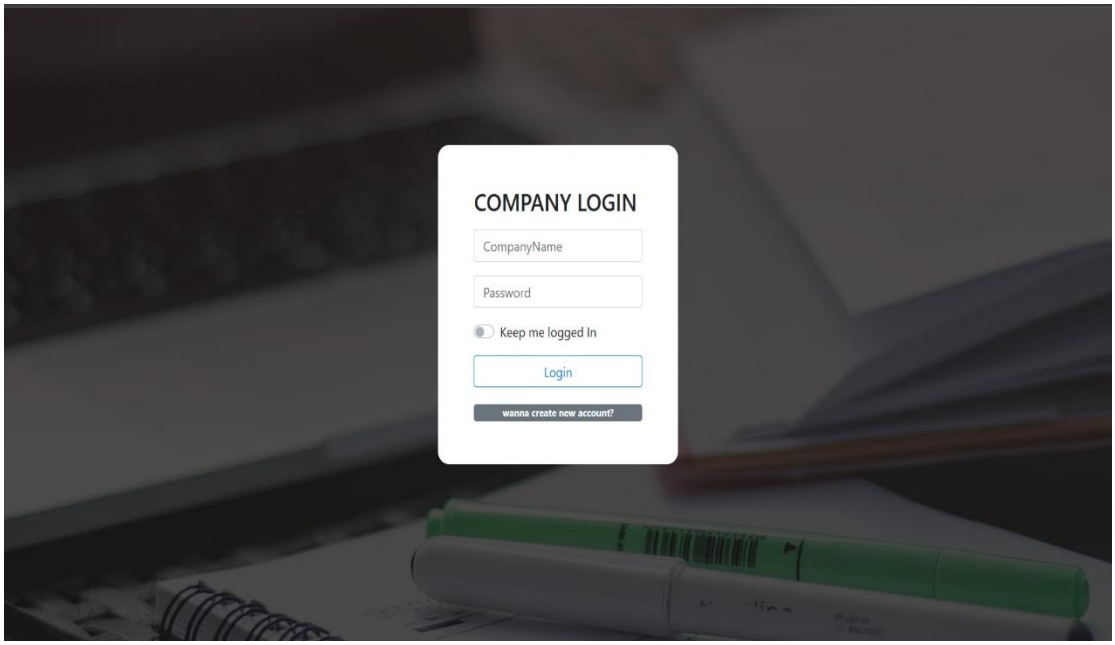
Welcome Page



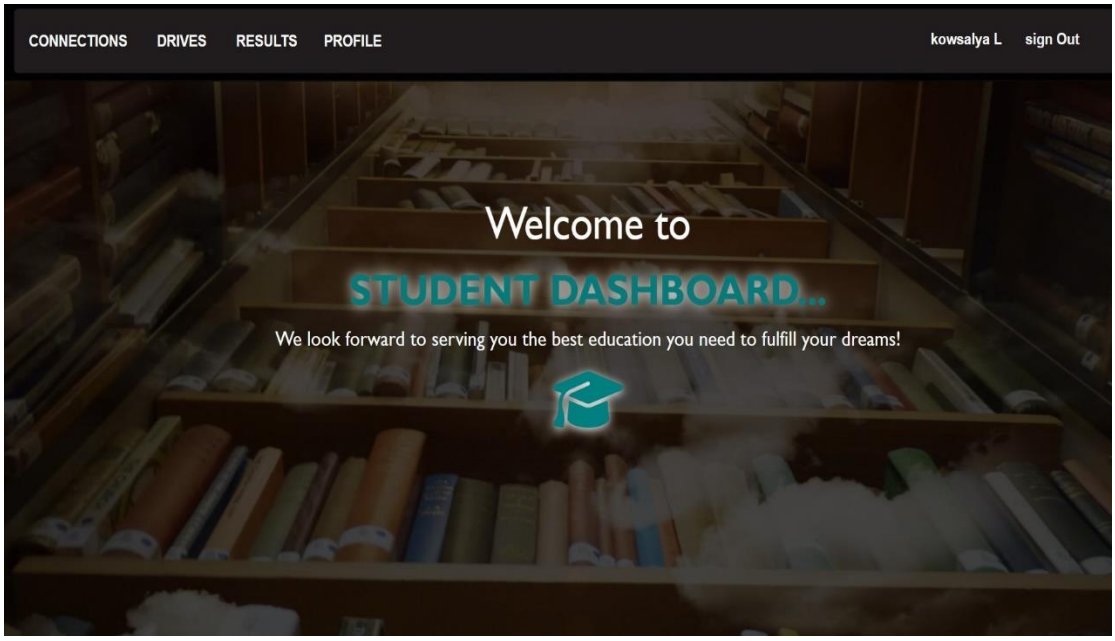
Registration Form



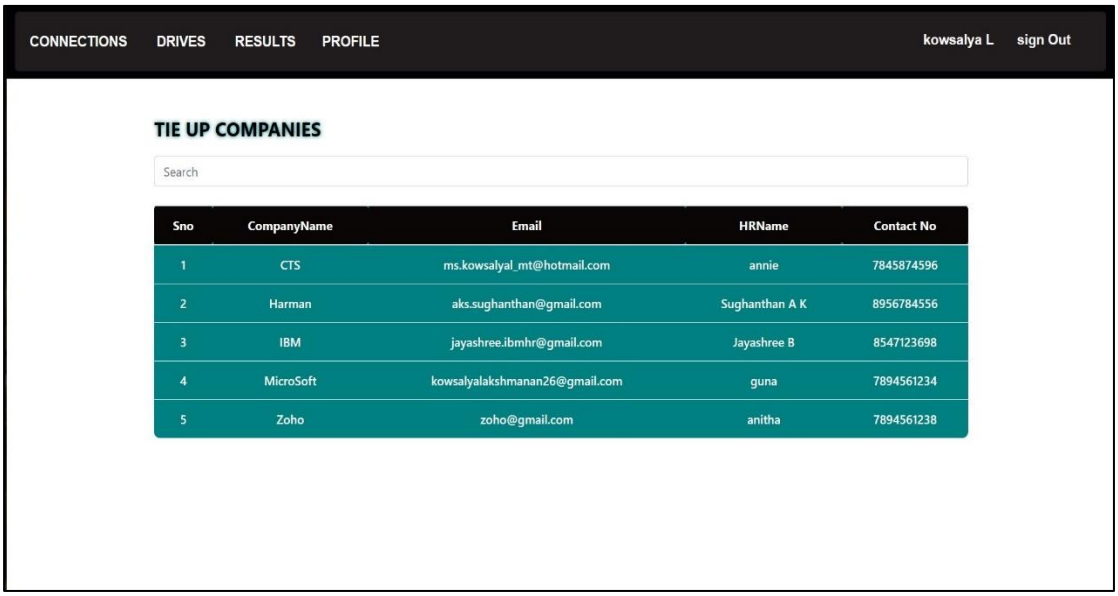
Login Form



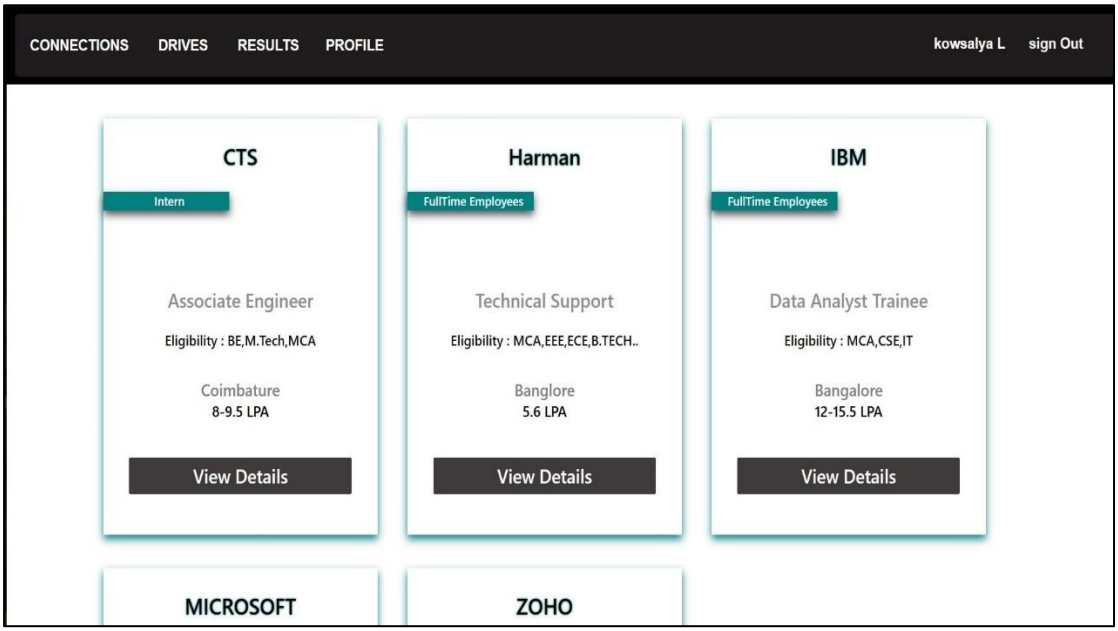
Student Module



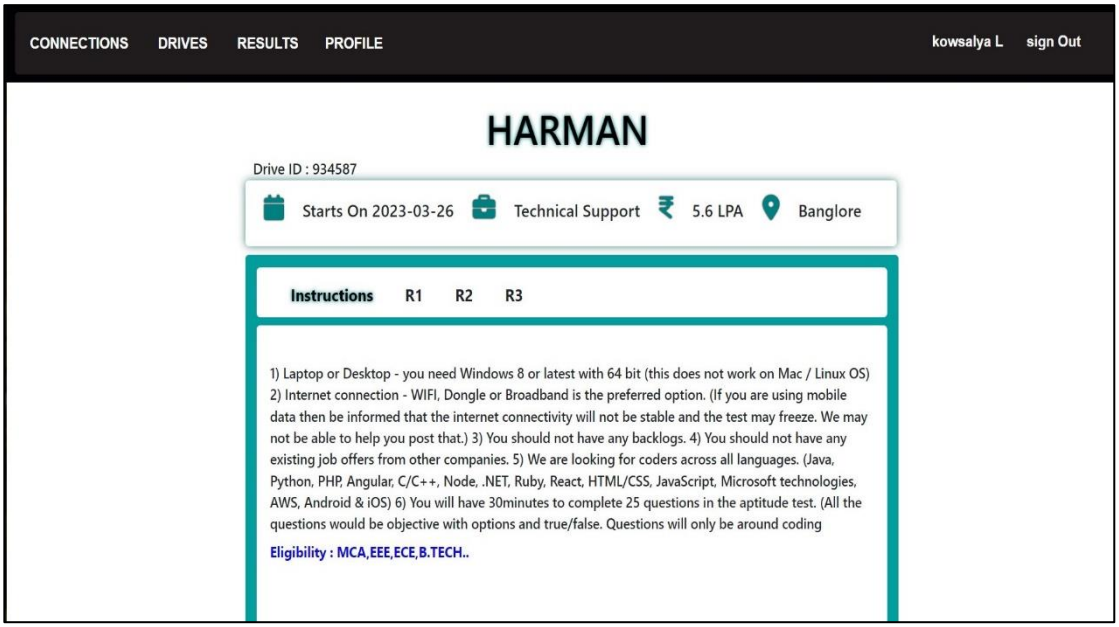
Students can view the companies tied up with college



