

Visvesvaraya Technological University

Belagavi, Karnataka, 590 014.



A Mini Project Report on

“College Placement Management System”

Submitted in partial fulfillment of the requirements for the award of

Bachelor of Engineering

in

Computer Science and Engineering

Semester V

(18CSL58)

Academic Year 2020-21

Submitted By

Ms. Rashmi Ramakantha Pai USN: 2KE18CS032

Under the Guidance of

Prof. Shridhar Allagi

Department of Computer Science & Engineering



K. L. E. SOCIETY'S

K. L. E. INSTITUTE OF TECHNOLOGY,

Opp. Airport, Gokul, Hubballi-580 030

Phone: 0836-2232681

Website: www.kleit.ac.in





K. L. E. SOCIETY'S

K. L. E. INSTITUTE OF TECHNOLOGY,**Opp. Airport, Gokul, Hubballi-580 030**

Phone: 0836-2232681

Website: www.kleit.ac.in

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING****CERTIFICATE**

Certified that the mini project work entitled “**College Placement Management System**” is a bonafide work carried out by **Ms. Rashmi Ramakantha Pai** bearing USN number **2KE18CS032** in partial fulfillment for the award of degree of **Bachelor of Engineering** in **V Semester, Computer Science and Engineering** of **Visvesvaraya Technological University**, Belagavi, during the year **2020-21**. It is certified that all corrections/suggestions indicated for internal assessment have been incorporated in the report deposited in the department library. The mini 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. Shridhar Allagi)****Signature of the HOD****(Dr. Yerriswamy T.)****Signature of the Principal****(Dr. B. S. Anami)****Name of the Examiners****1.****2.****Signature with Date**

ACKNOWLEDGEMENT

The mini project report on “**College Placement Management System**” is the outcome of guidance, moral support and devotion bestowed on me throughout my work. For this I acknowledge and express my profound sense of gratitude and thanks to everybody who have been a source of inspiration during the project work.

First and foremost, I offer my sincere phrases of thanks with innate humility to our **Principal Dr. B. S. Anami** who has been a constant source of support and encouragement. I would like to thank our **Dean Academics Dr. S. G. Joshi** for his constant support and guidance. I feel deeply indebted to our **H.O.D. Dr. Yerriswamy T.** for the right help provided from the time of inception till date. I would take this opportunity to acknowledge our **Guide Prof. Shridhar Allagi**, who not only stood by us as a source of inspiration, but also dedicated his/her time for me to enable me present the project on time. I would be failing in endeavor, if I do not thank our **Coordinator Prof. Pradeep Surasura** who has helped me in every aspect of my mini project work. Last but not the least, I would like to thank my parents, friends & well-wishers who have helped me in this work.

Rashmi Ramakantha Pai

CONTENTS	Page No
1. Introduction	
1.1 Background Study.	6
1.2 Problem Statement.	6
1.3 Aim & Objectives.	6
1.4 Project Overview.	7
1.5 Project Scope.	7
1.6 Literature Overview.	7
1.7 Advantages.	9
1.8 Disadvantages.	9
2. Requirement and System Analysis	
2.1 System Requirement Specification.	
2.1.1 General Description.	10
2.1.2 System Objectives.	10
2.1.3 System Requirements.	
2.1.3.1 Non-Functional Requirements.	10
2.1.3.2 Functional Requirements.	11
2.2 Software Specification.	11
3. Normalization	13
3.1 Functional Dependencies.	15
4. Conceptual representation of application	
4.1 Entity Relationship Diagram.	17
4.2 Schema Diagram.	19
5. Design and Implementation	
5.1 Design.	20
5.2 Implementation.	27
5.2.1 Front End.	27
5.2.2 Back End.	30
6. Testing and Validation	
6.1 Use case Diagram.	37
6.2 Validation.	39
7. Conclusion and future scope	40

ABSTRACT

The “College Placement Management System” is an online application that can be accessed throughout the organization and outside as well with proper login provided. It offers many flexible and convenient features, allowing admin, students and company to maximize time and efficiency. It has been developed to override the problems prevailing in the practicing manual system.

The application is reduced as much as possible to avoid errors while entering the data. It also provides error messages while entering invalid data. No formal knowledge is needed for the user to use this system. Thus, by this all it proves it is user-friendly.

This system allows to update the information for students and companies. Also, the admin manages both students and companies in the way such that the admin can access and update their information.

CHAPTER 1

INTRODUCTION

In this chapter, we will see an overview about aim and objectives, background study, overview and scope of the project.

1.1 Background Study

The use of the Internet and the World Wide Web has revolutionized the provision of information and the facility for the user to take action on the information obtained. Use of the Internet to enable students and companies to manage the placement process with the active involvement of the Placement Coordinator. This led to a unique web-based training and placement management system. College placement management system provides information on placement providers and the placements they offer so that students may view and assess their opportunities.

1.2 Problem Statement

This system is being developed to override the problems prevailing in the manual system. The documents of the students for training and placements are maintained manually. TP organizer has to collect information about companies and their eligibility criteria and notify students time to time. If any modification or updating of details is required then it has to be searched and done manually.

1.3 Aim & Objectives

This project is aimed at developing a web application for the Placement Department of the College that can be accessed throughout the organization with proper login provided. This system can be used as a web application for the Training and Placement Officers (TPO) of the college to manage the student information with regard to placement.

The main objective of the project on “College Placement Management System” is to manage the details of students, company, training, placement, registration. It manages all the information about training, placement, students and companies. The project is totally built at administrative end and thus only admin can access all the information. However, students and companies can also access and update their information.

The purpose of the project is to build an application program to reduce the manual work for placement and training cell of college, for students and for companies. It maintains records instead of written materials.

1.4 Project Overview

College Placement Management System by overall aims to facilitate the applicant to apply for the job or internship. Indirectly, it is also to facilitate the managerial department of an organization for an optimized and systematic training and placement process.

The system has web-forms like registration forms, login forms and account pages. It includes the following main modules:

- Admin Module.
- Company Module.
- Student Module.

An administrator can be a manager who has full authority over the whole system. The administrator is able to retrieve data of the registered students, placed students and approve the companies. The students can register themselves for placement or internship of the approved Companies.

1.5 Project Scope

College Placement Management System enables the users to have the typical recruitment facilities and features at their disposal. It resolves typical issues of manual staffing processes and activities into a controlled and closely monitored work flow in the architecture of the application. This multi-platform solution brings in by default, the basic intelligence and immense possibilities for further extension of the application as required by the user. The system makes it simpler to share and manage the organization's human resource requirements with higher efficiency and easiness. The objective of these websites is to serve as a common meeting ground for jobseekers and organization, both locally and globally. This kind of systems is specifically designed for organization to help in solving staffing problems and managing human resource department activities at high degree of optimization.

1.6 Literature Overview

Each and every process in the existing system is carried out manually. The college training and placement officer had to refer all the records of previous years for even minor details. This used to be tedious and more time consuming than it sounds. It becomes more difficult when the number of students increase each passing year. There are other limitations of existing system. In manual placement management system, all the task is done by human interventions. Therefore, there is maximum chance of errors. The files are not stored in hierarchical form. Thus, searching for a particular becomes complex task. Updating certain information is difficult and ambiguous which may lead to data redundancy due to the chances of duplication of information. Not every student is aware of the placement updates by training and placement officer of the college, therefore they may lose an opportunity to grab a seat for job interview.

Related works:

Title: Design Paper on Online Training and Placement System (OTaP),
Author: Mr. Nilesh T. Rathod, Prof. Seema Shah,
Department of CSE

Year:2013

In this system they provide an online training and placement process. The admin of the system can see the user information and will validate it. The admin also generates the student list based on the company criteria and company details are also provided to the all users. searching and sorting can be done, and reports can be generated. Alumni data can be maintained in our system. Overall, all the process of the training and placement department is done automatically. We minimize the human intervention. But the drawback of proposed system is here all is done online. No message sending is provided. And also maintaining is little bit complicated.

Title: COLLEGE PLACEMENT MANAGEMENT SYSTEM.

Author: Maryam Sayyed, Faiza Umatiya, Seemab Zehera, Prof. Shiburaj Pappu, RCOE, Rizvi College of Engineering, Mumbai.

Year: 2020

Title: Web Based Placement Management System,

Author: Anjali.V, Jeyalakshmi.PR, Anubala.R, Sri Mathura devi.G, Ranjini.V

Year: 2016

In this paper, have the development of the system is improved facilities. The system can overcome all the limitation of the existing system, such as student's information is maintained in the database, it also gives more security to data, ensures data accuracy as well as reduces paper work and save time, only eligible students get chance, it makes information flow efficient and paves way for easy report generation, reduce the space and system is cost effective.

Title: Application for Training and Placement Cell

Author: K.Anand, Retheesh, J. Hemalatha, S. Karishma, R. Logeswari

Year: 2015

Title: College Collaboration Portal with Training and Placement

Authors: Shilpa Hadkar, Snehal Baing, Trupti Harer, Sonam Wankhade, K. T. V. Reddy, and I. T. Department Padmabhushan Vasantdada Patil Pratishthan's College of Engineering. Sion (East), Mumbai.

The project targets documentation of student's placement activities and automation of documentation. But lacks Event Management System's features along with report generation.

Title: Campus Recruitment Management: Platform based on dynamic electronic commerce

Authors: Diksha Varshney, Bhumika Sharma, Somya Jain

Year: 2014

In this paper, the electronic recruitment systems are used to facilitate and improve human resource management. They address the needs of employers and job-seekers via internetworking means which increase the speed of employment, and improve the quality of recruitment and services and They become vital assistance to human discrimination to put right people in right places. due to the rapid change in jobs demands and the required specialization and experience, it becomes more and more difficult for recruiters to find employees that are right for their business state the units for each quantity that you use in an equation.

