# VISVESVARAYA TECHNOLOGICAL UNIVERSITY

**BELAGAVI, KARNATAKA**

**A MINI PROJECT REPORT**

**on**

**PLACEMENT MANAGEMENT SYSTEM**

Submitted in partial fulfillment of the requirement

for the V semester be in Information Science and Engineering

**DBMS MINI-PROJECT-18CSL58**

**Bachelor of Engineering**

In

**Information Science & Engineering**

***Submitted by***

**ANIMESH DWIVEDI 1SG20IS008**

**CHIRAG TILWANI 1SG20IS023**

***Under the guidance of***

**Prof. CHAITANYA V**

**Assistant Professor**

***Dept. of I.S.E, S.C.E***

**DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING**

# SAPTHAGIRI COLLEGE OF ENGINEERING

**Bengaluru-57**

**2022-23**



SAPTHAGIRI COLLEGE OF ENGINEERING

(Affiliated to Visvesvaraya Technological University, Belagavi & Approved by AICTE, New Delhi) (IAO9001-2015 and ISO14001-2015 certified Institute)

#14/5, Chikkasandra, Hesaraghatta Main Road, Bengaluru – 560057



**Department of Information Science & Engineering**



**CERTIFICATE**

This is to Certify that the DBMS Mini-Project work entitled **PLACEMENT MANAGEMENT SYSTEM** carried out by **Mr.Animesh Dwivedi(1SG20IS008) and Mr.Chirag Tilwani(1SG20IS023)** students of 5thsemester, department of **Information Science & Engineering** carried out at our college **Sapthagiri College of Engineering**, Bengaluru in partial fulfillment of the award of **Bachelor of Engineering** in **Information Science & Engineering** of the **Visvesvaraya Technological University,** Belagavi during the year 2022-23. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the Report deposited in the departmental library. The project report has been approved as it satisfies the academic requirements in respect of Mini- Project work prescribed for the said Degree.

|  |  |  |
| --- | --- | --- |
| **Signature of the Guide**  Prof. Chaitanya V Assistant Professor |  | **Signature of the HOD**  Dr H R Ranganatha HOD, I.S.E., S.C.E. |

|  |  |
| --- | --- |
| **External Viva**  **Name of the Examiners**  1.……………………………………  2.…………………………………… | **Signature of the Examiners with date**  1. ……………………………….  2. ………………………………. |

## ACKNOWLEDGEMENT

Any achievement doesn’t depend solely on the individual efforts but on the guidance, encouragement and co-operation of intellectuals, elders and friends. A number of personalities have helped us. We would like to take this opportunity to thank them all.

We would like to express our heart-felt gratitude to **Dr. H Ramakrishna**, Principal, Sapthagiri College of Engineering, Bengaluru, for his help and inspiration during the tenure of the course.

It is great privilege to extend our deep sense of gratitude to **Dr. H R Ranganatha**, Head of the Department, Information Science and Engineering, Sapthagiri College of Engineering, Bengaluru, who patronized throughout our career, for his constant support and encouragement and for the facilities provided to carry out this work successfully.

We wish to express our sincere thanks to our guides **Prof. Chaitanya V,** Assistant Professor, **Prof. Ramya R,** Assistant Professor, Information Science and Engineering, Sapthagiri College of Engineering, Bengaluru for helping us throughout and guiding us from time to time.

We also extend our sense of gratitude and sincere thanks to all faculty members and non-teaching staff members of Information Science and Engineering, Sapthagiri College of Engineering, Bengaluru for their views and encouraging ideas.

Finally, we also thank our family and friends for their co-operation and motivation.

**Animesh Dwivedi(1SG20IS008)**

**Chirag Tilwani(1SG20IS023)**

# ABSTRACT

The PLACEMENT MANAGEMENT SYSTEM is a web based application developed in windows platform for the placement department of the college in order to provide the details of its students in a database for the companies to their process of recruitment provided with a proper login. The system can used for college to manage the student information with regards to placement details .This project contains all the details of the students that can be viewed by all the users (read only), but can be modified only by the student with an authorized service.

# TABLE OF CONTENTS

### Chapter No Chapter Name Page No.

|  |  |  |
| --- | --- | --- |
| **1** | **INTRODUCTION** | **01** |
| 1.1 | Introduction to DBMS | 01 |
| 1.2 | Project Introduction | 01 |
| 1.3 | Objectives | 02 |
| **2** | **LITERATURE SURVEY** | **03** |
| 2.1 | Problem Statement | 03 |
| 2.2 | React JS | 03 |
| 2.3 | JavaScript | 03 |
| 2.4 | NodeJS | 03 |
| 2.5 | CSS | 04 |
| 2.6 | MySql | 04 |
| 2.7  2.7.1  2.7.2 | Requirement Specifications  Software Specifications  Hardware Specifications | 04  04  05 |
| 2.8 | Functional Requirements | 05 |
| **3** | **SYSTEM DESIGN** | **06** |
| 3.1 | Introduction | 06 |
| 3.2 | Schema Diagram | 06 |
| 3.3 | E.R Diagram | 08 |
| **4** | **IMPLEMENTATION** | **09** |
| 4.1 | Project Implementation | 09 |
| 4.2  4.2.1  4.2.2  4.2.3 | Pseudo Code  Front-end Code  Code to establish connection with database  Source code to setup port | 10  10  11  12 |
| 4.3 | Queries | 12 |
| 4.4  4.4.1  4.4.2 | Stored Procedures and Triggers  Stored Procedures  Triggers | 13  13  14 |
| **5** | **RESULTS** | **16** |
| 5.1 | Snapshots | 16 |