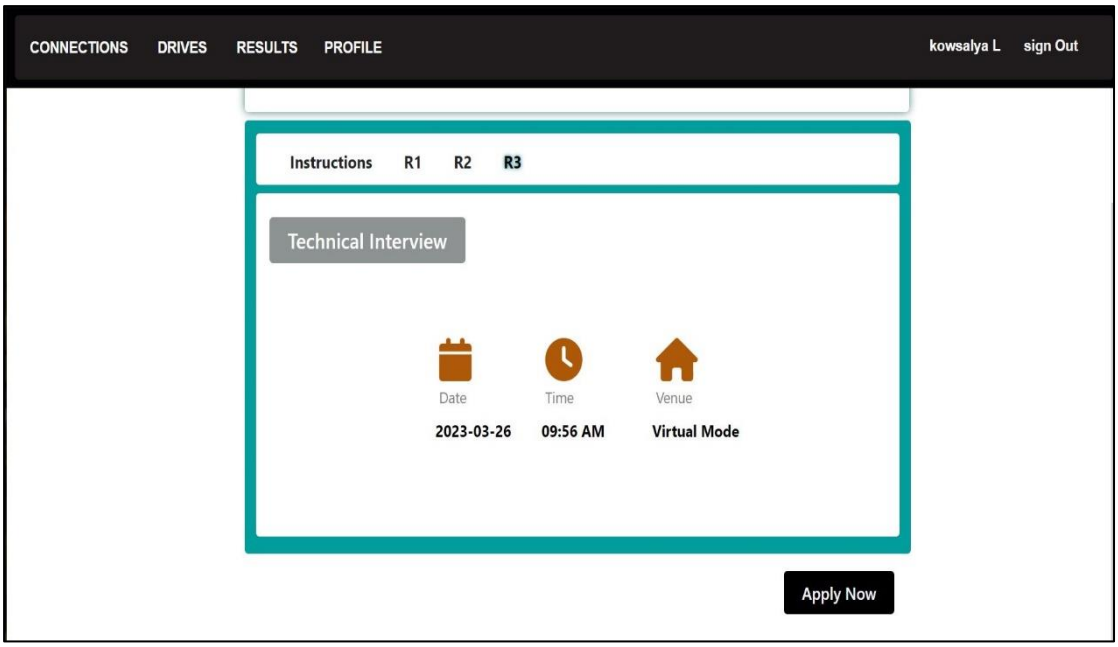
Student can view Active Drive for placement



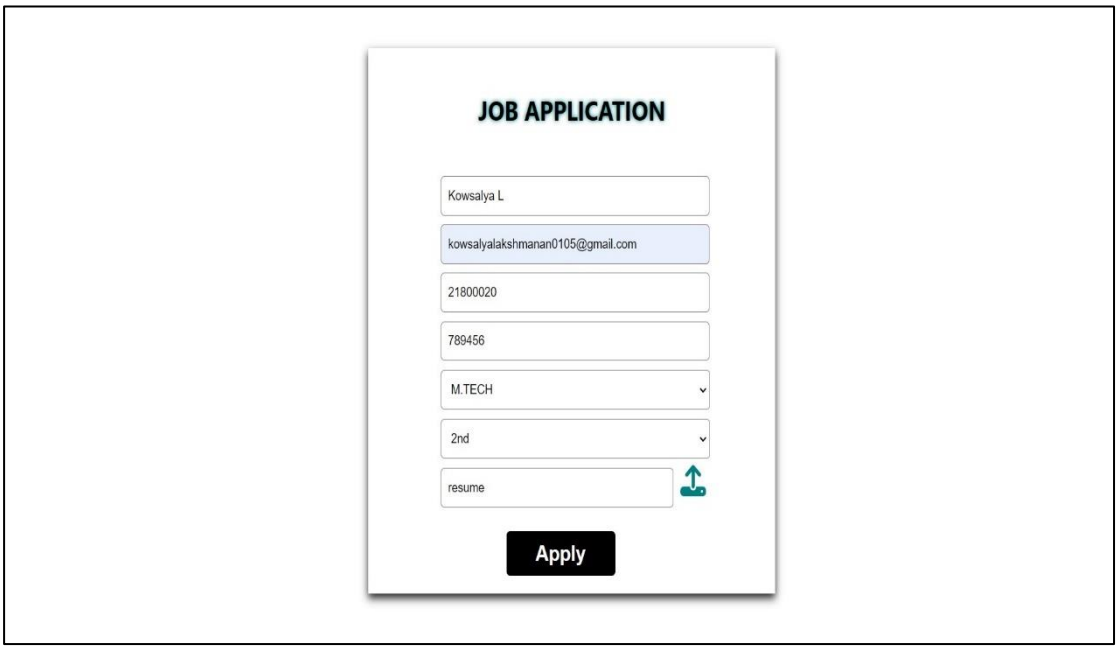
Here students can see instruction and other drive details



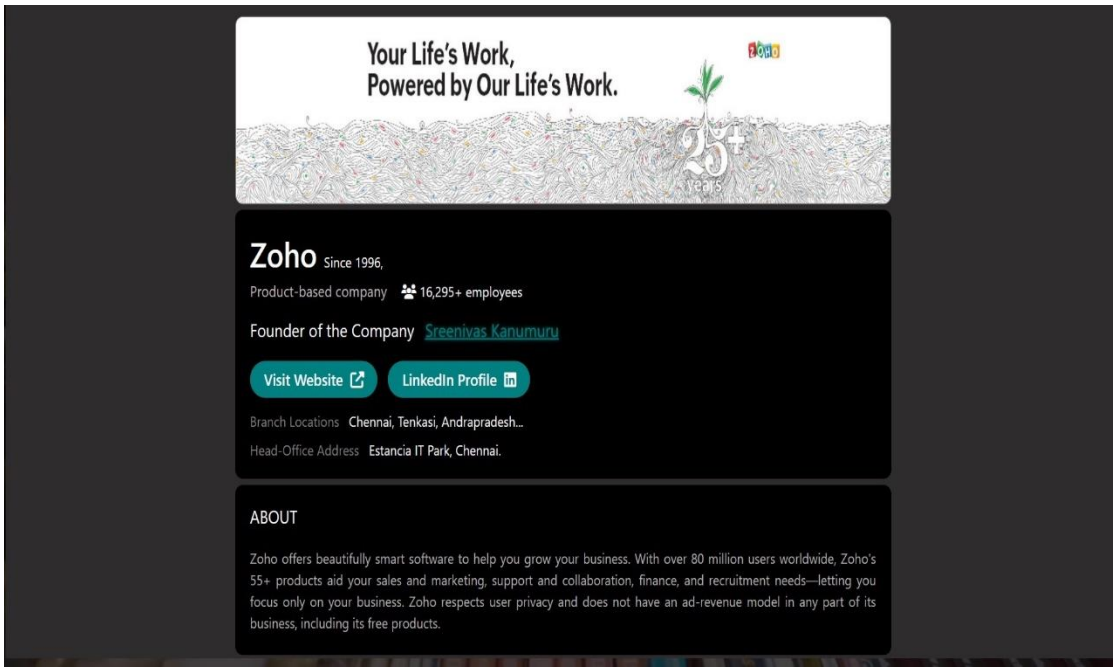
Here they can note the interview rounds, venue, timing and apply