1.7 Advantages

- 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.

1.8 Disadvantages

This application is in the local host, it cannot be run on public platform yet.

CHAPTER 2

REQUIREMENT AND SYSTEM ANALYSIS

In this chapter, we will discuss and analyze about developing process of College Placement Management System including software requirement specification. The functional and non-functional requirements are included in requirement part to provide complete description and overview of system requirement before the developing process is carried out.

2.1 System Requirement Specification

2.1.1 General Description

The primary purpose to develop this system is to optimize the recruitment process for an organization. Besides, the qualified applicants could be sort by this system based on their qualifications and company requirements.

2.1.2 System Objectives

- To provide the registering and login facilities for companies and students.
- To manage student details and company approval.
- Track all the information of placement, internships, students etc.
- Editing, adding and updating of records is included which results in proper resource management of training data.

2.1.3 System Requirements

2.1.3.1 Non-Functional Requirements

Usability

- The software must have a simple and user-friendly interface.
- The navigation to various pages should make it more convenient to the users so as to save time and confusion.

Performance Requirements

- The users must get the response within seconds
i.e., the response time of a particular function should be minimum.
- Completely separate business login at server side from the student interface ensures good performance.
- The system would exhibit high performance because it would be well optimized.

Backup

There should be an easy back-up feature for the entire data, to prevent losing any data.

Platform/Browser independence

The system should be able to work on any of the modern browsers like Firefox / Explorer / Opera / Chrome, and any of the common Operating Systems like Linux, Windows and Mac OS.

2.1.3.2 Functional Requirements

- For the students:
 - Student can register at training and placement cell.
 - Can login and view his/her profile.
 - Can update his/her information that were provided at the time of registration.
 - Student can view general notifications from T&P cell and can also see the details of companies that are visiting.
 - A student can apply for the visiting company for internship or placement if he/she is eligible for that.
- For the companies:
 - A company can register at training and placement cell.
 - Companies add basic details such as the eligibility criteria, max number of backlogs etc.
- For the admin:
 - Can add new admin at training and placement cell.
 - Add general notifications for the students.
 - Have access to all the information about students and companies i.e., admin can view or update information of students and company.
 - Only the admin gives approval to the company after verifying that the company details.
 - Admin can change the status of the students after receiving final result from the company.
 - Admin can view the list of students who got placement in a company and also the list of the students selected for placement.
- The password of every user is stored in database in encrypted form, i.e., the password is crypted to md5 hash code and then stored in the database. When a password is entered, it is checked whether the md5 hash of the entered password is equal to the stored md5 hash code.
- If any user is logged in then no other user can log in at the same time on the same browser.

2.2 Software Specification

Tools used:

- Frontend – HTML, CSS, JavaScript
- Backend – MYSQL, PHP
- Software – Xampp Server

For users:

- Internet connectivity.
- Web browser (Chrome, Mozilla Firefox).

For administrator:

- Apache server – to host.
- phpmyadmin root access.
- MYSQL knowledge.

Hardware Specification:

- Operating System- Windows
- Browser that supports HTML, CSS, JavaScript
- RAM: 2 GB RAM
- Hard Disk: 80 GB HDD
- Keyboard: Any Keyboard

CHAPTER 3

NORMALIZATION

Normalization is the process of organizing the columns (attributes) and tables (relations) of a relational database to reduce data redundancy and improve data integrity. Normalization is also the process of simplifying the design of a database so that it achieves the optimal structure composed of atomic elements.

Normalization rules are divided into following normal form.

- First Normal Form: A relation is said to be in 1NF, if all values in given relation are atomic.
- Second Normal Form: A relation is said to be in 2NF, if it satisfies 1NF and the non-prime attributes should be fully functionally dependent on key of relation.
- Third Normal Form: A relation is said to be in 3NF, if it satisfies 2NF and non-prime attribute is transitively dependent on primary key.
- Boyce Codd Normal Form: A relation is said to be in BCNF, if and only if there are no non-trivial functional dependencies of attributes on anything or other than a super key or candidate key.

There is a main authenticated admin who manages the system. Admin can access the system through ID, username and password. To gain the access, approve companies, change status of students etc, admin should register into the system.

- admin (**ADMIN_ID**, ADMIN_NAME, A_PASSWORD, POST, EMAIL, CONTACT_NO, DOB, QUALIFICATION)
 - ⇒ Primary key is ADMIN_ID and no other functional dependency is present.
Therefore, all values are atomic.
Relation is in 1NF.
 - ⇒ Now, keys = {ADMIN_ID, (ADMIN_NAME, DOB), ...}
Therefore, there is no partial functional dependency.
Relation is in 2NF.
 - ⇒ Therefore, there is no transitive dependency.
Relation is in 3NF.
- student (**STUDENT_ID**, S_PASSWORD, STUDENT_NAME, FATHER NAME, MOTHER NAME, GENDER, DOB, EMAIL, ADDRESS, CONTACT_NO, BRANCH, TENTH_PER, TENTH_PASS_YEAR, TWELTH_PER, TWELTH_PASS_YEAR, CGPA, PASSING_YEAR, BACKLOGS, APPLY_FOR, STATUS, APPLY_COUNT, ABSENT)
 - ⇒ Primary key is STUDENT_ID and no other functional dependency is there.
Therefore, all values are atomic.
Relation is in 1NF.

⇒ Now, keys = {STUDENT_ID, (STUDENT_NAME, FATHER NAME), (STUDENT_ID, DOB), ...}

Therefore, there is no partial functional dependency.

Relation is in 2NF.

⇒ Therefore, there is no transitive dependency.

Relation is in 3NF.

- company (**COMPANY_ID**, COMPANY_NAME, C_PASSWORD, WEBSITE, ADDRESS, STATUS, COMING_DATE, APPROVAL)

COMPANY_NAME \longrightarrow WEBSITE, ADDRESS

⇒ Primary key is COMPANY_ID.

Therefore, all values are atomic.

Relation is in 1NF.

⇒ Now, keys = {COMPANY_ID, (COMPANY_NAME, WEBSITE), ...}

Therefore, there is no partial functional dependency.

Relation is in 2NF.

⇒ Therefore, there is no transitive functional dependency.

Relation is in 3NF.

- company_branch (**COMPANY_NAME**, C_TYPE, **BRANCH**, MIN_CGPA, MAX_BACKLOGS, MAX_SALARY, MAX_STIPEND, JOB_PROFILE, PLACE_OF_POSTING)

⇒ Primary key is COMPANY_NAME and no other functional dependency is there.

Therefore, all values are atomic.

Relation is in 1NF.

⇒ Now, keys = {(COMPANY_NAME, C_TYPE, BRANCH)}

Therefore, there is no partial functional dependency.

Relation is in 2NF.

⇒ Therefore, there is transitive functional dependency.

Relation is in 3NF.

- student_placement (**STUDENT_ID**, COMPANY_ID, STUDENT_NAME, COMPANY_NAME, PACKAGE)

⇒ Primary key STUDENT_ID and no other functional dependency is there.

Therefore, all values are atomic.

Relation is in 1NF.

⇒ Now, keys = {STUDENT_ID, (STUDENT_NAME, COMPANY_ID), ...}

Therefore, there is no partial functional dependency.

Relation is in 2NF.

⇒ Therefore, there is transitive functional dependency.

Relation is in 3NF.

- student_internship (**STUDENT_ID**, COMPANY_ID, STUDENT_NAME, COMPANY_NAME, STIPEND)
 - ⇒ Primary key STUDENT_ID and no other functional dependency is there.
Therefore, all values are atomic.
Relation is in 1NF.
 - ⇒ Now, keys = {STUDENT_ID, (STUDENT_NAME, COMPANY_ID), ...}
Therefore, there is no partial functional dependency.
Relation is in 2NF.
 - ⇒ Therefore, there is transitive functional dependency.
Relation is in 3NF.

3.1 Functional Dependencies

Full dependency

In a relation the attribute(S) B is fully functionally dependent on A if B is functionally dependent on A but not on any proper subset of A.

Partial dependency

If there is some attribute that can be removed from A and dependency still holds.

Transitive dependency

In a relation if attribute(S) $A \rightarrow B$ and $B \rightarrow C$, then C is transitively dependent on A via B.

The function dependencies can be depicted as shown in the below table:

Relation	Functional Dependencies
Admin	$ADMIN_ID \rightarrow ADMIN_NAME, A_PASSWORD, POST, EMAIL, CONTACT_NO, DOB, QUALIFICATION$
Student	$STUDENT_ID \rightarrow S_PASSWORD, STUDENT_NAME, FATHER NAME, MOTHER NAME, GENDER, DOB, EMAIL, ADDRESS, CONTACT_NO, BRANCH, TENTH_PER, TENTH_PASS_YEAR, TWELTH_PER, TWELTH_PASS_YEAR, CGPA, PASSING_YEAR, BACKLOGS, APPLY_FOR, STATUS, APPLY_COUNT, ABSENT$
Company	$COMPANY_ID \rightarrow COMPANY_NAME, C_PASSWORD, WEBSITE, ADDRESS, STATUS, COMING_DATE, APPROVAL$

Company Branch	COMPANY_NAME → C_TYPE, BRANCH, MIN_CGPA, MAX_BACKLOGS, MAX_SALARY, MAX_STIPEND, JOB_PROFILE, PLACE_OF_POSTING
Student_placement	STUDENT_ID → COMPANY_ID, STUDENT_NAME, COMPANY_NAME, PACKAGE
Student_internship	STUDENT_ID → COMPANY_ID, STUDENT_NAME, COMPANY_NAME, STIPEND
Registered_placements	STUDENT_ID → STUDENT_NAME, COMPANY_NAME
Registered_interns	STUDENT_ID → STUDENT_NAME, COMPANY_NAME

CHAPTER 4

CONCEPTUAL REPRESENTATION OF APPLICATION

In this chapter, we will study detailed design used to build the college placement management system. The design chapter includes the schema diagram and the entity relationship diagram of the vehicle insurance management system that is required to design the database. A database schema diagram is the structure that represent the logical view of entire database. A database schema defines its entities and the relationship among them. An entity relationship diagram also called as ER diagram is a graphical representation of entities and their relationships to each other typically used in computing in regard to the organization of data within database on information systems. The schema diagram of vehicle insurance management system is normalized hence it helps to avoid redundancy as well as reduce anomalies.