**Conclusion 19**

**Bibliography 20**

|  |  |  |
| --- | --- | --- |
|  | **LIST OF FIGURES** |  |
| **Figure No.** | **Figure Name** | **Page No.** |
| 3.2 | Schema Diagram | 07 |
| 3.3 | ER Diagram | 08 |
| 5.1 | Student Login Page | 16 |
| 5.2 | Home Page | 17 |
| 5.3 | Admin Dashboard | 17 |
| 5.4 | Companies Details | 18 |
| 5.5 | Placement Details | 18 |

|  |  |
| --- | --- |
|  | **LIST OF TABLES** |
| **Table No.** | **Table Name** |
| 4.1 | studentdetails Table |
| 4.2 | slogin Table |
| 4.3 | admindetails Table |
| 4.4 | alogin Table |
| 4.5 | companydetails Table |
| 4.6 | updateddrive Table |

**CHAPTER 1**

# INTRODUCTION

## 1.1 Introduction to DBMS

Income and Expense Management is a web application intended to run on any device namely smart phone or desktop. Income and Expense Management System is designed to efficiently cater the needs of users by eliminating imparting costs and settling vows to friends. The application encourages corresponding users help in who owes who, and for what. Aim is use better approaches to help users and their companions to share expenses easily. This new application will let bunch users and their companions to have detailed view inside this application around individual costs. The app allows its users to add a remark to an expense, click on the expense name in any expense list. Users can enter their expenses which they are spending daily or weekly basis. Hence it makes them to keep track on their expenses. Users can also enter the receivables which they are supposed to receive. Expenses Viewing list is enabled where user can view the expenses list which they have entered earlier. Along with the expenses title users can also add the image attachment of their bill.

**1.2 Project Introduction**

Placement Management System is a total management and informative system, which provides the up-to date information of all the students in a particular college. Placement Management System helps the colleges to overcome the difficulty in keeping records of hundreds and thousands of students and searching for a student eligible for recruitment criteria from the whole thing. It helps in effective and timely utilization of the hardware and the software resources.

The administrator will create the users and the users will use the accounts created by administrator. When the user enters into his respective page he can update his details, and the details are to be approved by the administrator. All the users have some common services like changing password, updating details, searching for details, checking the details, mailing to administrator, and reading the material uploaded by admin if the user is a student. Administrator can upload materials, search for student details, and he has the right to approve the students.

**1.3 Objectives**

The main objective of the Project on Income and Expense Management System is to manage the details of Expense, Category, Type, User, Registration. It manages all the information about Expense, Payment. Registration, Expense. The project is totally built at administrative end and thus only the administrator is guaranteed the access. The purpose of the project is to build an application program to reduce the manual work for managing the Expense, Category. Payment. Type. It tracks all the details about the Type, User, Registration.

## 2.4 Node Js

Node.js is an open source and cross-platform runtime environment for executing JavaScript code outside of a browser. NodeJS is not a framework and it’s not a programming language. Node.js is generally used for building back-end services like APIs like Web App or Mobile App. NodeJS is easy to learn and code because it uses JavaScript.

As Node.js is asynchronous programming we don’t need to wait for one request to complete fully, it’s simply take the next request so there is no waiting time and that’s is the best part of the Node.js. Node.js is very memory efficient because of its features.

## 2.3 Javascript

JavaScript is a very powerful client-side scripting language. JavaScript is used mainly for enhancing the interaction of a user with the webpage. In other words, you can make your webpage livelier and more interactive, with the help of JavaScript. JavaScript is also being use widely in game development and mobile application development.

**CHAPTER 2**

# LITERATURE SURVEY

## Problem Statement

The problem that this project aims to combat is dealing with the huge database for the complete details of the students as well as Companies in the Placement process.

### 2.2 React Js

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. React Router is an example of such a library.

## 

## 

## 

## 

## 2.5 Css

Cascading Style Sheets (CSS) is a stylesheet language used to describe the presentation of a document written in HTML or XML. CSS describes how elements should be rendered on screen, on paper, in speech, or on other media.

CSS is designed to enable the separation of presentation and content, including layout, colors, 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, and reduce complexity and repetition in the structural content.

**2.6 MySql**

MySQL is an open-source relational database management system. Its name is a combination of "My", the name of co-founder Michael Widenius's daughter, and "SQL", the abbreviation for Structured Query Language. A relational database organizes data into one or more data tables in which data types may be related to each other; these relations help structure the data.

## 2.7 Requirement Specification

### 2.7.1 Software Specification

* + - * Operating system: Windows 10, Mac OS
      * Front End: Java Script, ReactJS, CSS
      * Back End: NodeJS
      * Database: Mysql

## 2.7.2 Hardware Specification

* + - * Processor:x86 compatible processor with 1.7 GHz Clock Speed
      * RAM:512 MB
      * Hard Disk: 20 GB or greater
      * Monitor: VGA/SVGA
      * Key Board: 104 keys standard
      * Mouse: 2/3 button. Optical/mechanical

**2.8 Functional Requirement**

* Displays all the students in the database.
* Displays company details in the database.
* Displays each student’s detail with name, USN, CGPA, branch and personal details. Performs update, insert operations.