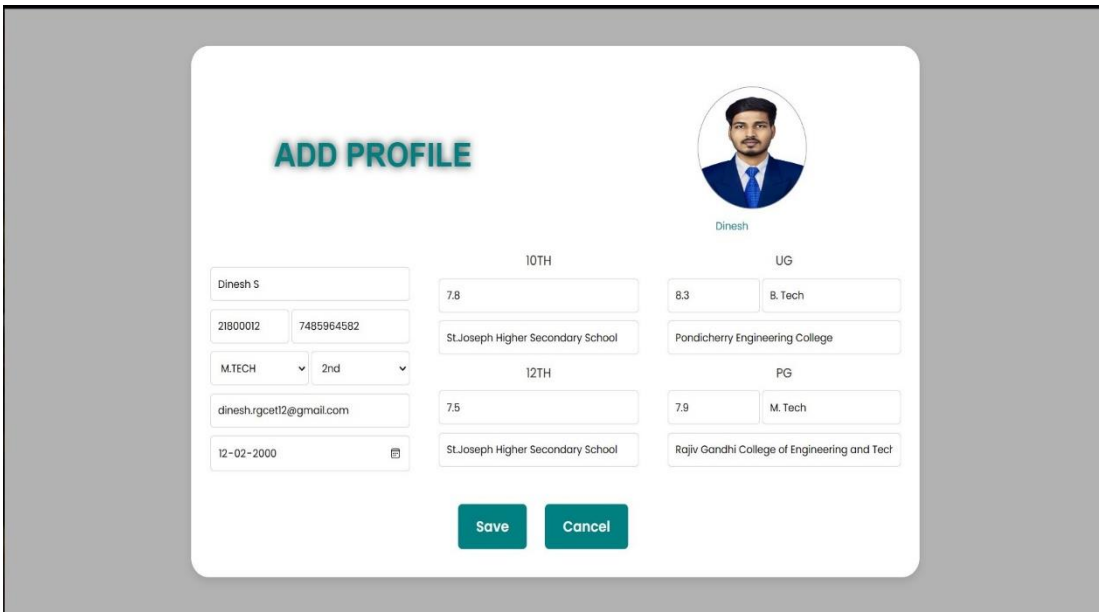
Job application form for students



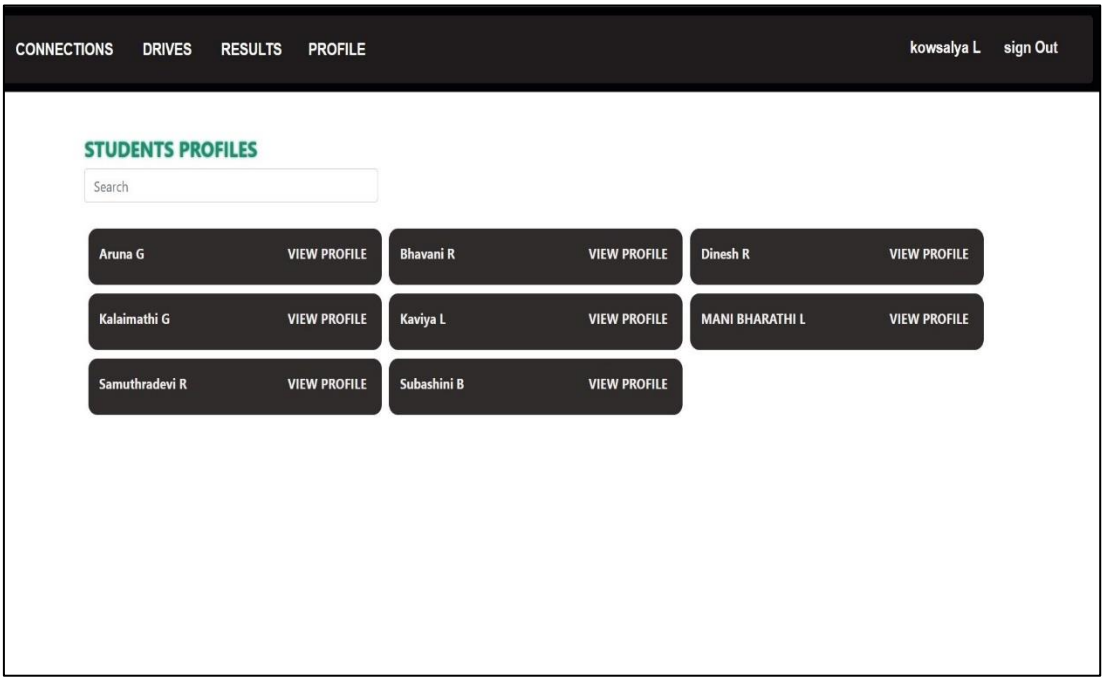
Students can view the company profile to know more.



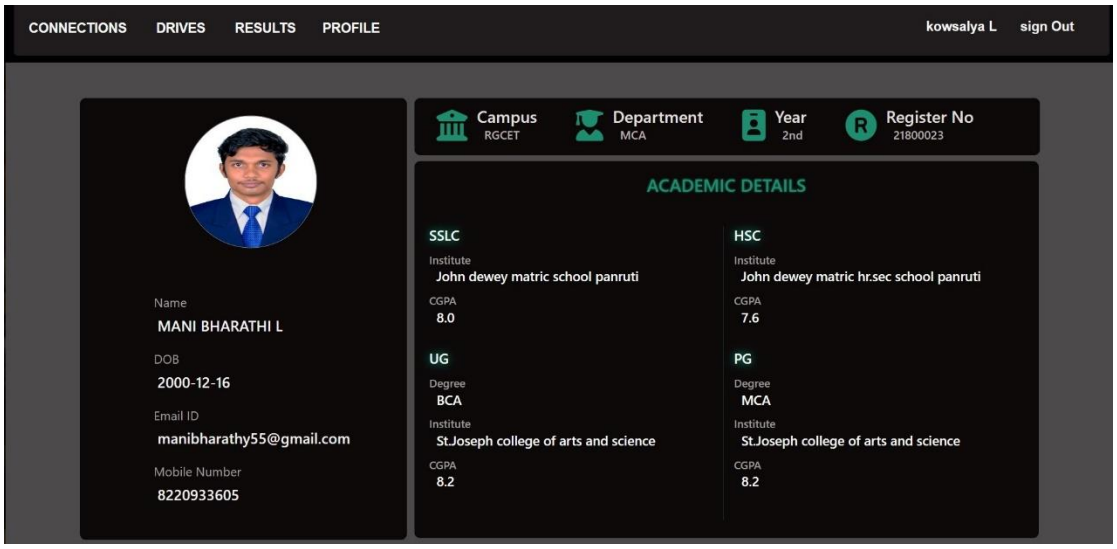
Student can add their own academic details