4.1 Entity Relationship Diagram (ER Diagram)

An entity-relationship (ER) diagram is a specialized graphic that illustrates the interrelationships between entities in a database. ER diagrams often use symbols to represent three different types of information. Boxes are commonly used to represent entities. Diamonds are normally used to represent relationships and ovals are used to represent attributes.

An entity-relationship model (ERM) in software engineering is an abstract and conceptual representation of data. Entity-relationship modeling is a relational schema database modeling method, used to produce a type of conceptual schema or semantic data model of a system, often a relational database, and its requirements in a top-down fashion.

The ER diagram for college placement management system:

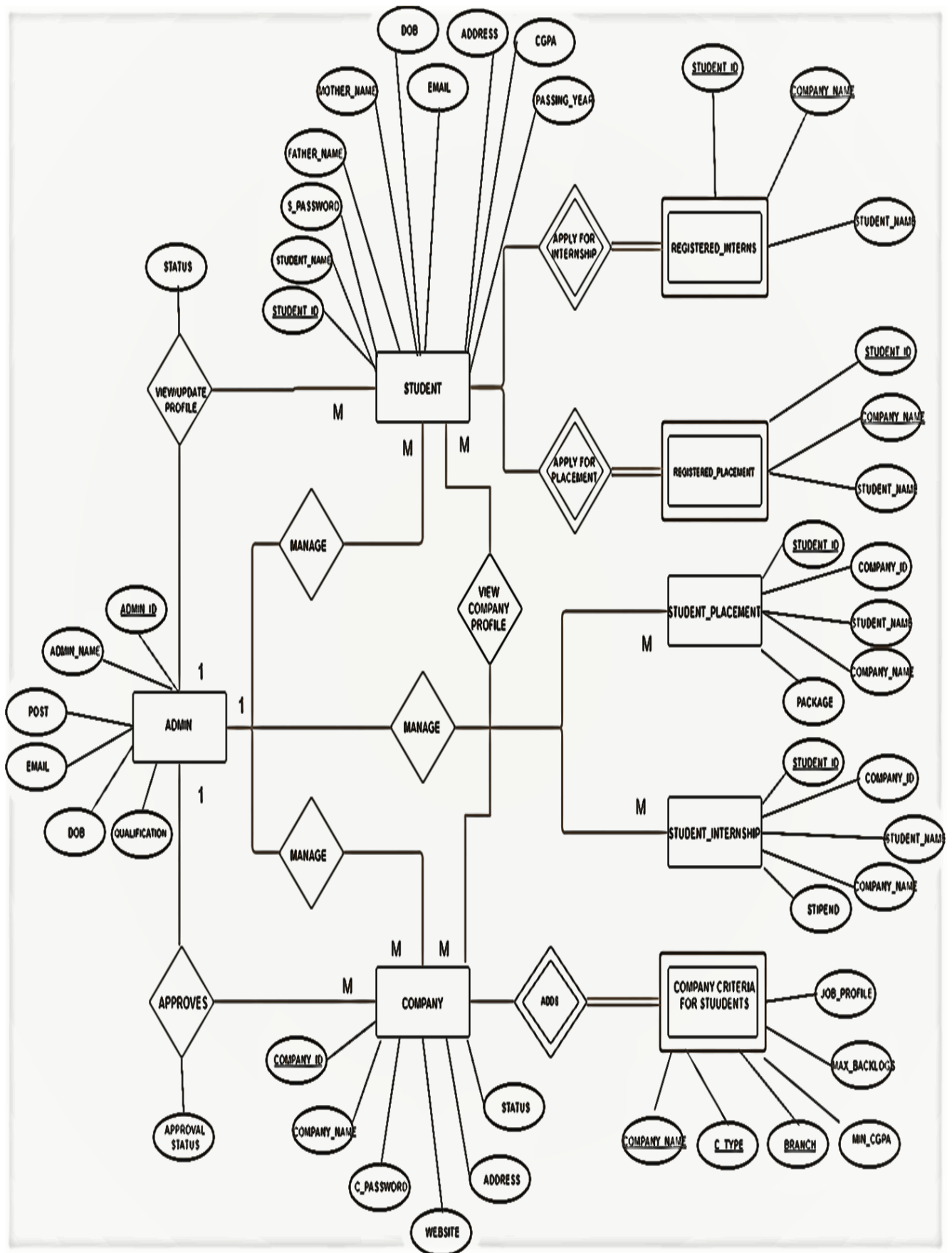


Fig 4.2: ER Diagram of College Placement Management System

4.2 Schema Diagram

A schema is the structure behind data organization. It is a visual representation of how different table relationships enable the schema's underlying mission business rules for which the database is created.

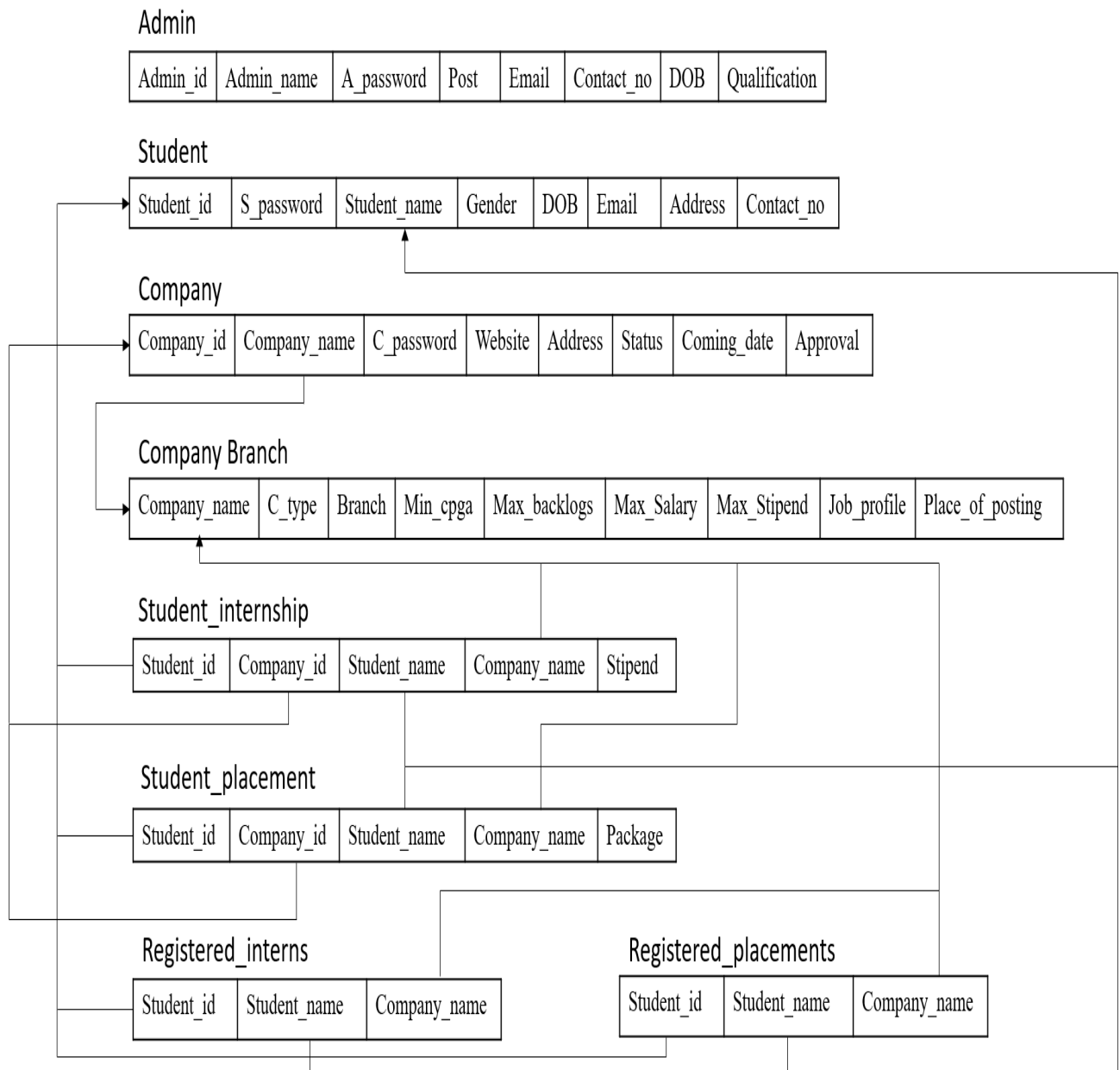


Fig 4.2: Schema Diagram of College Placement Management System

CHAPTER 5

DESIGN AND IMPLEMENTATION

5.1 Design

It contains a detailed description about each interface along with a screen shot of the interface.