**CHAPTER 3**

The design of the database is called a schema. This tells us about the structural view of the database. It gives us an overall description of the database. A database schema defines how the data is organized using the schema diagram. A schema diagram is a diagram which contains entities and the attributes that will define that schema. A schema diagram only shows us the database design. It does not show the actual data of the database. Schema can be a single table or it can have more than one table which is related. The schema represents the relationship between these tables.

## Introduction

System design is the process or art of defining the architecture, components, modules, interfaces and data for a system to satisfy specifying requirements. One could see it as application of systems theory to product development.

This project is implemented using ReactJs, CSS, JavaScript which is proven to be efficient front-end tools. For backend, we used Mysql, which is one of the most widely used SQL database.

## 3.2 Schema Diagram

An entity-relationship model or the ER Diagram describes inter-related things of interest in a specific domain of knowledge. An ER model is composed of entity types and specifies relationships that can exist between instances of those entity types.

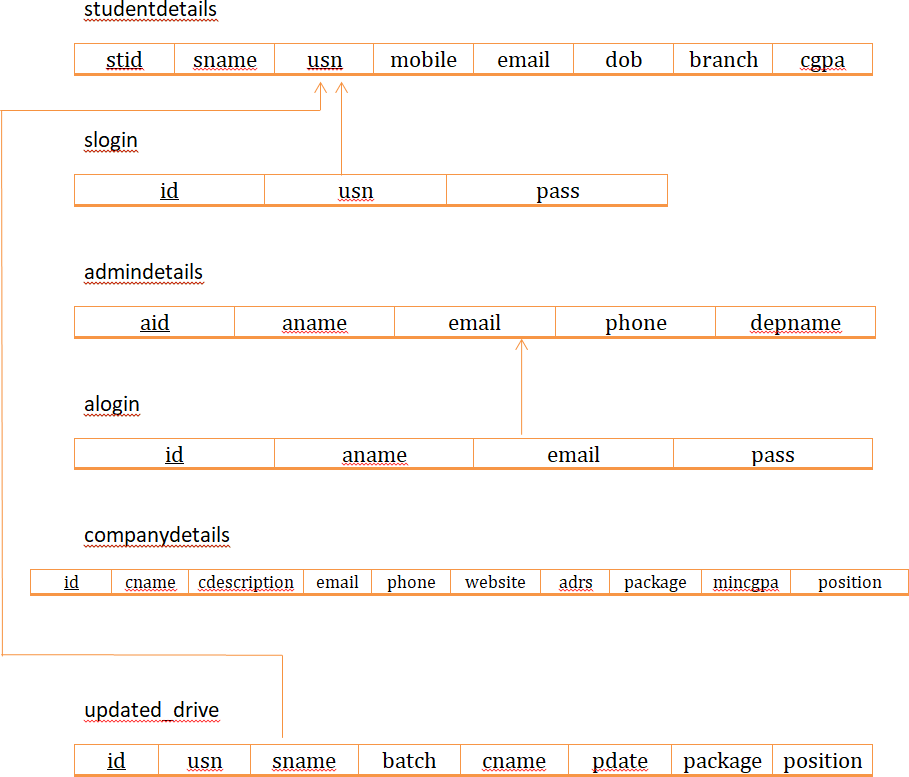
In engineering an ER model is commonly formed to represent things that a business needs to remember in order to perform business processes. Consequently, the ER model becomes an abstract model that defines a data or information structure that can be implemented in a database, typically a relational database.

Relationships are represented by diamond-shaped box. Name of the relationship is written inside the diamond-box. All the entities (rectangles) participating in a relationship, are connected to it by a line.