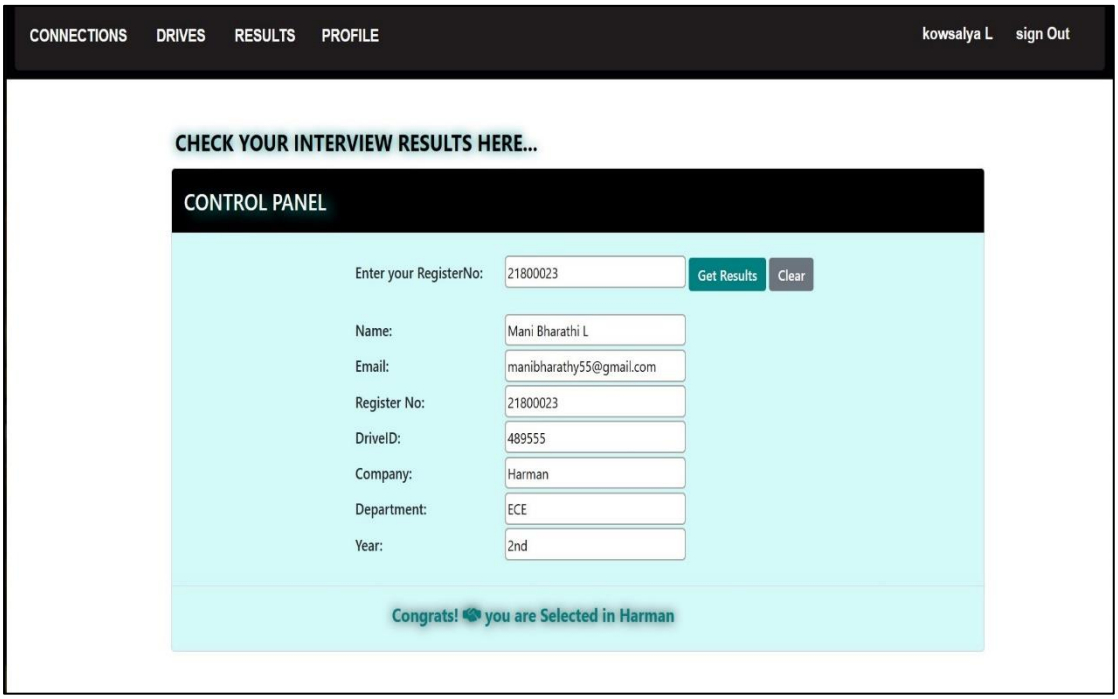
And view the all student details



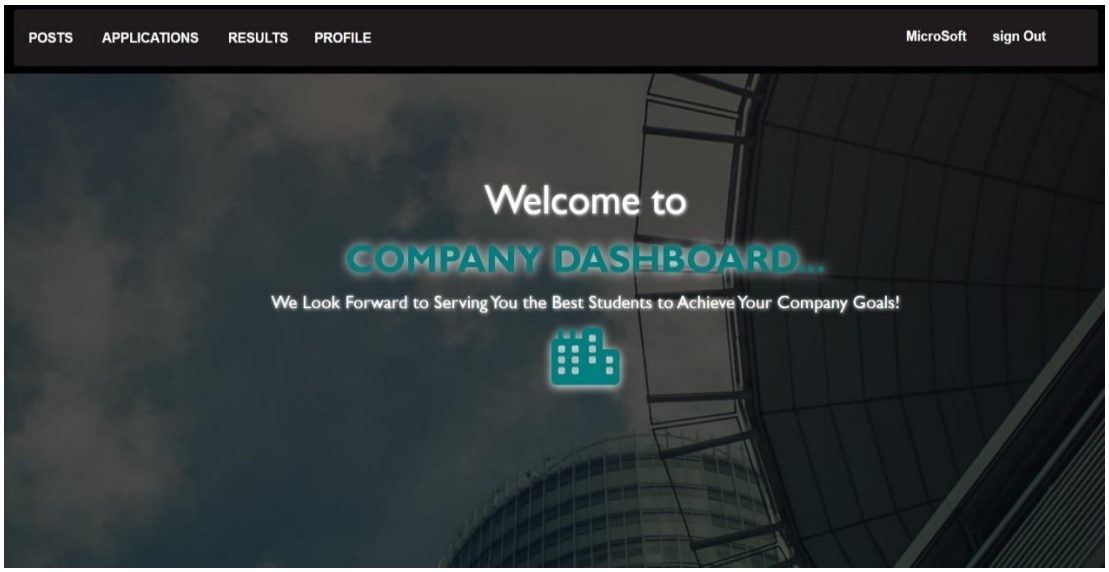
View of Student profile after updated



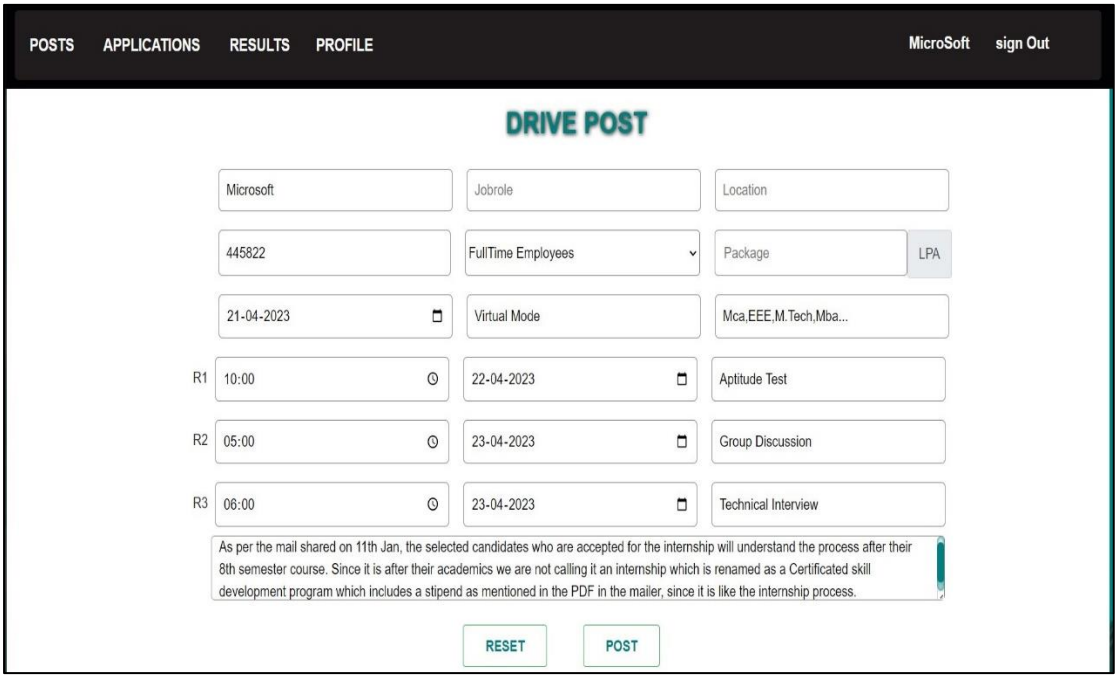
Students can check their results by their register number.



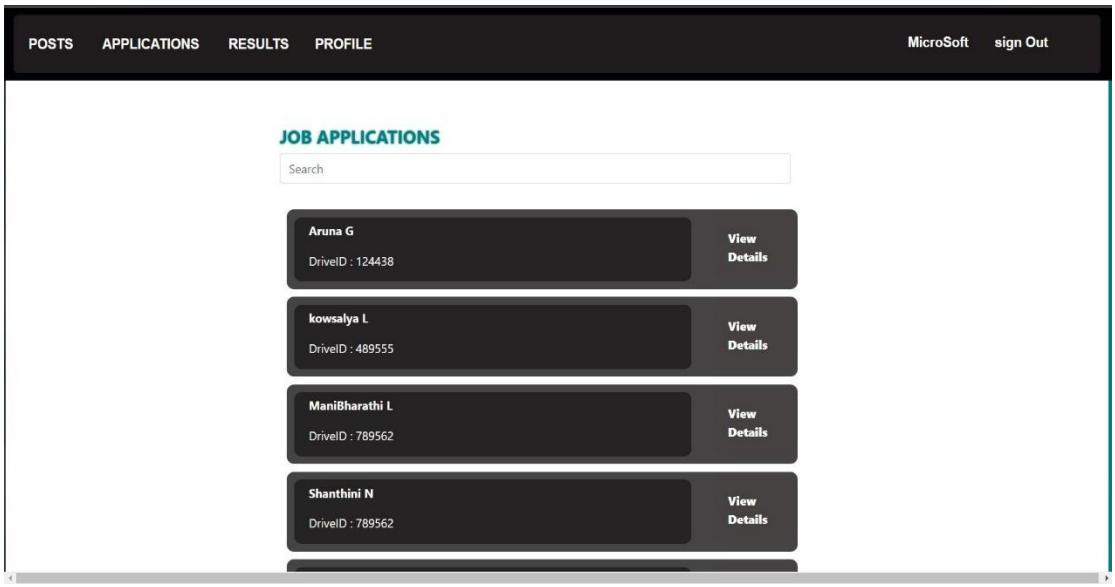
Company Module



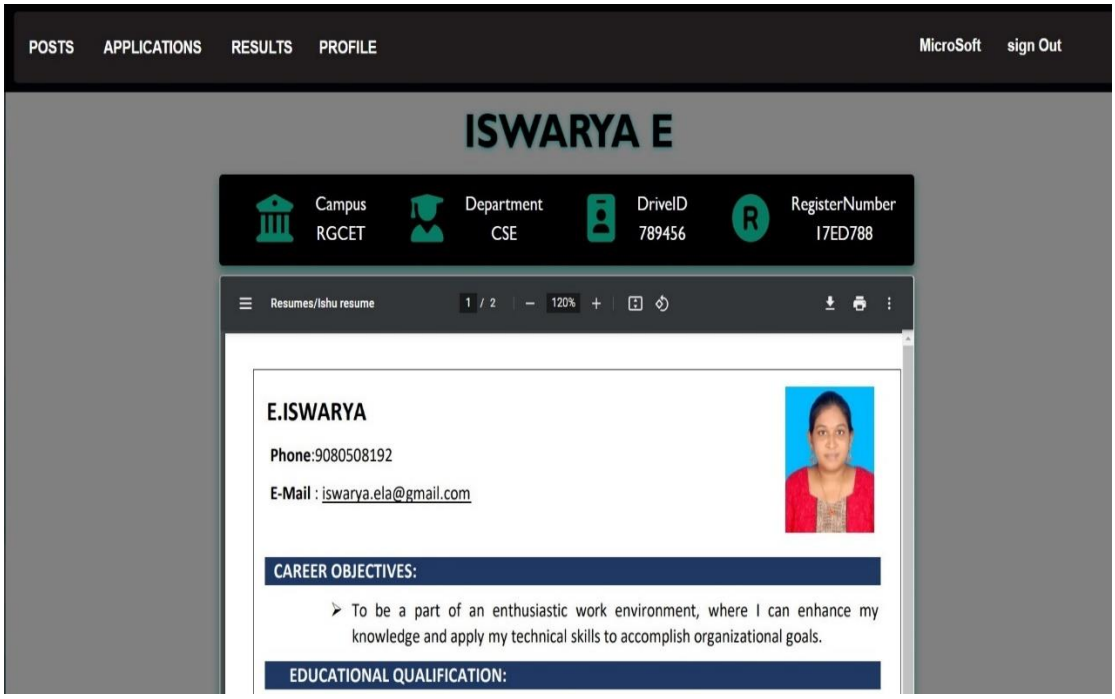
Company coordinator can post the drive with required details



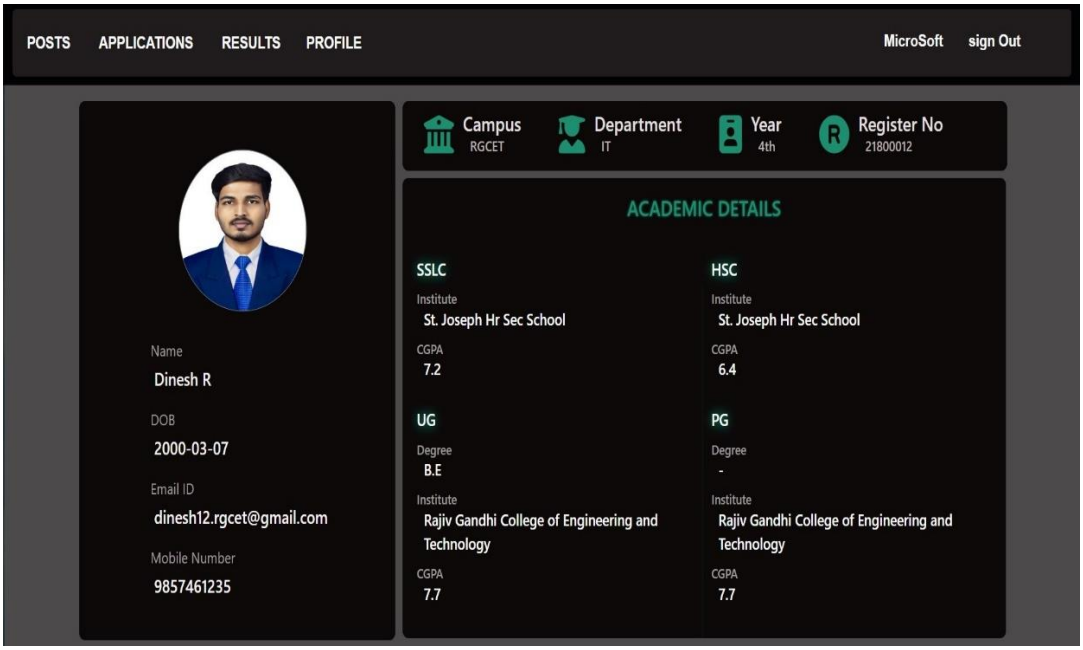
Company coordinator view the list of Job Applications