Home Page:

The home page of the application is common to all the system users/administrators. The home page shows the following three categories:

- a) Company.
- b) Student.
- c) Admin.

The below links allow the user to view the facilities provided by the college, about the college and contact and address information.

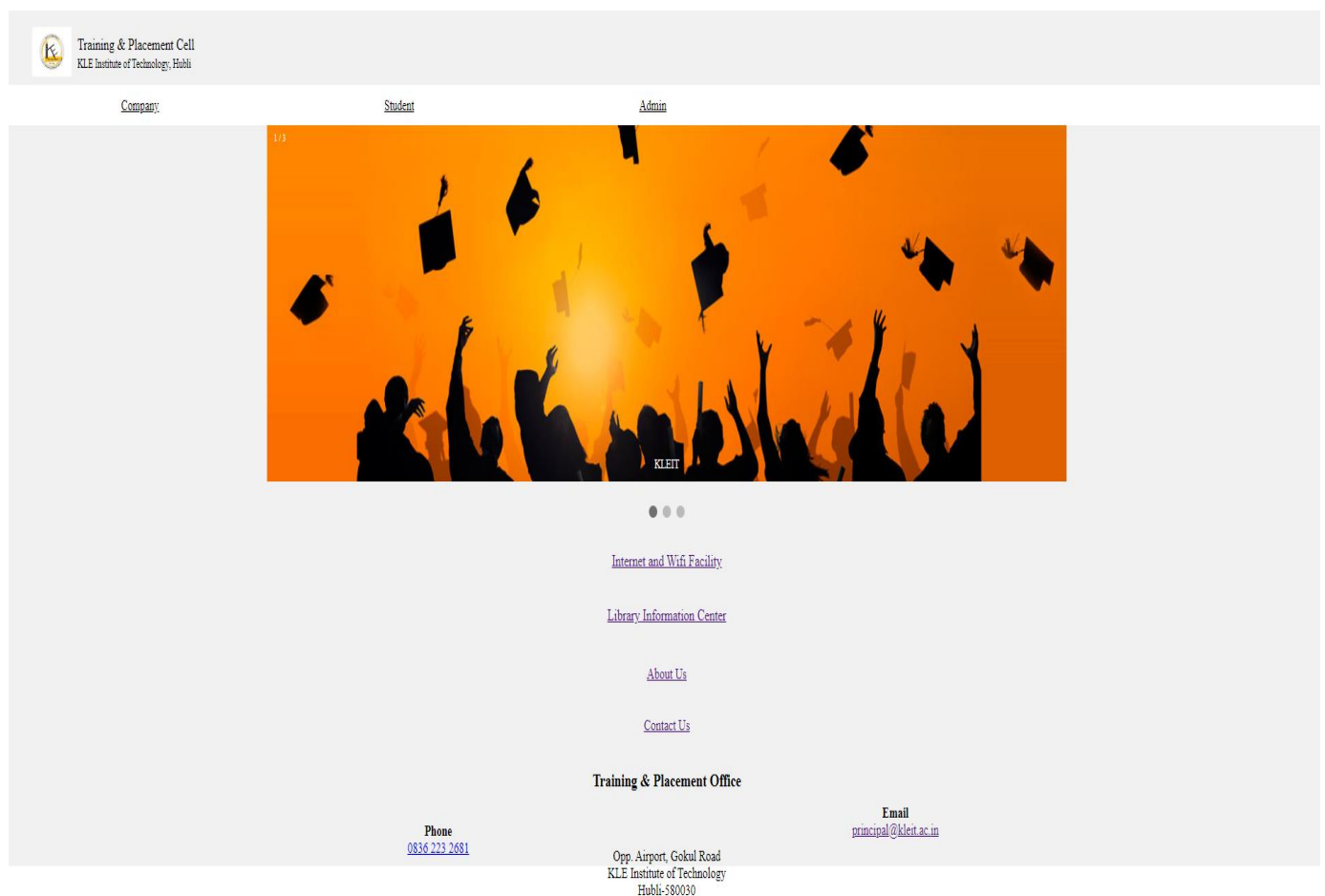


Fig 5.1.1: Home Page

a) Company Category:

The company category has two options-

- ⇒ Company Register.
- ⇒ Company Login.

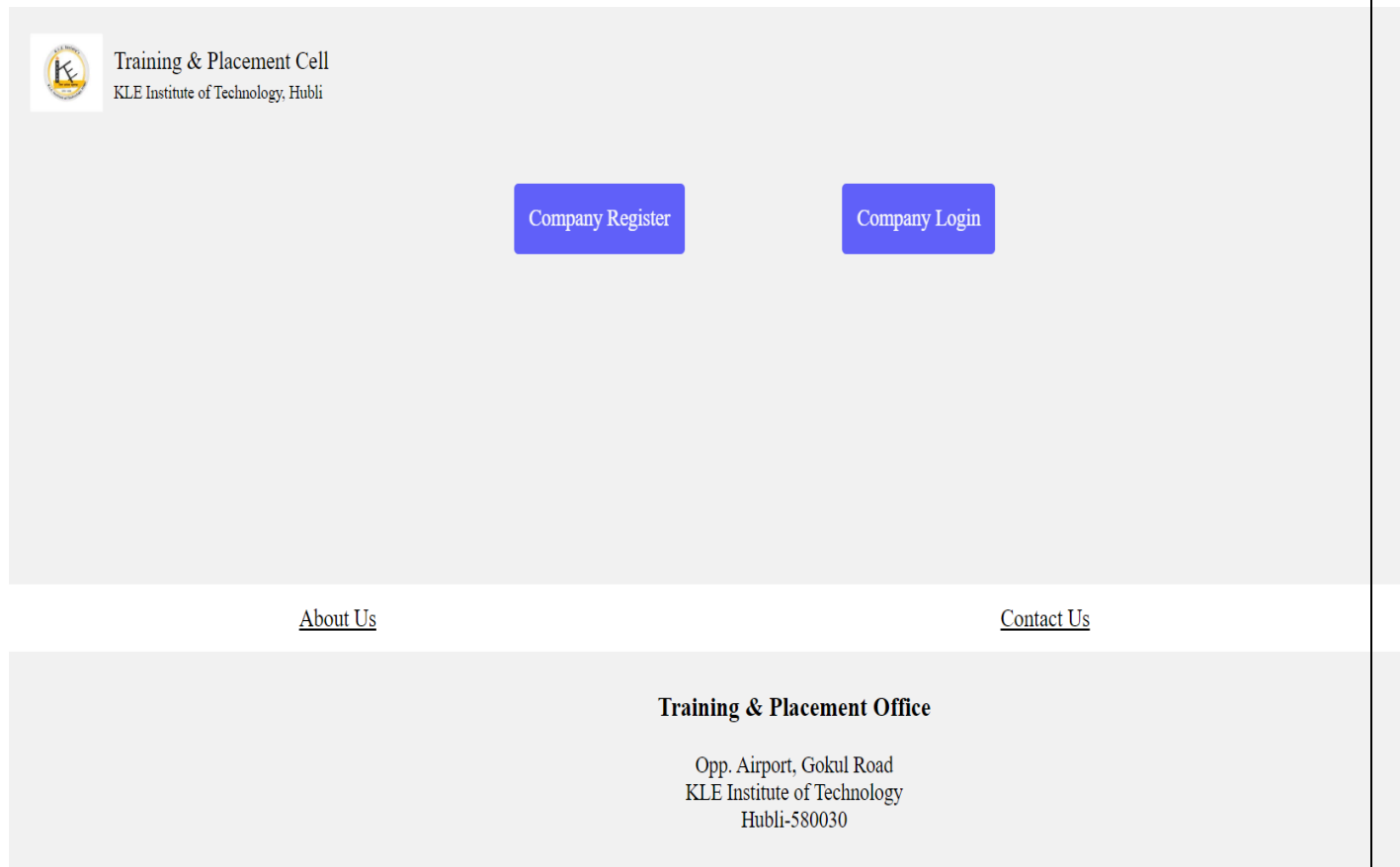


Fig 5.1.2: Company Home Page

b) Student Category:

The student category has three options-

- ⇒ Student Register.
- ⇒ Student login-placement.
- ⇒ Student login-internship.