## SCHEMA

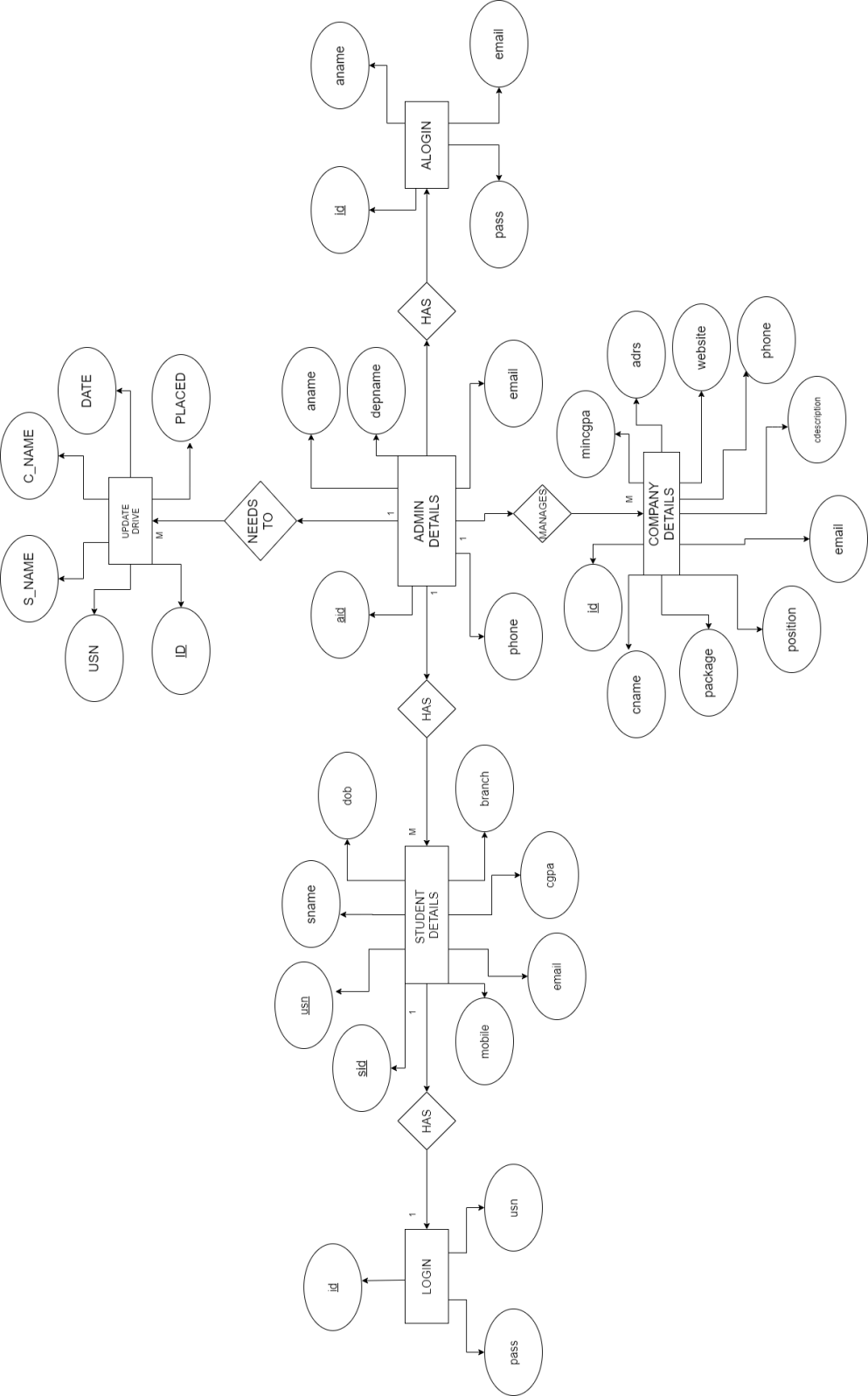
# SYSTEM DESIGN

**SCHEMA DIAGRAM**



**Fig 3.2 : Schema Diagram**

**3.3 ER DIAGRAM**



**Fig 3.3 ER Diagram**

**CHAPTER 4**

# IMPLEMENTATION

## 4.1 Project Implementation

We first started by discussing about out project and the ways in which we could implement it. We then decided on the front end and back end to be used. Then we started collecting the necessary data for our project. Once all the data was ready, we used the appropriate platform to execute our queries.

As project managers and sub-project managers we had to make sure that we

* Take action, in-line with plan and / or contract.
* Record and document all the work, work results, special events, decisions
* Analyze, communicate, report and document status and result of action
* Take decision if and what kind of change we need; in case any result is not made.
* Implement agreed changes, in-line with the plan and/ or contract.

## 4.2 Pseudo Code

* + 1. **Code for front end**

<https://github.com/chiragtilwani/dbms-pms-frontend.git>

* + 1. **Code to establish connection with database**

app.use(cors()); app.use(express.json());

const db = mysql.createConnection({ user: "root",

host: "localhost", password: "12345", database: "pms1",

});

* + 1. **Source code to setup port**

////////////////////S E R V E R P O R T SETUP///////////////

app.listen(3001, () => {

console.log("hurrayy , server running on port 3001");

});

* 1. **Queries**

//////////////////GET REQUEST TO SHOW/READ DATA FOR STUDENTS//////////////

app.get("/students", (req, res) => {

db.query("SELECT \* FROM studentdetails", (err, result) => { if (err) {

console.log(err);

}

///////////////ROUTE FOR REGISTERATION /////////////

app.post("/register", (req, res) => { const usn = req.body.usn;

const pass = req.body.pass;

db.query(

"INSERT INTO slogin (usn,pass) VALUES (?,?)", [usn, pass],

(err, result) => { if (err) {

console.log(err);

res.send({ err: err }); return;

}

if (result) { res.send(result);

} else {

res.send({ message: "already exists" });

}

}

);

});

db.query(

"SELECT \* FROM slogin WHERE usn = ? AND pass = ?", [usn, pass],

(err, result) => { if (err) {

res.send({ err: err });

}

if (result.length > 0) { res.send(result);

} else {

res.send({ message: "Wrong username/password combination" });

}

}

);

});

/////////////////////ROUTE FOR ADMIN LOGIN ///////////// app.post("/admin", (req, res) => {

const email = req.body.email; const pass = req.body.pass;

db.query(

"SELECT \* FROM alogin WHERE email = ? AND pass = ?", [email, pass],

(err, result) => { if (err) {

res.send({ err: err });

}

}

}

);

});

/////////////////////ROUTE FOR ADD COMPANIES /////////////

app.post("/addcompany", (req, res) => { const cname = req.body.cname;

const cdescription = req.body.cdescription; const email = req.body.email;

const phone = req.body.phone; const website = req.body.website; const adrs = req.body.adrs;

const package = req.body.package; const mincgpa = req.body.mincgpa; const position = req.body.position;

db.query(

"INSERT INTO companydetails (cname,cdescription,email,phone,website,ad rs,package,mincgpa,position) VALUES (?,?,?,?,?,?,?,?,?)",