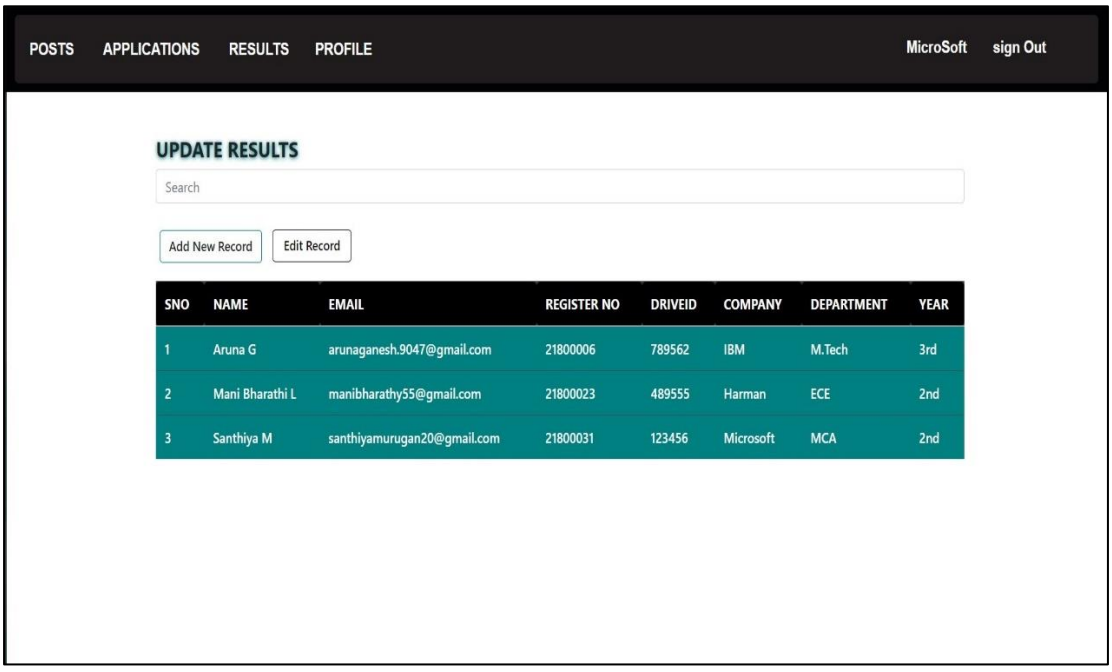
And view & download the Students Resumes to validate.



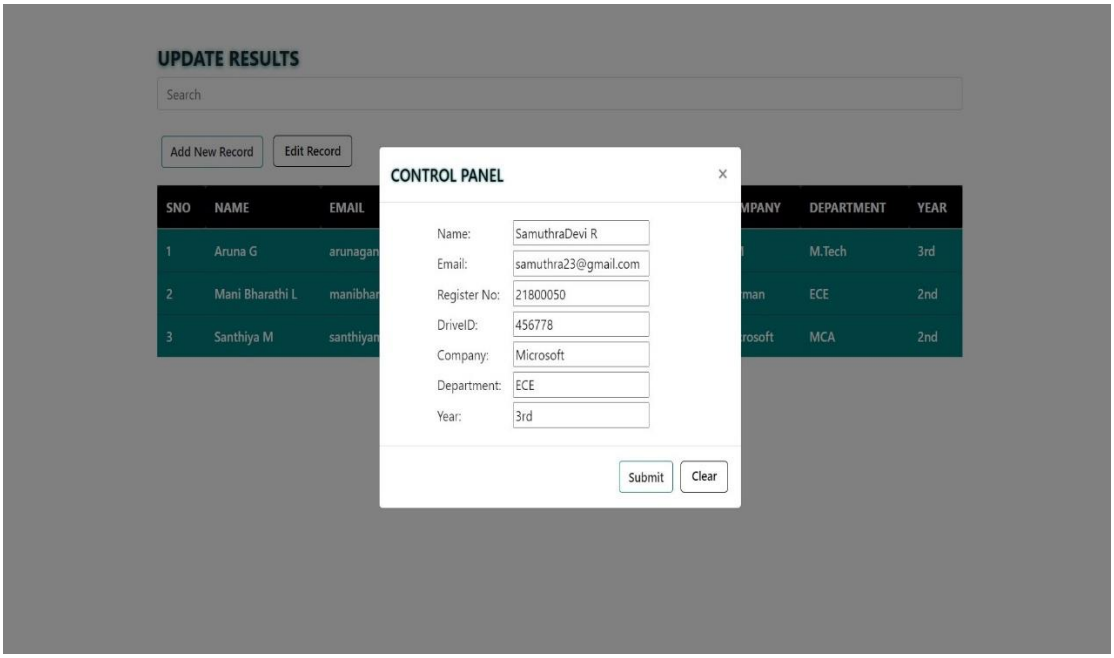
Company coordinator can also view the profile of applied students



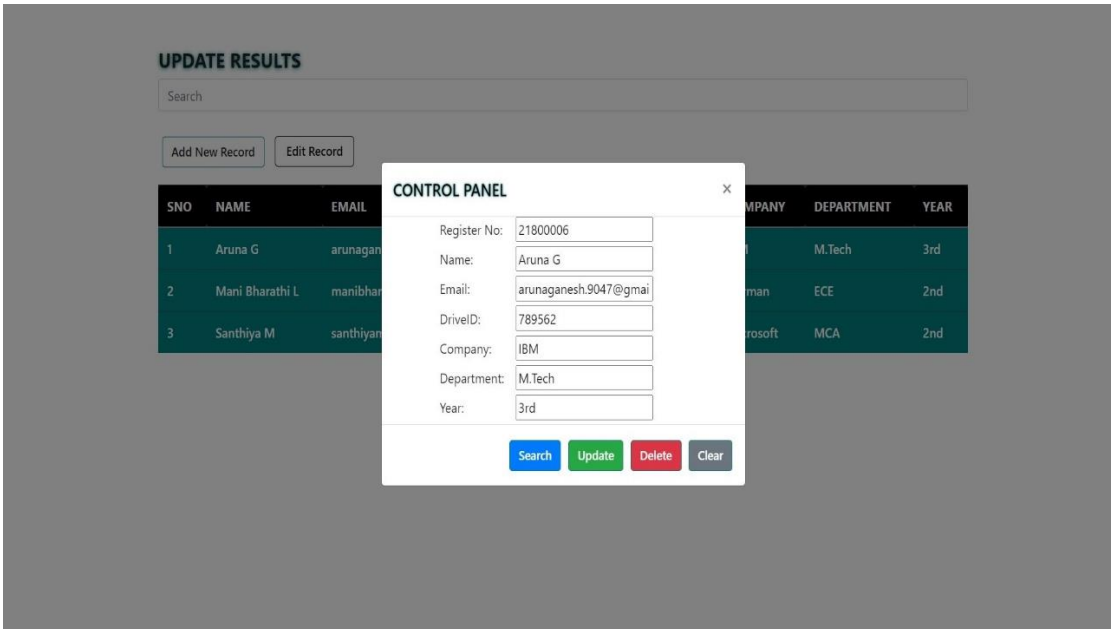
After completed interview process, company cordinator update results