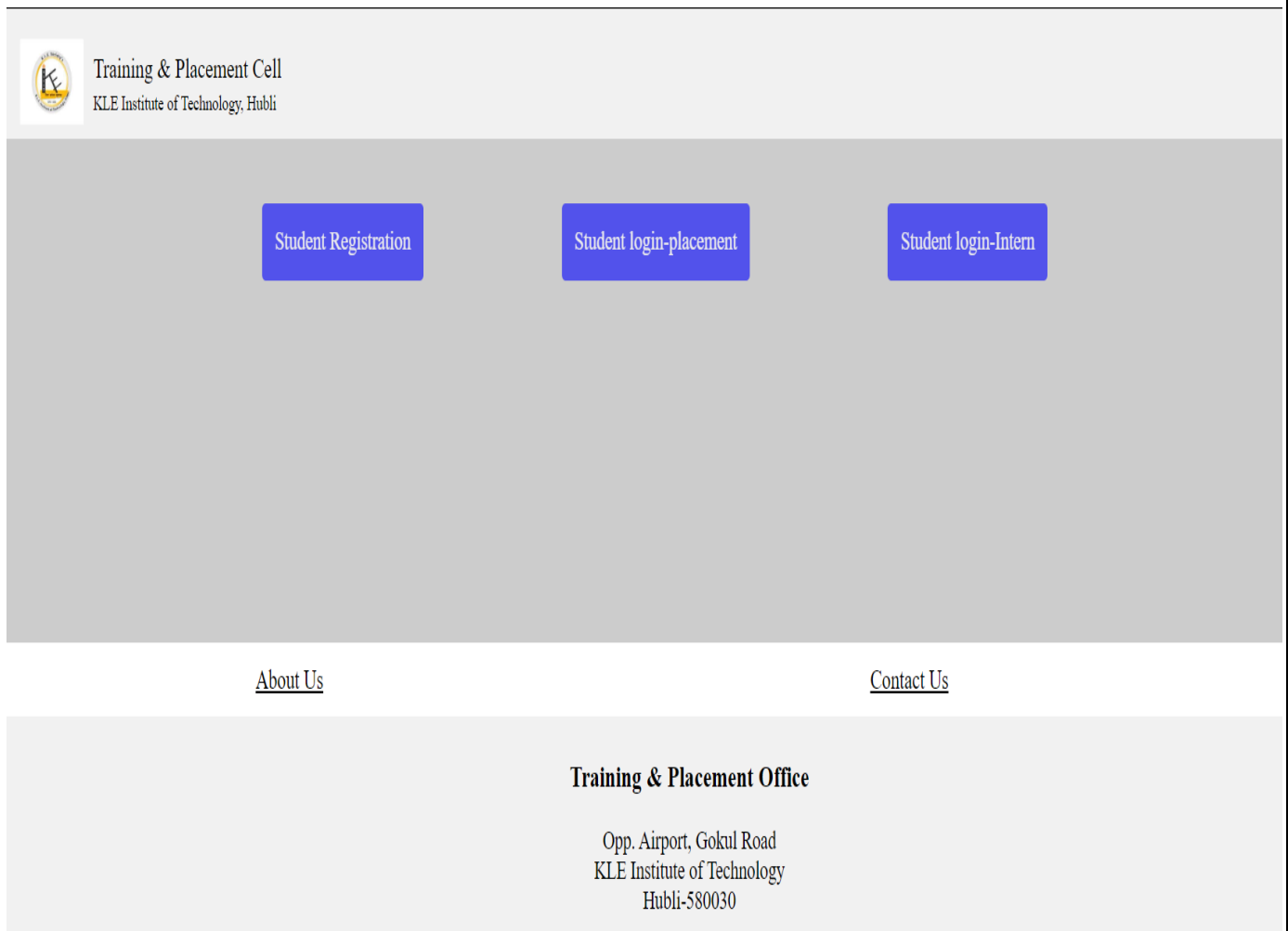


Fig 5.1.3: Student Home Page

Student Login:

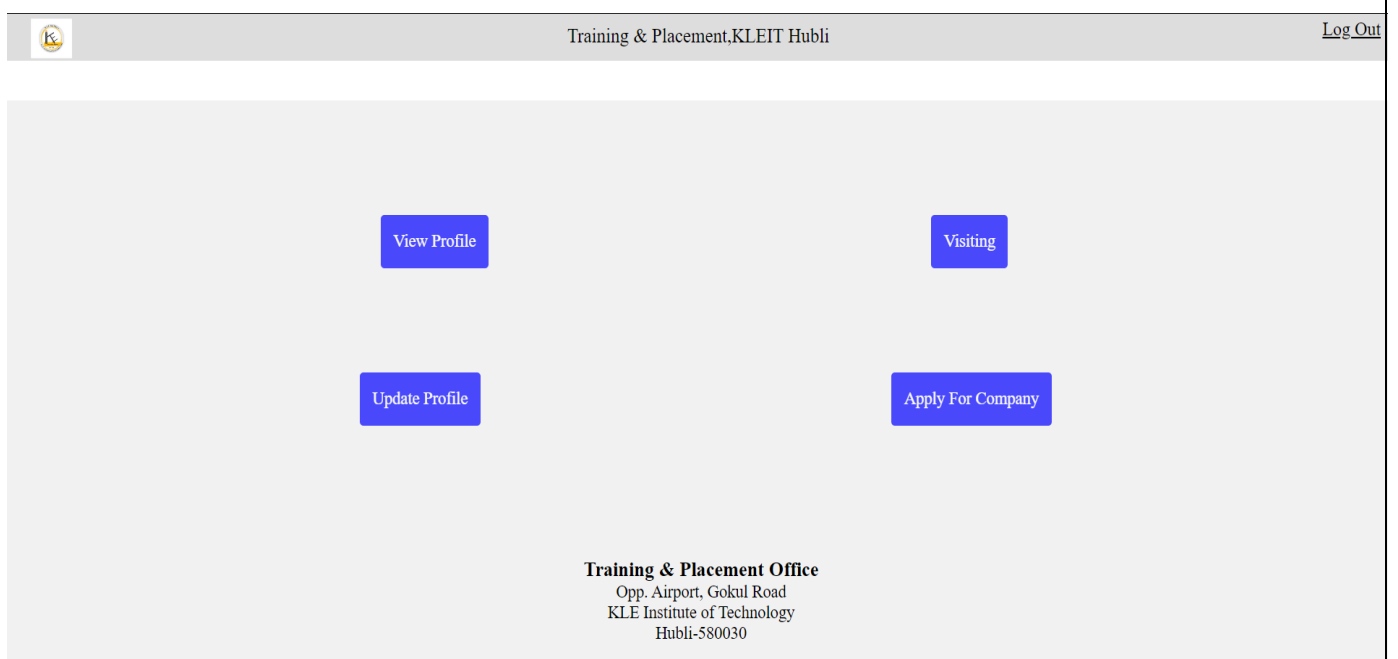




Fig 5.1.4: After Student login

View Profile:

 Training & Placement, KLEIT Hubli

Student Profile



Student Name : renuka

Student Id : 108

Status : -

D.O.B : 2000-03-03

Gender : Female

Email : rashmipai2000@gmail.com

Address: Hubli

Conatct Number : 8618015254

Branch: CSE

10th % : 96

10th pass year : 2013

12% : 88

12th pass year : 2015

CGPA : 8.7

Final Year : 2019

Backlogs : 0

Apply For : Internship

Fig 5.1.5: Student Profile Page

Update Profile:
Also includes, change of password.



Update Profile

Student name

Date of birth

Email

Address

Contact Number

10th percentage

12th percentage

CGPA

Current Password

New Password

Updated the details [Go Back](#)

Fig 5.1.6: Update Profile Page

Apply for Company:



Apply For Intern


Successfully applied for the company

Company For Intern

Applied for the company? [Go Back](#)

Fig 5.1.7: Apply for Company Page

Companies for internships and placements can be viewed as shown below,



Training & Placement, KLEIT Hubli

Log Out

Companies-Internship

Company Name	Visiting Date	Branch	Min CGPA	Max Backlogs	Max Stipend	Job Profile	Place Of Posting	Company Website
Microsoft	2021-02-20	CSE	6.5	0	10000	Program Manager	Bangalore	https://careers.microsoft.com/us/en
Infosys	2021-02-23	CSE	6.5	1	5000	System Testing	Mysore	https://www.infosys.com/careers/apply.html
IBM	2021-02-25	CSE	6.5	1	5000	Technical Specialist	Hubli	https://www.ibm.com/in-en/employment/

Fig 5.1.8: Companies for internship page

c) Admin Category:

The admin category shows two option-

- ⇒ Admin Register (But only an admin can add other admins).
- ⇒ Admin Login.


	Training & Placement Cell KLE Institute of Technology, Hubli
<div>Admin Register</div> <div>Admin Login</div>	
About Us	Contact Us
Training & Placement Office Opp. Airport, Gokul Road KLE Institute of Technology Hubli-580030	

Fig 5.1.9: Admin Home Page

In the Admin Login category, we have the following options-

- ⇒ Company Approval.
- ⇒ Change student status.
- ⇒ Add Admin.
- ⇒ Data.