[

cname, cdescription, email,

phone, website, adrs, package, mincgpa, position,

]

/////////////////////ROUTE FOR ADD PLACEMENTS /////////////

app.post("/addplacement", (req, res) => { const sname = req.body.sname;

const usn = req.body.usn; const batch = req.body.batch;

// const cgpa = req.body.cgpa; const cname = req.body.cname; const pdate = req.body.pdate; const package = req.body.package;

const position = req.body.position;

db.query(

"INSERT INTO updateddrive (sname,usn,batch,cname,pdate,package,positio n) VALUES (?,?,?,?,?,?,?)",

[sname, usn, batch, cname, pdate, package, position], (err, result) => {

if (err) { console.log(err); res.send({ err: err });

// res.send({ message: "Wrong username/password combination" });

}

if (result.length > 0) { res.send(result);

} else {

res.send({ message: "already exists" });

}

}

);

});

///////////////ROUTE FOR ADD STUDENTS /////////////

### Stored Procedures and Triggers

* + 1. **Stored Procedures**

A stored procedure is a set of Structured Query Language (SQL) statements with an assigned name, which are stored in a relational database management system as a group, so it can be reused and shared by multiple programs. Stored Procedures are used in the database to run a certain line or lines of code multiple times by calling that procedure in any number of .php files, any number of times.

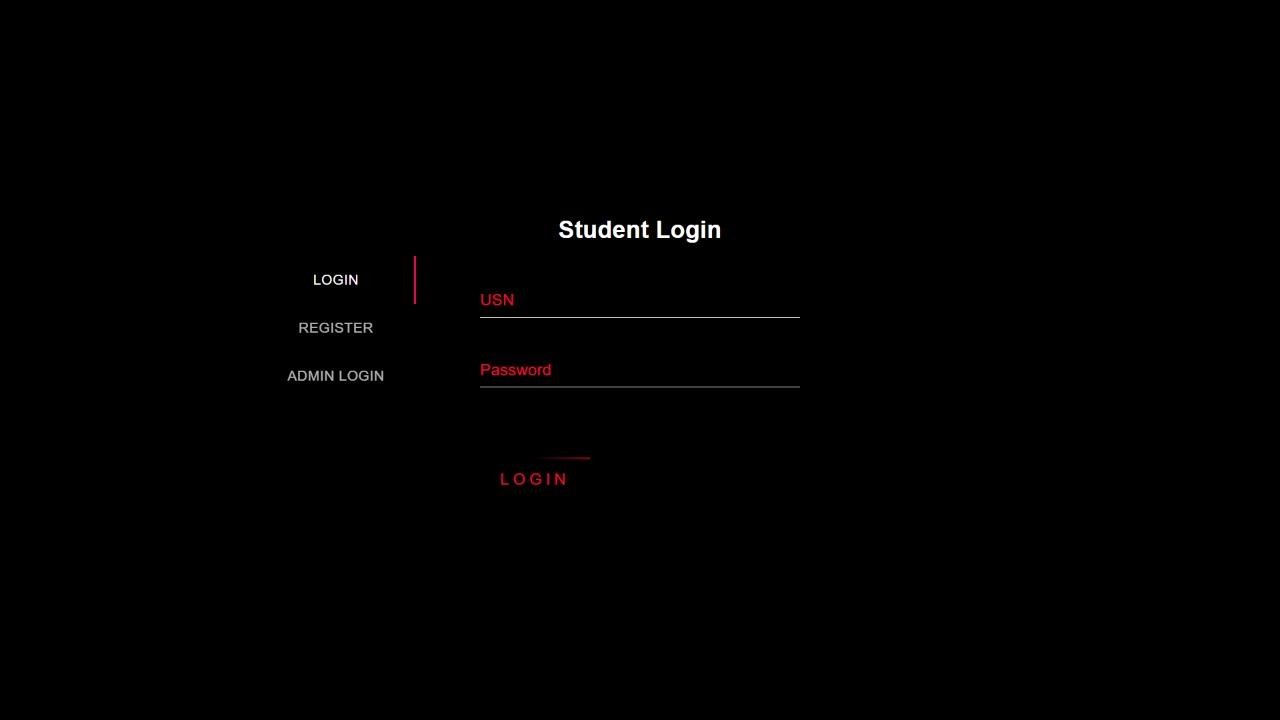
* + 1. **Triggers**

A trigger is a stored procedure in database which automatically invokes whenever a special event in the database occur s. For example, a trigger can be invoked when a row is inserted into a specified table or when certain table columns are being updated.

* Triggers created for this project is
* Alert when a player is deleted
* Alert when player details is updated
* Alert when player details is inserted