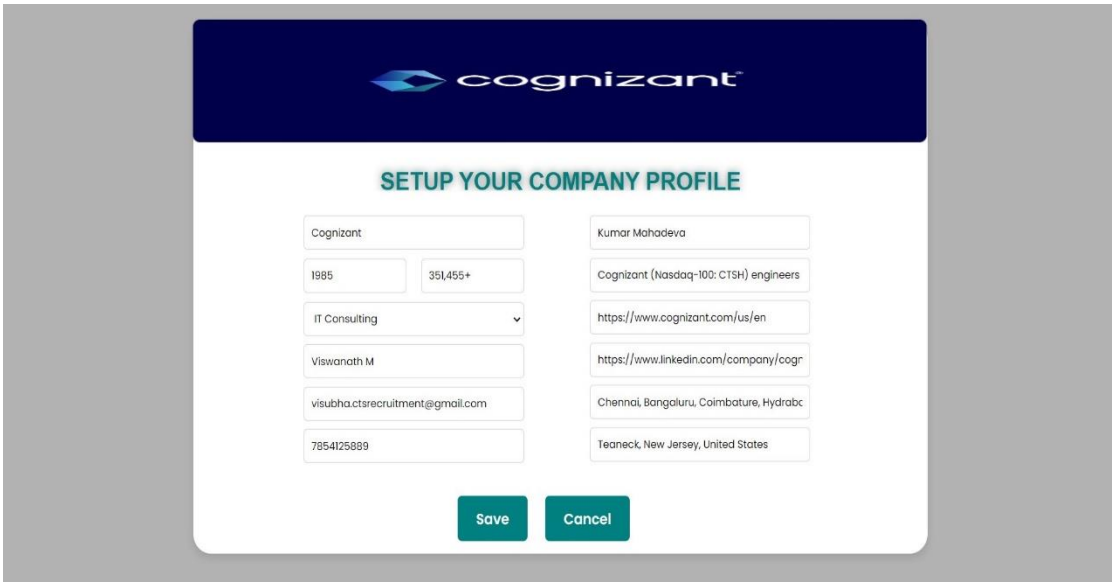
They can add new student details



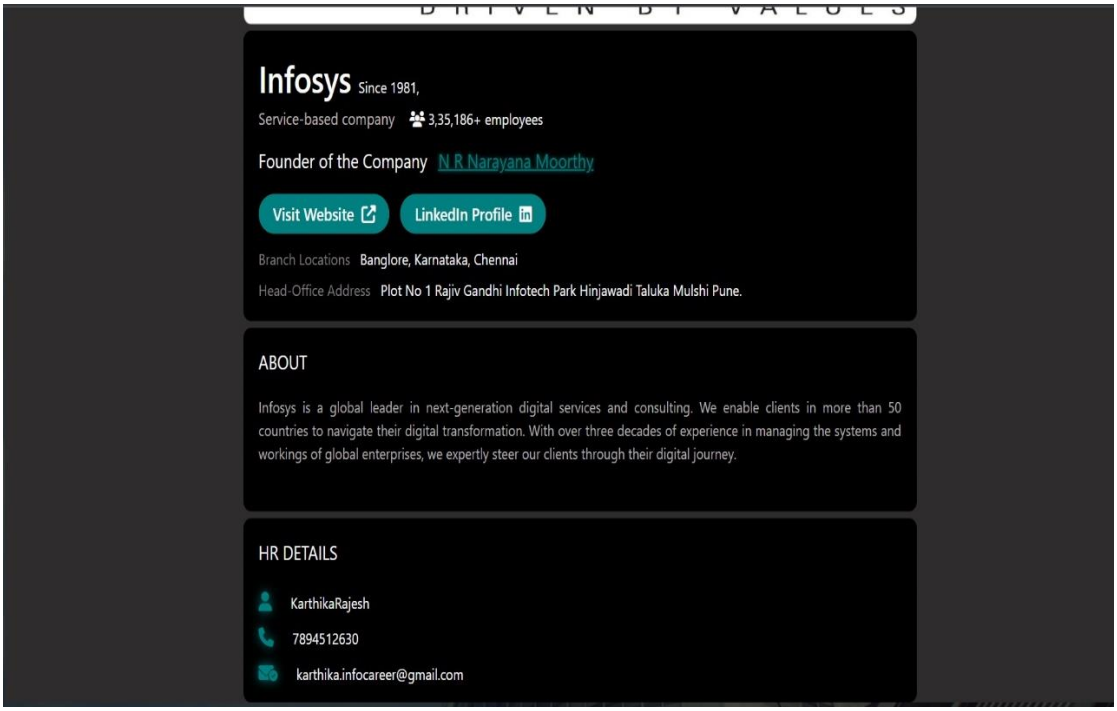
And can edit the existing record or delete the records



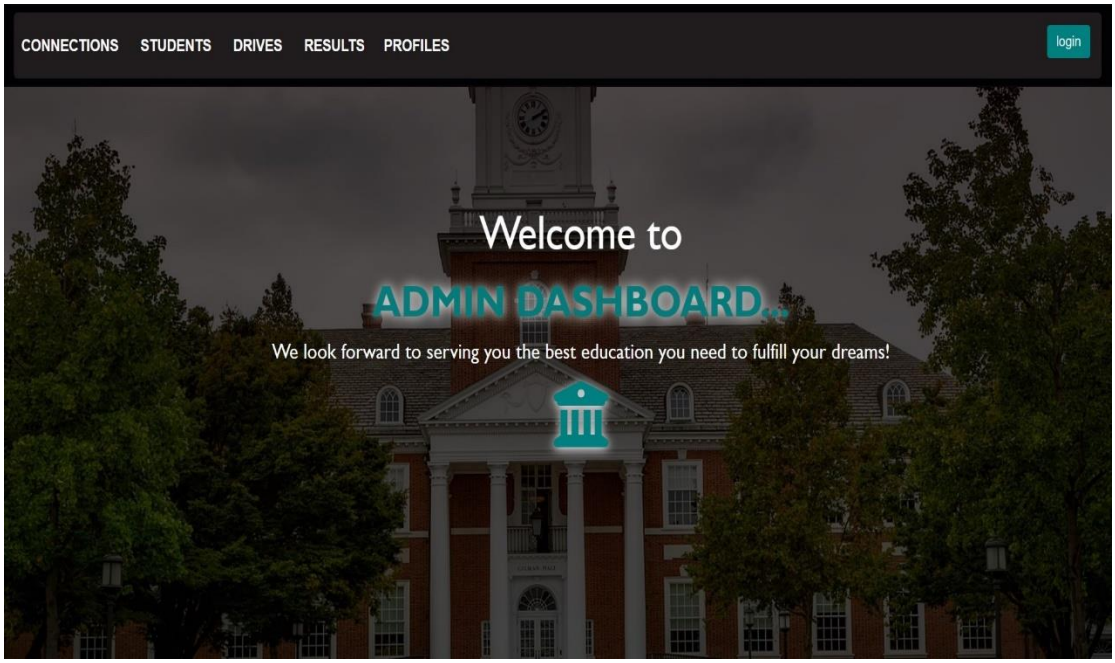
Company coordinator can add the company profile