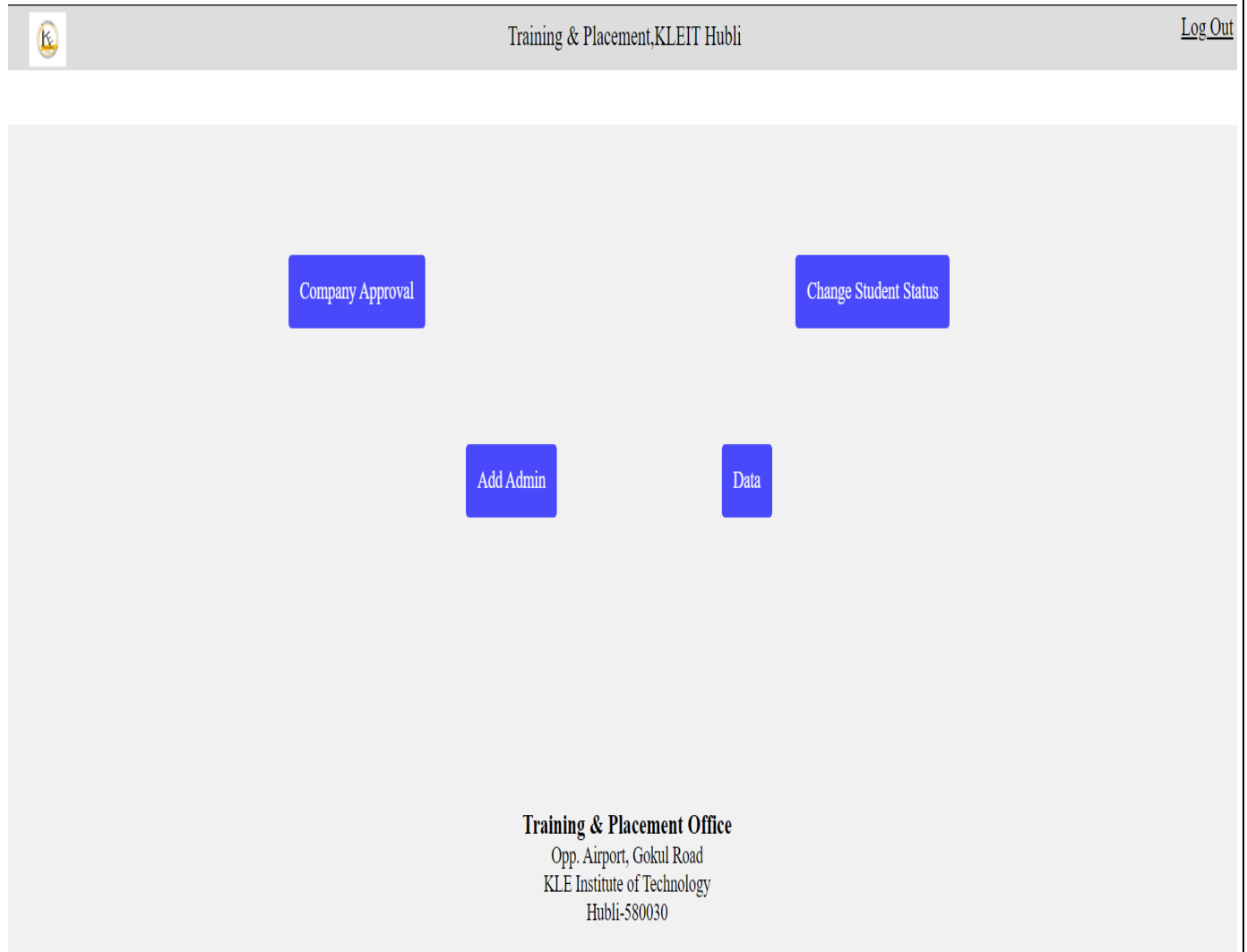


Fig 5.1.10: Admin Login Page

The Data category is again having four options which shows data related to that particular module-

- ⇒ Registered students for internship.
- ⇒ Registered students for placement.
- ⇒ Selected students for internship.
- ⇒ Selected students for placement.

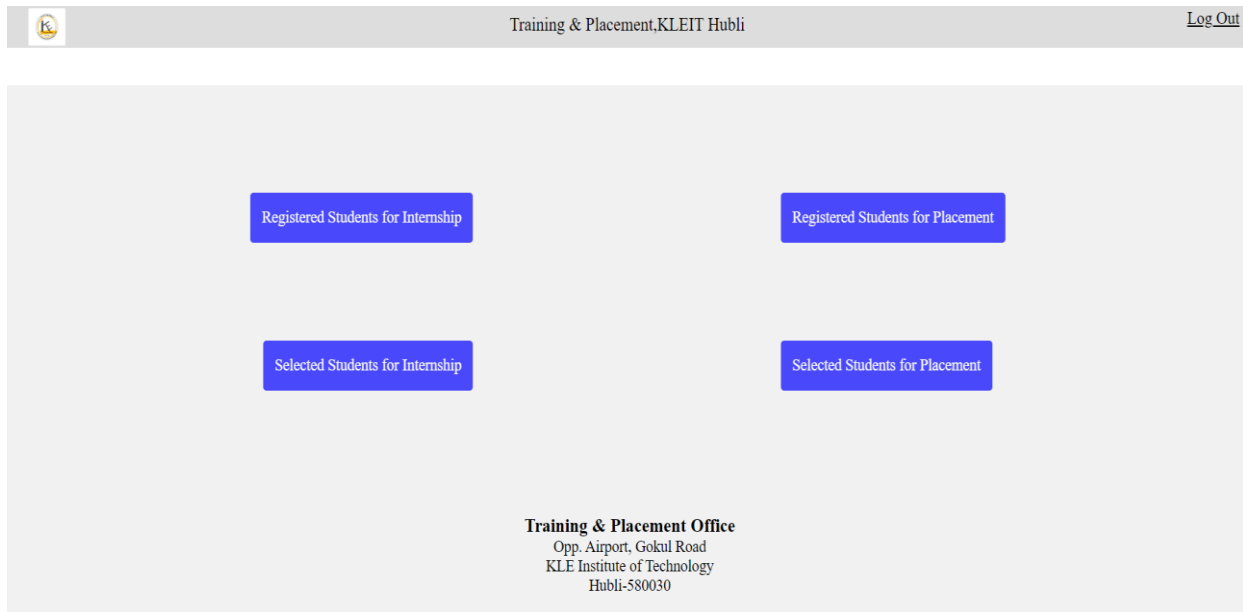


Fig 5.1.11: Data category in Admin Login

5.2 Implementation

5.2.1 Front End Session Protection

```
<?php
//This is for Protection
//The user with active session will be redirected to its specific index page
session_start();
if(isset($_SESSION['user']))
{
    if($_SESSION['user']=='admin')
    {
        header("Location: index_admin.php");
    }
    if($_SESSION['user']=='student_int' )
    {
        header("Location: index_student_intern.php");
    }
    if($_SESSION['user']=='student_place' )
    {
        header("Location: index_student_placement.php");
    }
    if($_SESSION['user']=='company')
    {
        header("Location: index_company.php");
    }
}
?>
```

To check in a database whether a user already exists.

```
// first check the database to make sure
// a user does not already exist with the same username and/or email
$user_check_query = "SELECT * FROM student WHERE STUDENT_ID='$vari' LIMIT 1";
$result = mysqli_query($db, $user_check_query);
```

```
if ($user)
{ // if user exists
  if ($user['STUDENT_NAME'] === $student_name)
    array_push($errors, "Student already registered");
  if ($user['STUDENT_ID'] === $student_id)
    array_push($errors, "Id already exists");
  if ($user['EMAIL'] === $st_email)
    array_push($errors, "email already exists");
}
```

Various Constraints check

```
if (count($errors) == 0)
{
  $password = md5($st_password1);//encrypt the password before saving in the database
  if(isset($_FILES['image']['tmp_name']))
  {
    $file = addslashes(file_get_contents($_FILES['image']['tmp_name']));
  }
  else
  {
    //$file = null;
    //echo "not set";
  }
}
```

Password encryption

```
$password = md5($c_password);
```

Displaying error and success message

```
if (mysqli_num_rows($results) == 1) //Return the number of rows in a result set:
{
    $_SESSION['admin_id'] = $admin_id;
    $_SESSION['admin_name'] = $admin_name;
    $_SESSION['success'] = "Admin logged in";
    $_SESSION['user'] = "admin";//session variable used for protection
    header('location: index_admin.php');
}
else
{
    array_push($errors, "Wrong username/password combination");
}
```

Code for error

```
<?php if (count($errors) > 0) : ?>
<div class="error">
    <?php foreach ($errors as $error) : ?>
        <p><?php echo $error ?></p>
    <?php endforeach ?>
</div>
<?php endif ?>
```

Insert, delete, update queries

```
$query = "INSERT INTO admin
(ADMIN_ID,ADMIN_NAME,A_PASSWORD,POST,EMAIL,CONTACT_NO,DOB,QUALIFICATION)
VALUES('$admin_id', '$admin_name', '$password', '$post', '$a_email', '$con_number', '$dob2',
'$qualification')";
mysqli_query($db, $query);
header('location: admin_login.php');
```

```
$query = "UPDATE STUDENT set CGPA='$cgpa' where STUDENT_ID='$vari'";
mysqli_query($db, $query);
$query = "UPDATE STUDENT set S_PASSWORD='$password' where STUDENT_ID='$vari'";
mysqli_query($db, $query);
```

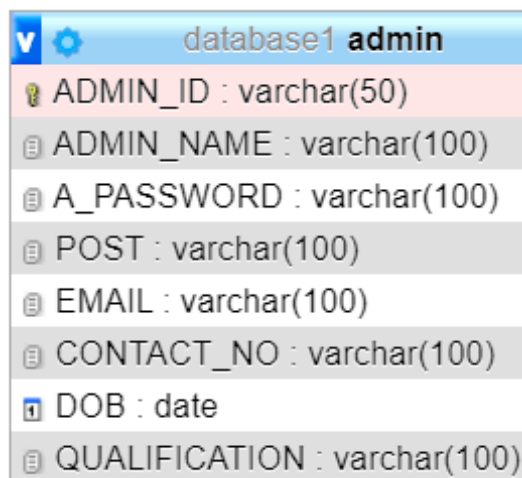
```
if (count($errors) == 0)
{
    $query = "INSERT INTO registered_interns
(STUDENT_ID,STUDENT_NAME,COMPANY_NAME)VALUES('$vari', '$student_name',
'$company_name')";
    mysqli_query($db, $query);
    $query = "UPDATE student set APPLY_COUNT=APPLY_COUNT+1 where
STUDENT_ID='$vari'";
    mysqli_query($db, $query);
}
```