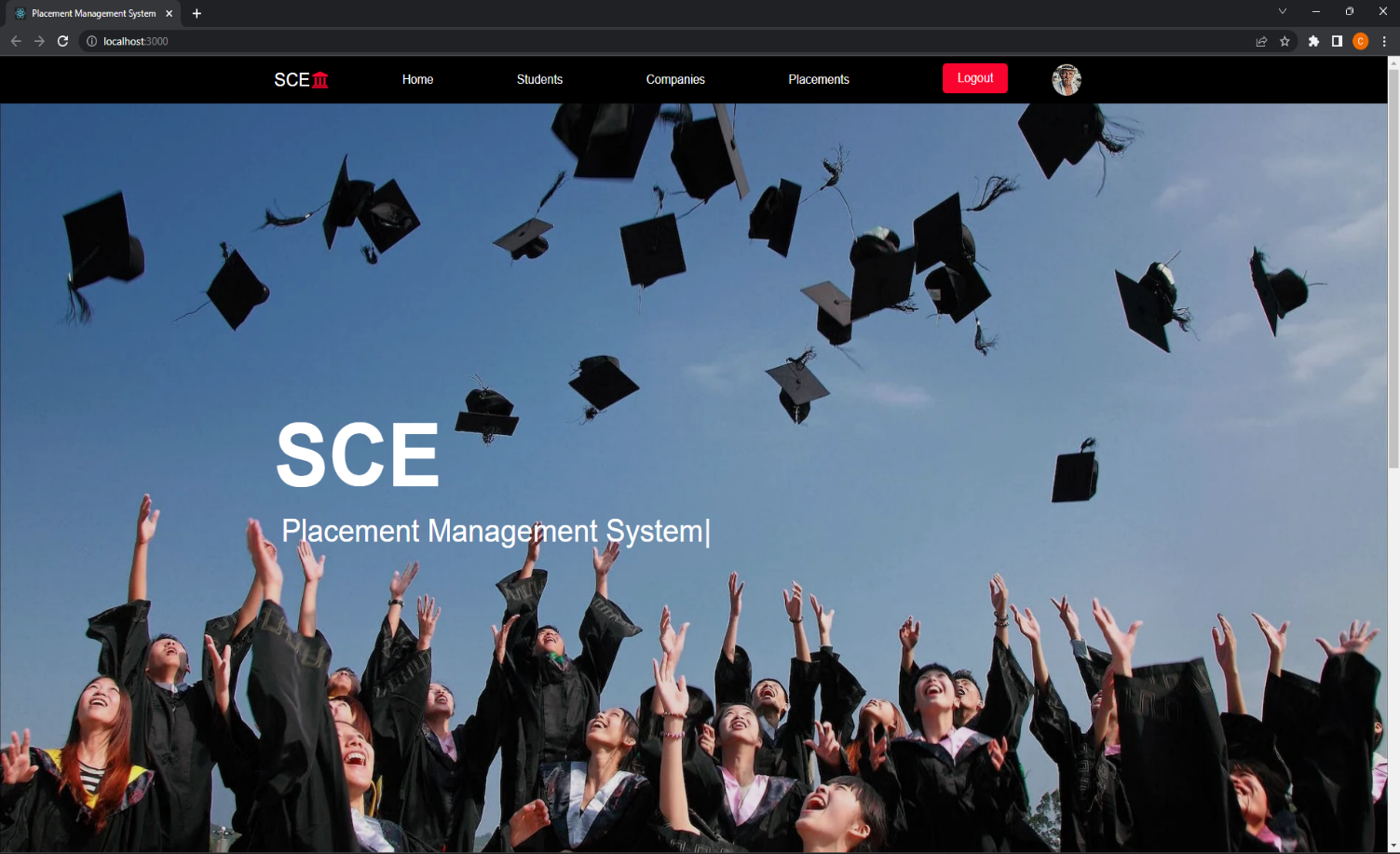
**CHAPTER 5**

**SNAPSHOTS**



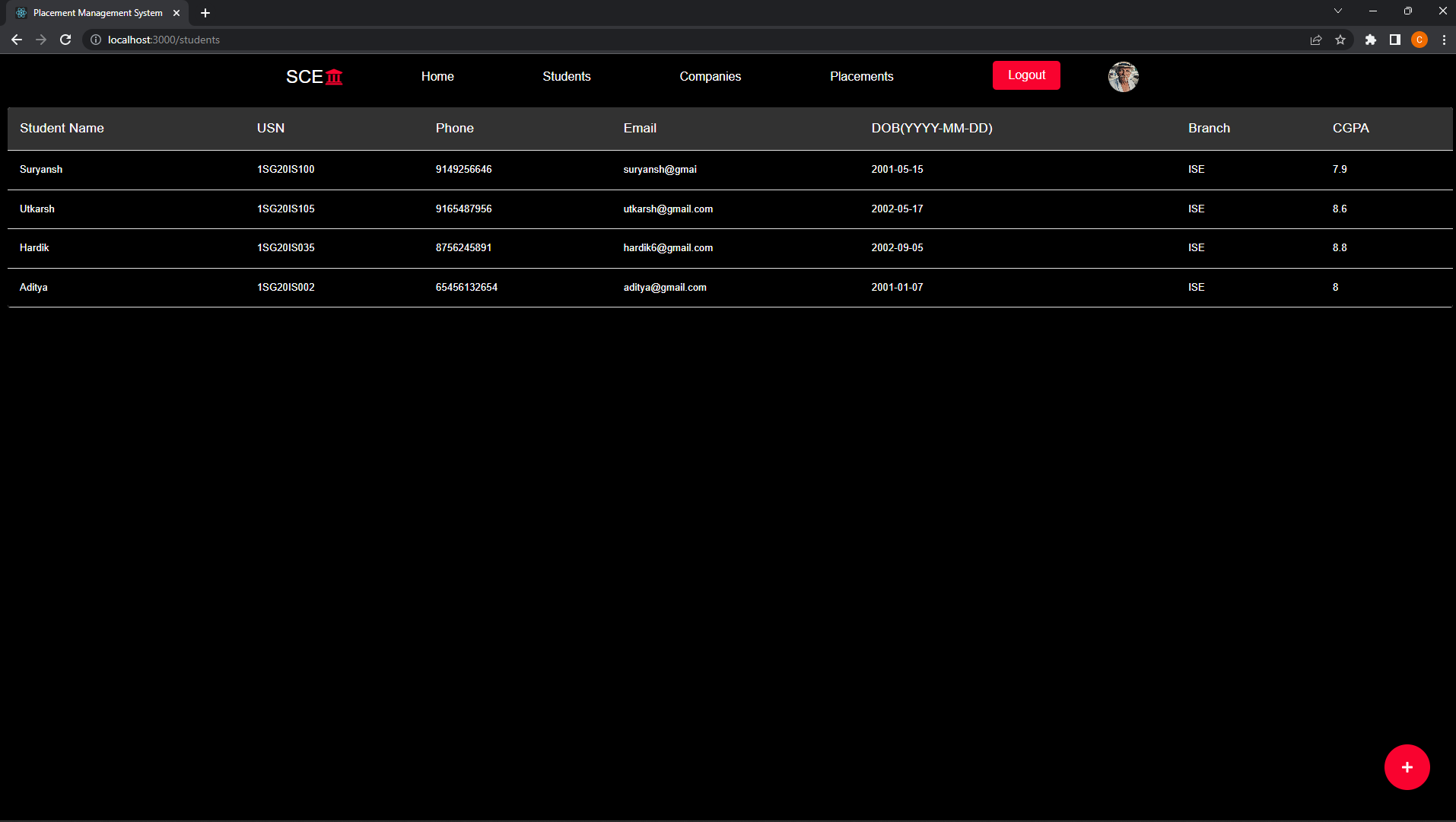
**Fig 5.1 : Student login page**

* This is the landing page of **PLACEMENT MANAGEMENT SYSTEM**.
* Here student can login filling his/her credentials.