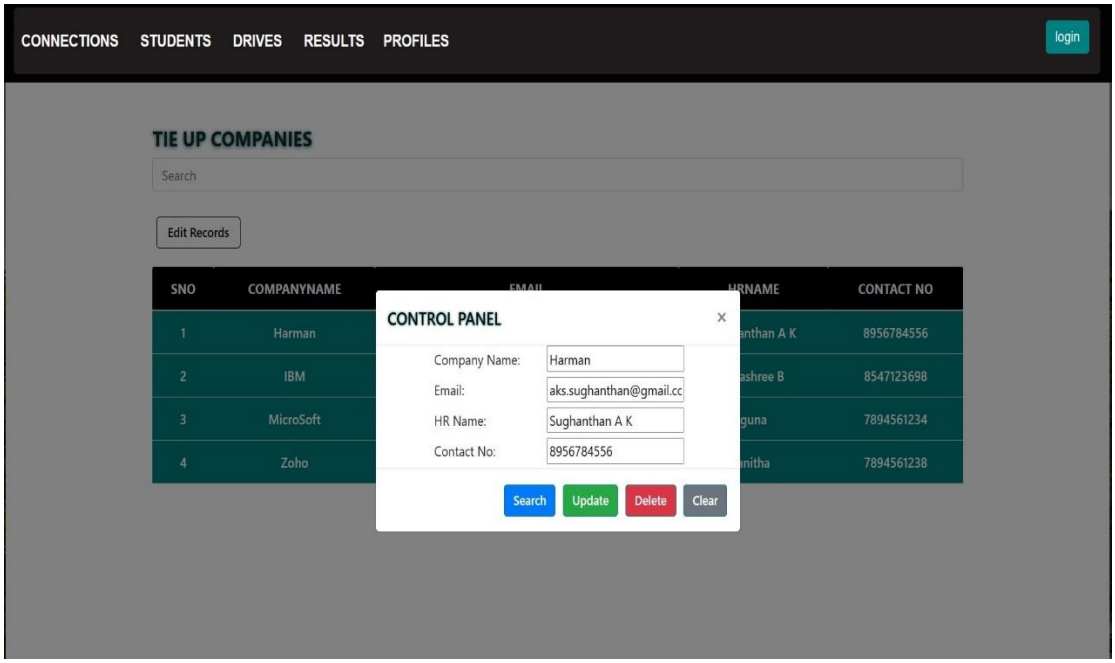
View of updated company profile



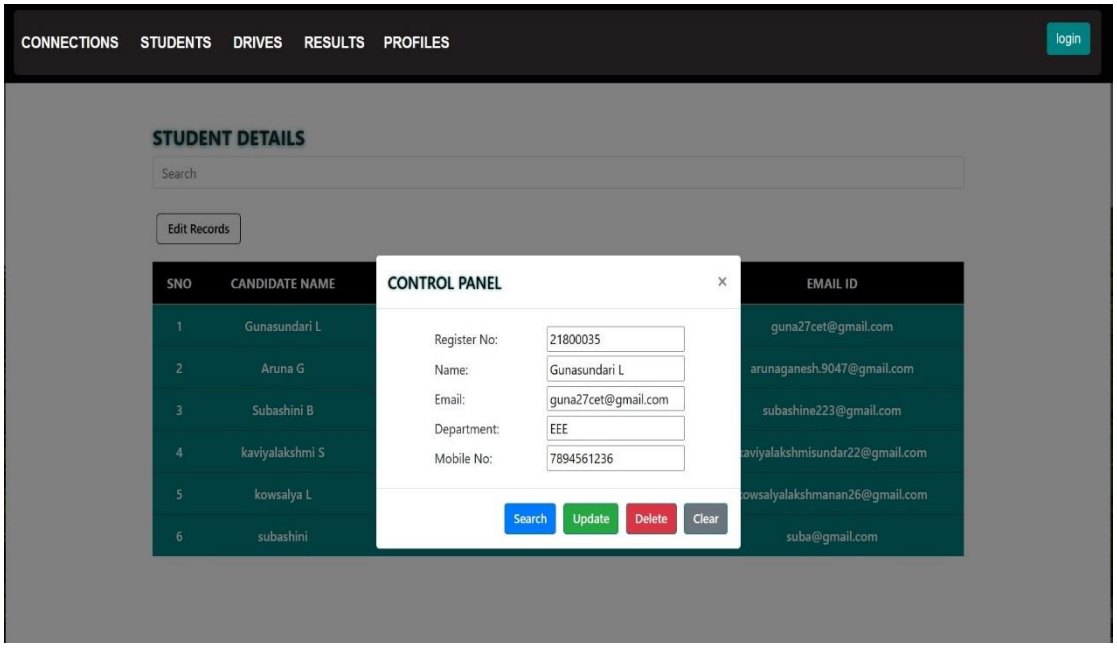
Admin Module



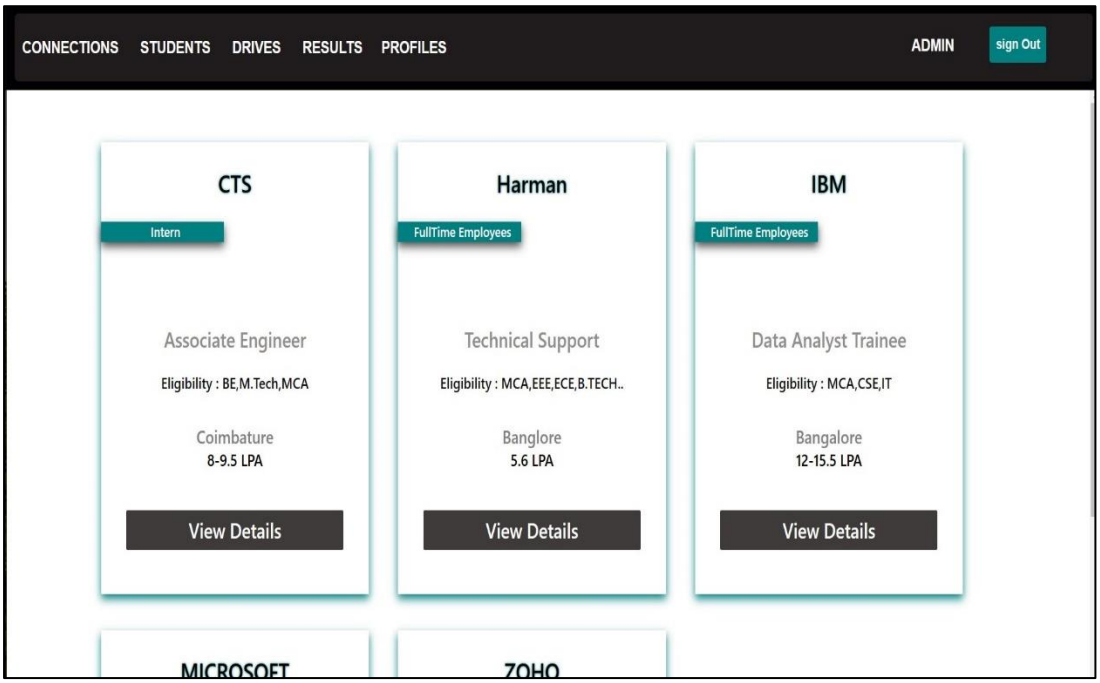
Admin can edit the company records



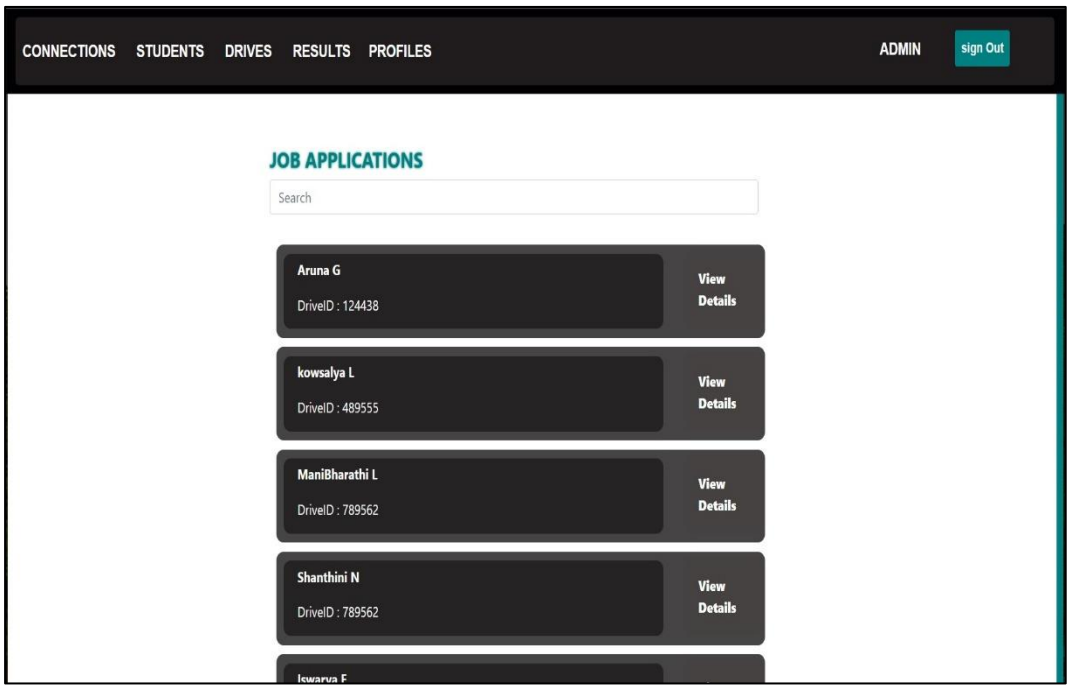
Admin can edit the student records as well



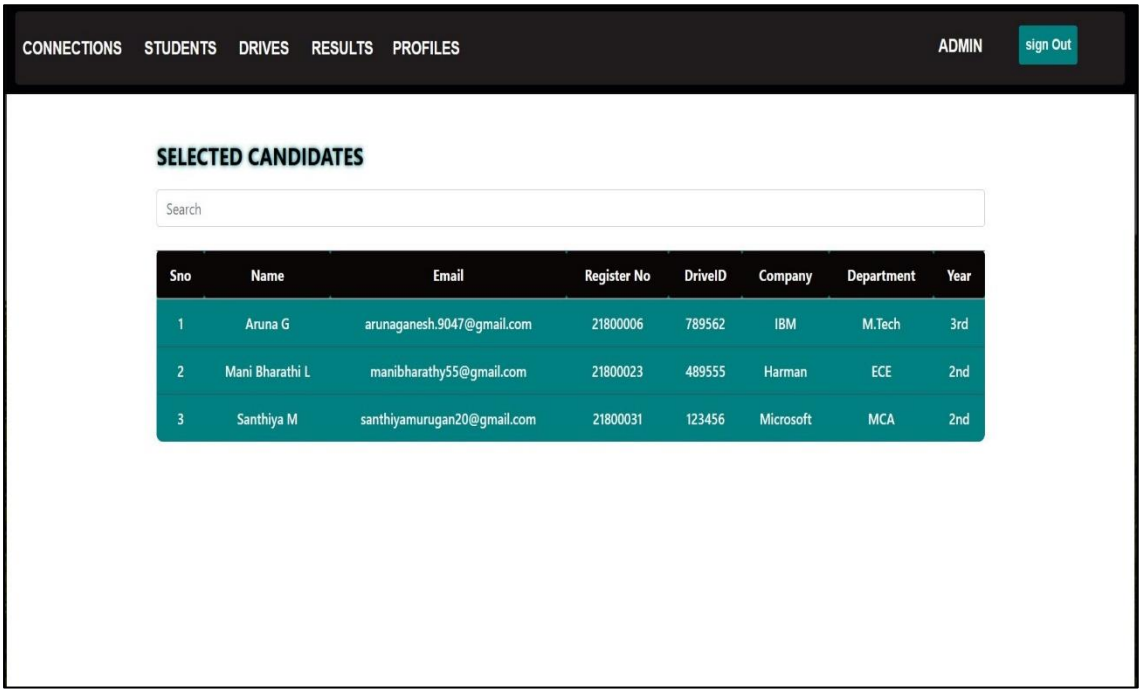
Admin can also view the Drive post and details