5.2 Back end

Creation of tables

Admin table:

```
CREATE TABLE IF NOT EXISTS admin
(
    ADMIN_ID varchar(50) NOT NULL,
    ADMIN_NAME varchar(100) NOT NULL,
    A_PASSWORD varchar(100) NOT NULL,
    POST varchar(100) NOT NULL,
    EMAIL varchar(100) NOT NULL,
    CONTACT_NO varchar(100) NOT NULL,
    DOB date NOT NULL,
    QUALIFICATION varchar(100) NOT NULL,
    PRIMARY KEY (ADMIN_ID)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```



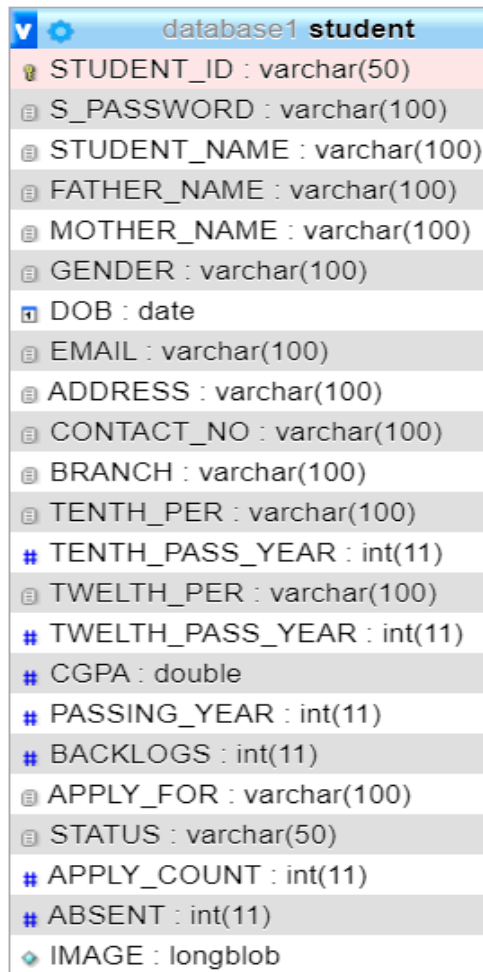
The screenshot shows a database table named 'admin' in a database named 'database1'. The table has the following columns and data types:

Column Name	Data Type
ADMIN_ID	varchar(50)
ADMIN_NAME	varchar(100)
A_PASSWORD	varchar(100)
POST	varchar(100)
EMAIL	varchar(100)
CONTACT_NO	varchar(100)
DOB	date
QUALIFICATION	varchar(100)

Fig 5.2.1: Admin Table

Student table:

```
CREATE TABLE IF NOT EXISTS student
(
    STUDENT_ID varchar(50) NOT NULL,
    S_PASSWORD varchar(100) NOT NULL,
    STUDENT_NAME varchar(100) NOT NULL,
    FATHER_NAME varchar(100) NOT NULL,
    MOTHER_NAME varchar(100) NOT NULL,
    GENDER varchar(100) NOT NULL,
    DOB date NOT NULL,
    EMAIL varchar(100) NOT NULL,
    ADDRESS varchar(100) NOT NULL,
    CONTACT_NO varchar(100) NOT NULL,
    BRANCH varchar(100) NOT NULL,
    TENTH_PER varchar(100) NOT NULL,
    TENTH_PASS_YEAR int NOT NULL,
    TWELTH_PER varchar(100) NOT NULL,
    TWELTH_PASS_YEAR int NOT NULL,
    CGPA double NOT NULL,
    PASSING_YEAR int NOT NULL,
    BACKLOGS int(11) NOT NULL,
    APPLY_FOR varchar(100) NOT NULL,    /*internship/placement*/
    STATUS varchar(50) DEFAULT "NS",    /*NS means Not Selected*/
    APPLY_COUNT int DEFAULT 0,
    ABSENT int DEFAULT 0,
    IMAGE longblob NOT NULL,
    PRIMARY KEY (STUDENT_ID)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```



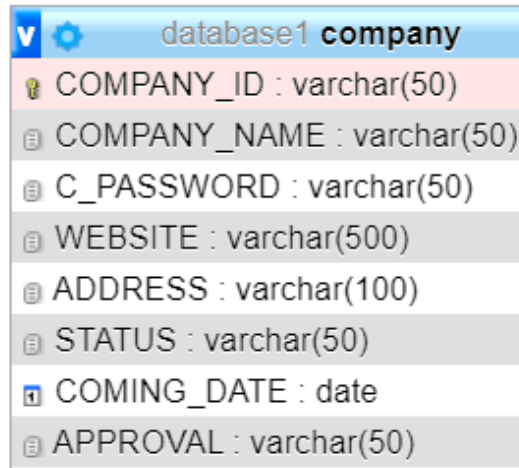
The screenshot shows a database table named 'student' in a database named 'database1'. The table has the following columns and data types:

Column Name	Data Type
STUDENT_ID	varchar(50)
S_PASSWORD	varchar(100)
STUDENT_NAME	varchar(100)
FATHER_NAME	varchar(100)
MOTHER_NAME	varchar(100)
GENDER	varchar(100)
DOB	date
EMAIL	varchar(100)
ADDRESS	varchar(100)
CONTACT_NO	varchar(100)
BRANCH	varchar(100)
TENTH_PER	varchar(100)
TENTH_PASS_YEAR	int(11)
TWELTH_PER	varchar(100)
TWELTH_PASS_YEAR	int(11)
CGPA	double
PASSING_YEAR	int(11)
BACKLOGS	int(11)
APPLY_FOR	varchar(100)
STATUS	varchar(50)
APPLY_COUNT	int(11)
ABSENT	int(11)
IMAGE	longblob

Fig 5.2.2: Student Table

Company table:

```
CREATE TABLE IF NOT EXISTS company
(
    COMPANY_ID varchar(50) NOT NULL,
    COMPANY_NAME varchar(50) NOT NULL,
    C_PASSWORD varchar(50) NOT NULL,
    -- C_TYPE varchar(50) NOT NULL, /*internship/placement*/
    WEBSITE varchar(500) NOT NULL,
    ADDRESS varchar(100) NOT NULL,
    STATUS varchar(50) DEFAULT "visiting", /*visited/visiting*/
    COMING_DATE date NOT NULL,
    /*INTERVIEW_TIME varchar(100) NOT NULL,*/
    APPROVAL varchar(50) DEFAULT "not approved",
    PRIMARY KEY (COMPANY_ID)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

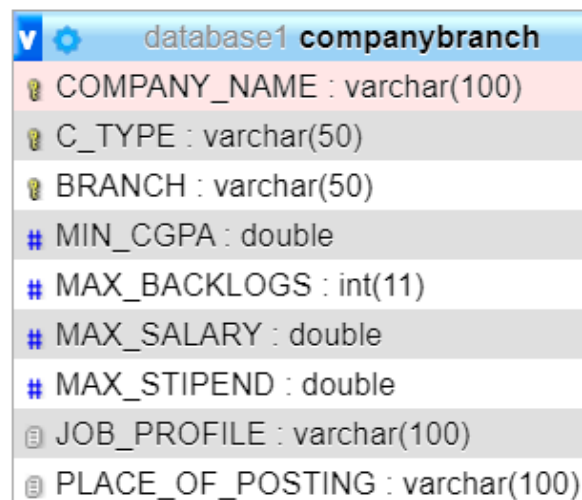
database1 company	
COMPANY_ID	varchar(50)
COMPANY_NAME	varchar(50)
C_PASSWORD	varchar(50)
WEBSITE	varchar(500)
ADDRESS	varchar(100)
STATUS	varchar(50)
COMING_DATE	date
APPROVAL	varchar(50)

Fig 5.2.3: Company Table

Company branch table:

CREATE TABLE IF NOT EXISTS companybranch

```
(
    COMPANY_NAME varchar(100) NOT NULL,
    C_TYPE varchar(50) NOT NULL,          /*internship/placement*/
    BRANCH varchar(50),
    MIN_CGPA double ,
    MAX_BACKLOGS int DEFAULT 0,
    MAX_SALARY double,
    MAX_STIPEND double,
    JOB_PROFILE varchar(100),
    PLACE_OF_POSTING varchar(100),
    PRIMARY KEY (COMPANY_NAME,C_TYPE,BRANCH)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

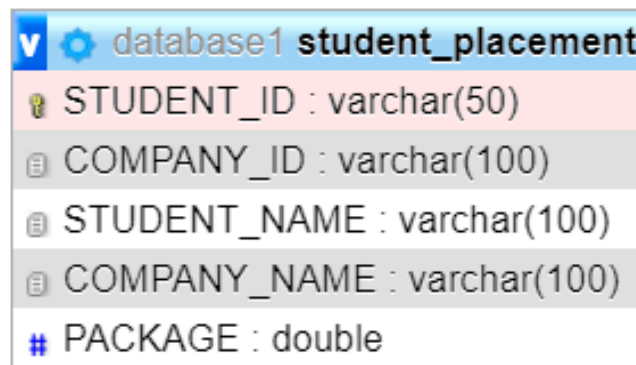


database1 companybranch	
COMPANY_NAME	varchar(100)
C_TYPE	varchar(50)
BRANCH	varchar(50)
MIN_CGPA	double
MAX_BACKLOGS	int(11)
MAX_SALARY	double
MAX_STIPEND	double
JOB_PROFILE	varchar(100)
PLACE_OF_POSTING	varchar(100)