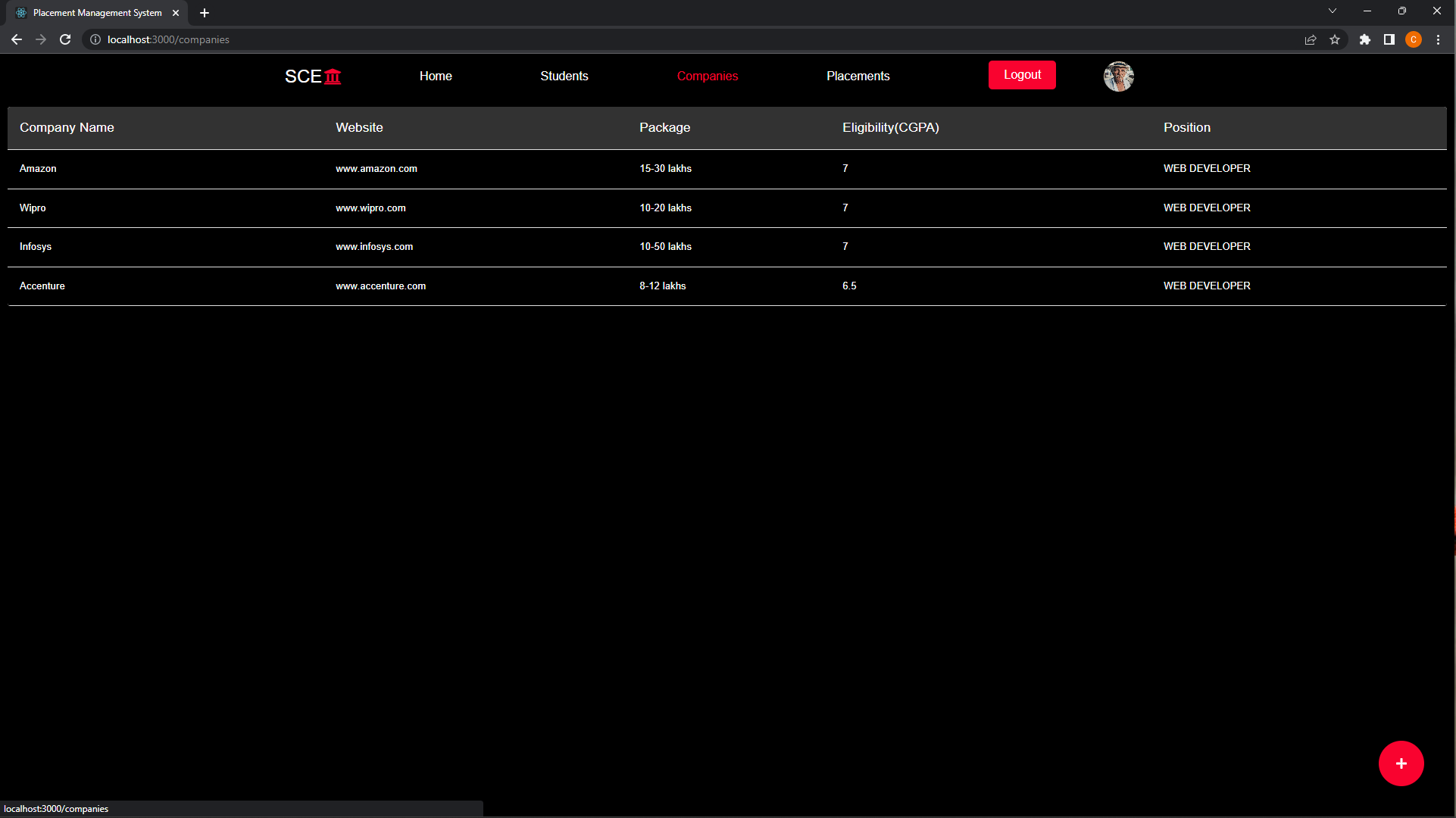
**Fig 5.2 :Home page**

* This is **HOME** page of **PLACEMENT MANAGEMENT SYSTEM.**



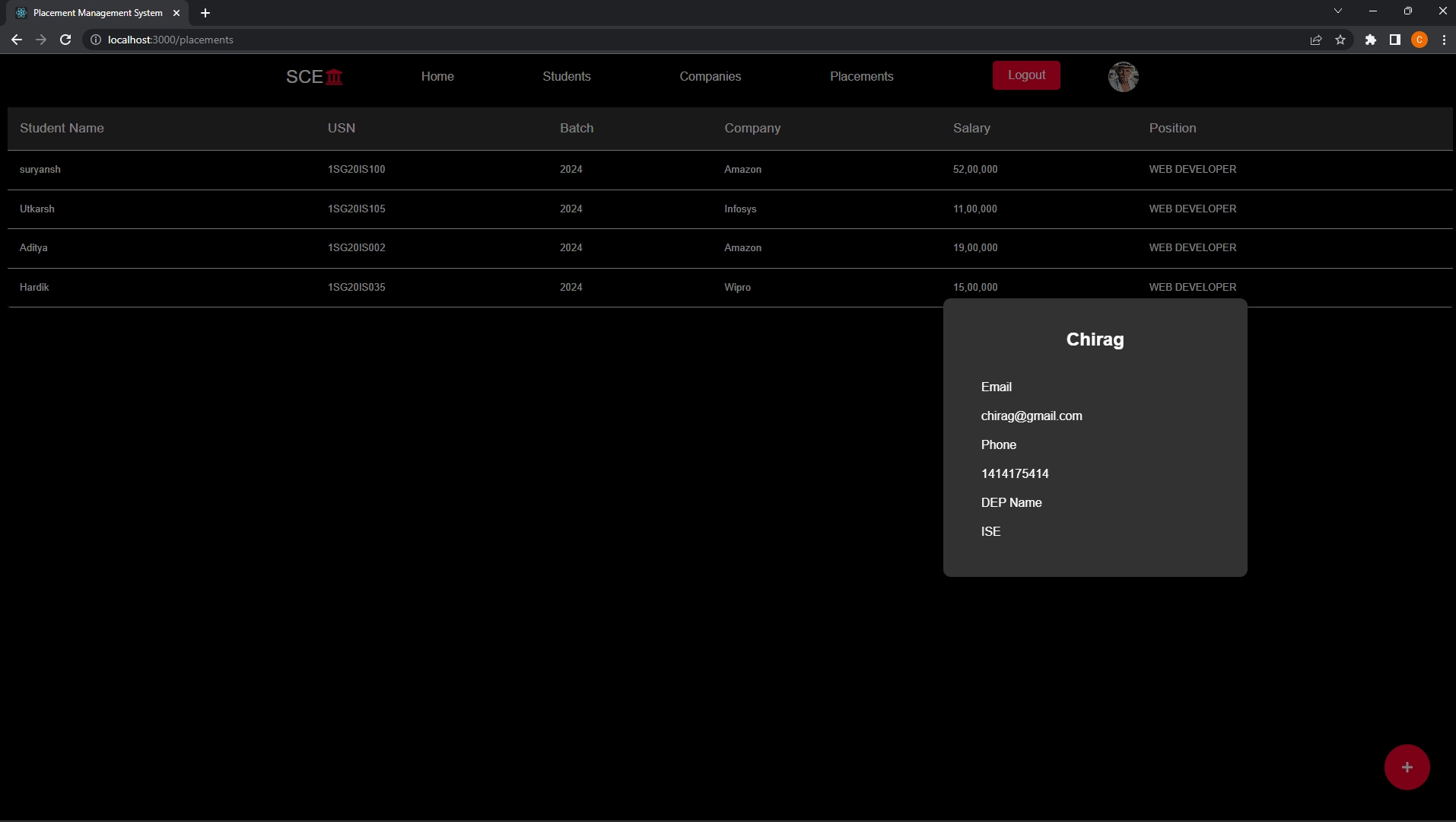
**Fig 5.3 : Admin Dashboard**

* This is **ADMIN DASHBOARD** of **PLACEMENT MANAGEMENT SYSTEM**.



**Fig 5.4 : Companies details**

* This page has details of all companies visited our college.



**Fig 5.5 :Placement details**

* This page shows details of students got placed.

**CONCLUSION**

In the existing system maximum work goes manually and it is error prone system, takes time for any changes in the system. The big problem is the searching and updation of the student data and also no any notification method available for giving information to student expect the notice board.

The proposed placement management system gives the automation in all the process like registration, updating, searching. It provides the detail solution to the existing system problem.

**BIBLIOGRAPHY**

1 .[www.w3schools.com](http://www.w3schools.com)

2.[www.stackoverflow.com](http://www.stackoverflow.com)

3.<https://codepen.io/>

4.www.github.com

5.[www.youtube.com](http://www.youtube.com)