Admin view the students applications

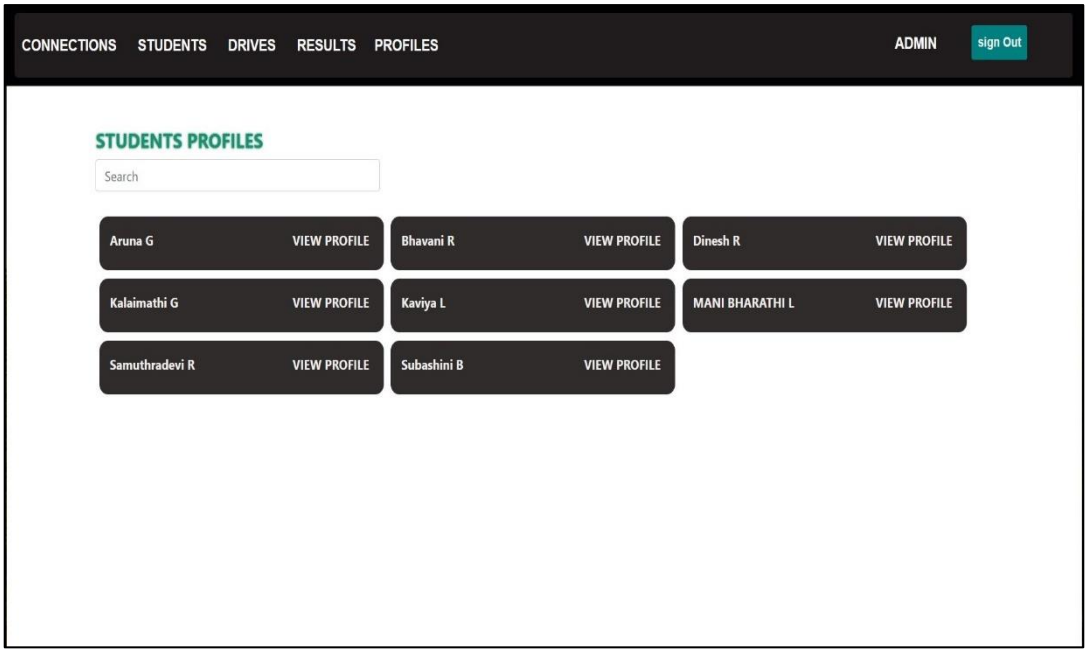


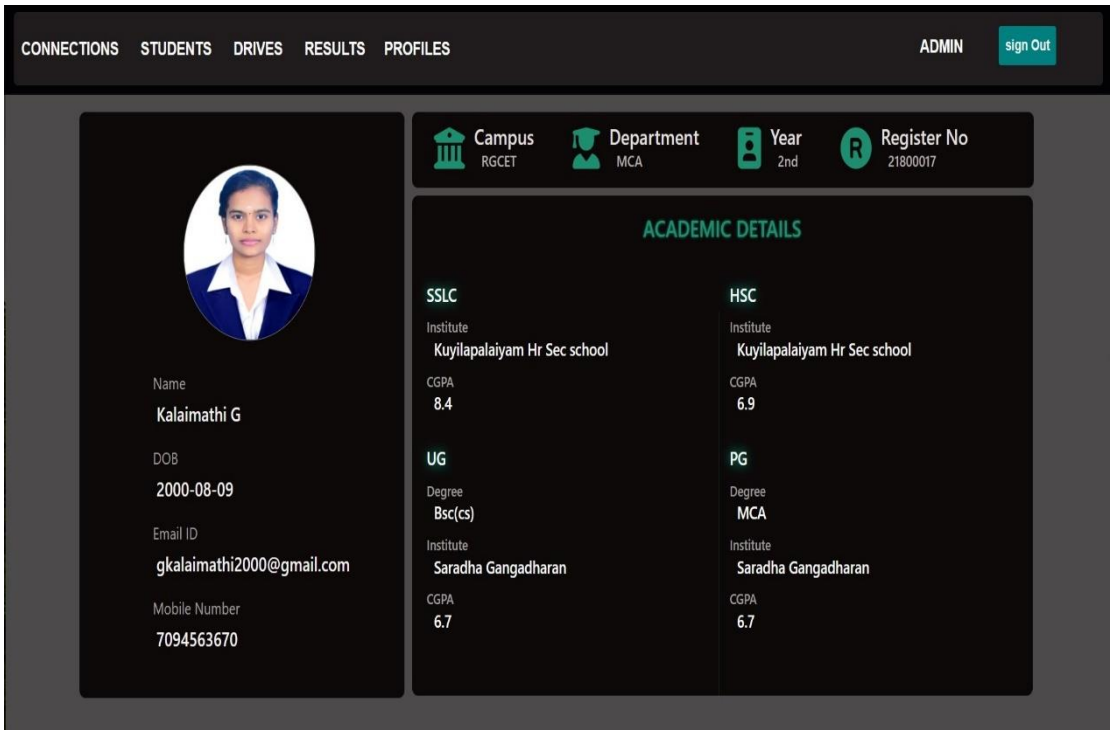
Selected students report will Show in admin panel



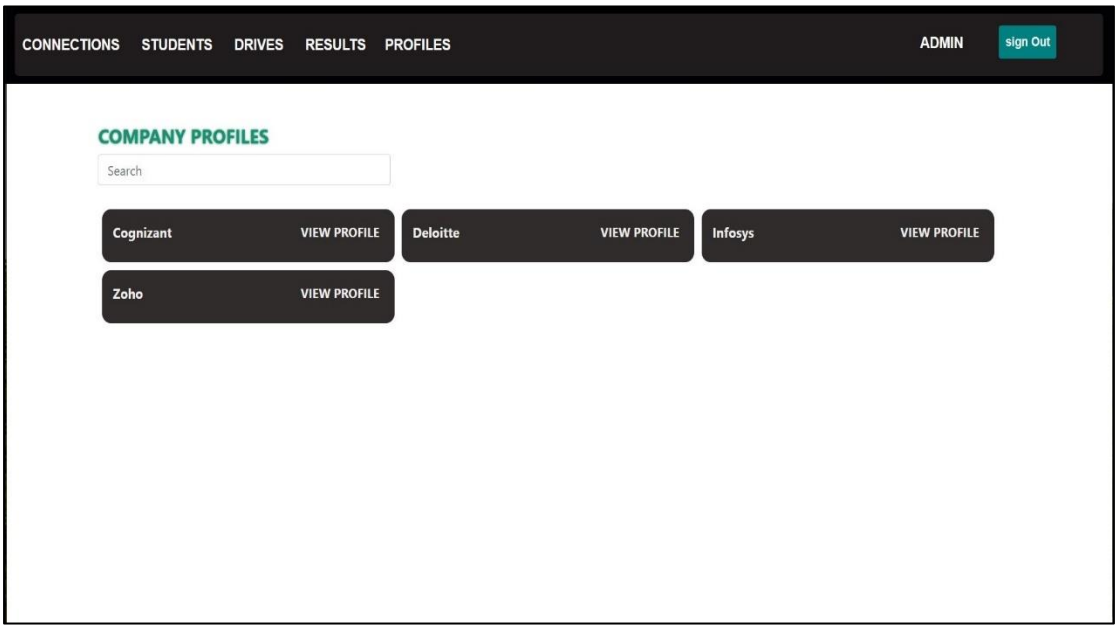
Sno	Name	Email	Register No	DriveID	Company	Department	Year
1	Aruna G	arunaganesh.9047@gmail.com	21800006	789562	IBM	M.Tech	3rd
2	Mani Bharathi L	manibharathy55@gmail.com	21800023	489555	Harman	ECE	2nd
3	Santhiya M	santhiyamurugan20@gmail.com	21800031	123456	Microsoft	MCA	2nd

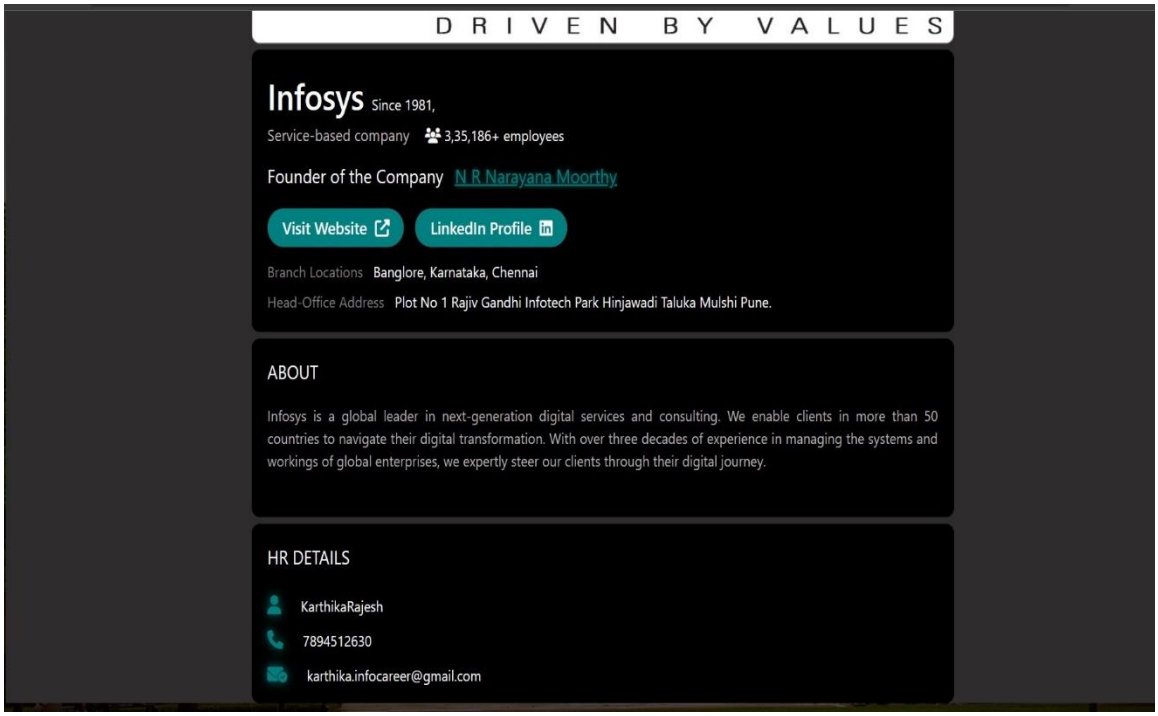
Admin can view the all students profiles who are updated in portal





Admin can view the all company profiles who are updated in portal





REFERENCES

1) **The Principles of Object-Oriented JavaScript** by Nicholas C. Zakas completely changes my understanding of JavaScript.
2) Jennifer Robbins' **Learning Web Design** A Beginner's Guide to HTML, CSS, JavaScript, and Web Graphics
3) Robin Wieruch's **The Road to Firebase** to master Firebase in Javascript.

WEB REFERENCES

- https://stackoverflow.com
- https://www.youtube.com
- https://firebase.google.com/?gclid=Cj0KCQiAq5meBhCyARIsAJrtdr5ZAfZqbUsmnKuqNobN4Fdb35uj11-iSAQ_nK_0lSGa7G_lgPx6EDcaAtaVEALw_wcB&gclsrc=aw.ds