Fig 5.2.4: Company branch Table

Student placement table:

```
CREATE TABLE IF NOT EXISTS student_placement
(
    STUDENT_ID varchar(50) NOT NULL,
    COMPANY_ID varchar(100) NOT NULL,
    STUDENT_NAME varchar(100) NOT NULL,
    COMPANY_NAME varchar(100) NOT NULL,
    PACKAGE double NOT NULL,
    /*PLACEMENT_DATE date NOT NULL, */
    PRIMARY KEY (STUDENT_ID)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```



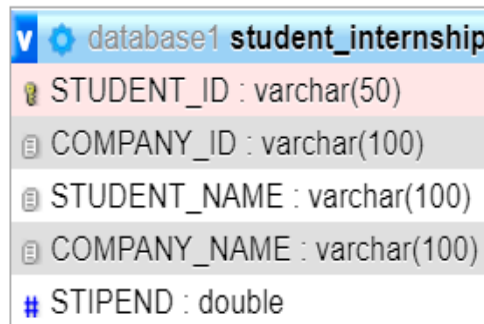
The screenshot shows a database interface with a table named 'student_placement' in 'database1'. The table structure is as follows:

Field	Type
STUDENT_ID	varchar(50)
COMPANY_ID	varchar(100)
STUDENT_NAME	varchar(100)
COMPANY_NAME	varchar(100)
PACKAGE	double

Fig 5.2.5: Student Placement Table

Student internship table:

```
CREATE TABLE IF NOT EXISTS student_internship
(
    STUDENT_ID varchar(50) NOT NULL,
    COMPANY_ID varchar(100) NOT NULL,
    STUDENT_NAME varchar(100) NOT NULL,
    COMPANY_NAME varchar(100) NOT NULL,
    STIPEND double NOT NULL,
    /*TRAINING_DURATION int NOT NULL, IN months*/
    PRIMARY KEY (STUDENT_ID)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

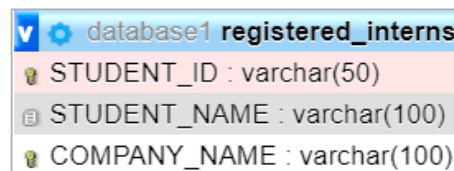


database1 student_internship	
STUDENT_ID	varchar(50)
COMPANY_ID	varchar(100)
STUDENT_NAME	varchar(100)
COMPANY_NAME	varchar(100)
STIPEND	double

Fig 5.2.6: Student Internship Table

Registered interns table:

```
CREATE TABLE IF NOT EXISTS registered_interns
(
    STUDENT_ID varchar(50) NOT NULL,
    STUDENT_NAME varchar(100) NOT NULL,
    COMPANY_NAME varchar(100) NOT NULL,
    PRIMARY KEY (STUDENT_ID, COMPANY_NAME)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

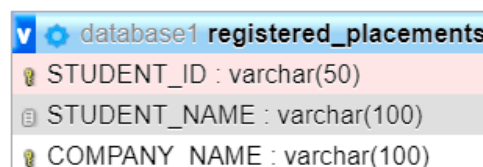


database1 registered_interns	
STUDENT_ID	varchar(50)
STUDENT_NAME	varchar(100)
COMPANY_NAME	varchar(100)

Fig 5.2.7: Registered Interns Table

Registered placements table:

```
CREATE TABLE IF NOT EXISTS registered_placements
(
    STUDENT_ID varchar(50) NOT NULL,
    STUDENT_NAME varchar(100) NOT NULL,
    COMPANY_NAME varchar(100) NOT NULL,
    PRIMARY KEY (STUDENT_ID, COMPANY_NAME)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```



database1 registered_placements	
STUDENT_ID	varchar(50)
STUDENT_NAME	varchar(100)
COMPANY_NAME	varchar(100)

Fig 5.2.8: Registered Placements Table

Intern notification table:

```
CREATE TABLE IF NOT EXISTS intern_notification  
(  
    noti varchar(200),  
    PRIMARY KEY (noti)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```



Fig 5.2.9: Intern notification Table

Placement notification table:

```
CREATE TABLE IF NOT EXISTS place_notification  
(  
    noti varchar(200),  
    PRIMARY KEY (noti)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

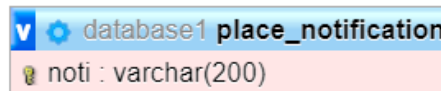


Fig 5.2.10: Placement Notification Table

CHAPTER 6

TESTING AND VALIDATION

6.1 Use case Diagram

Use case diagrams model the functionality of a system using actors and use cases. Use cases are a set of actions, services, and functions that the system needs to perform. In this context, a "system" is something being developed or operated, such as a web site. The "actors" are people or entities operating under defined roles within the system.

Use case diagram for Student:

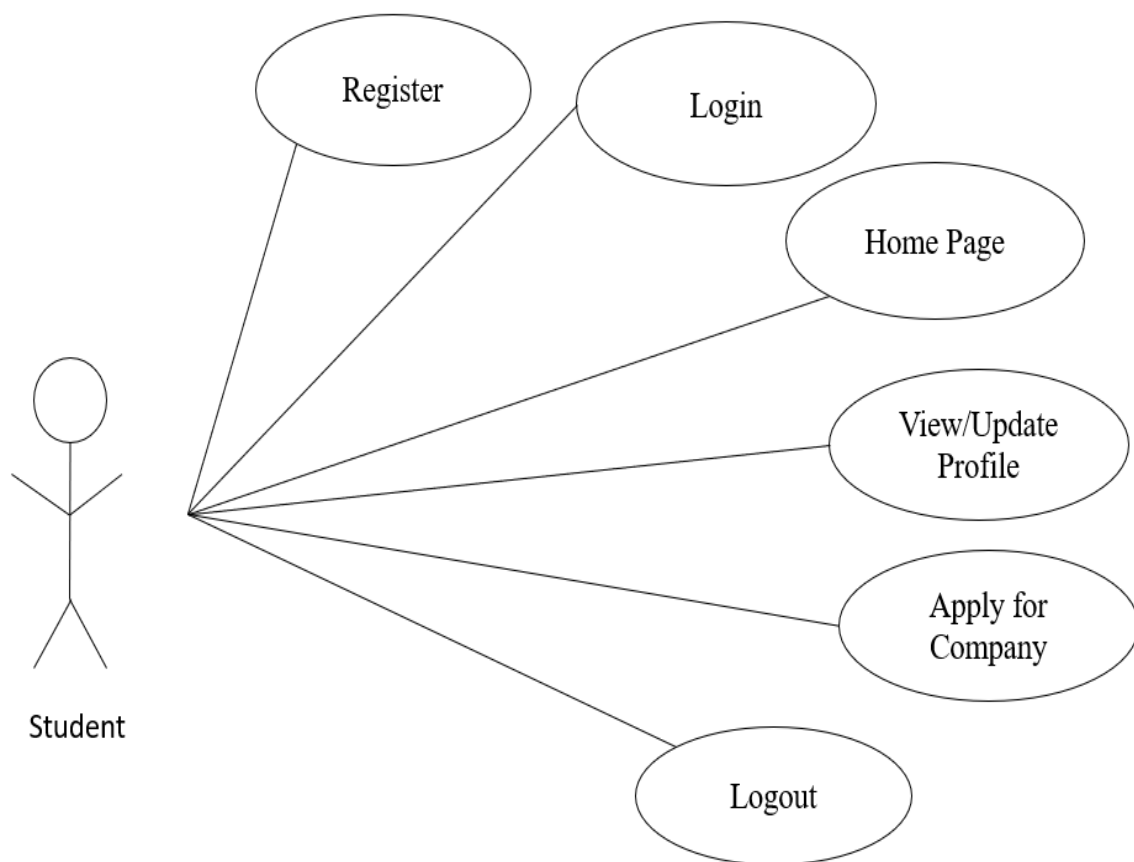


Fig 6.1.1: Student use case diagram

Use case diagram for Company:

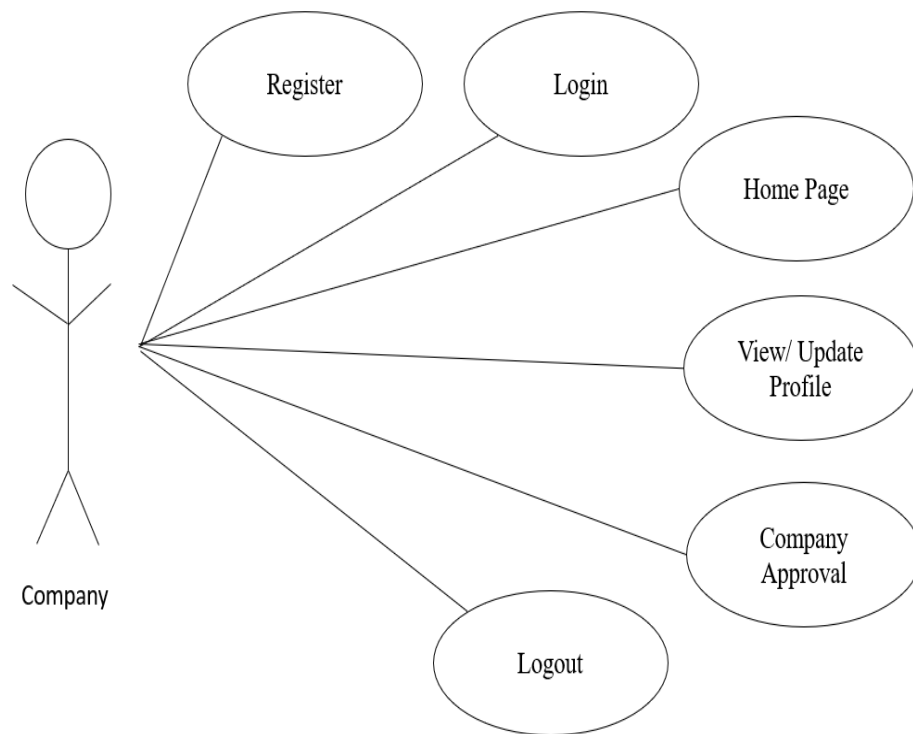


Fig 6.1.2: Company use case diagram

Use case Diagram for admin:

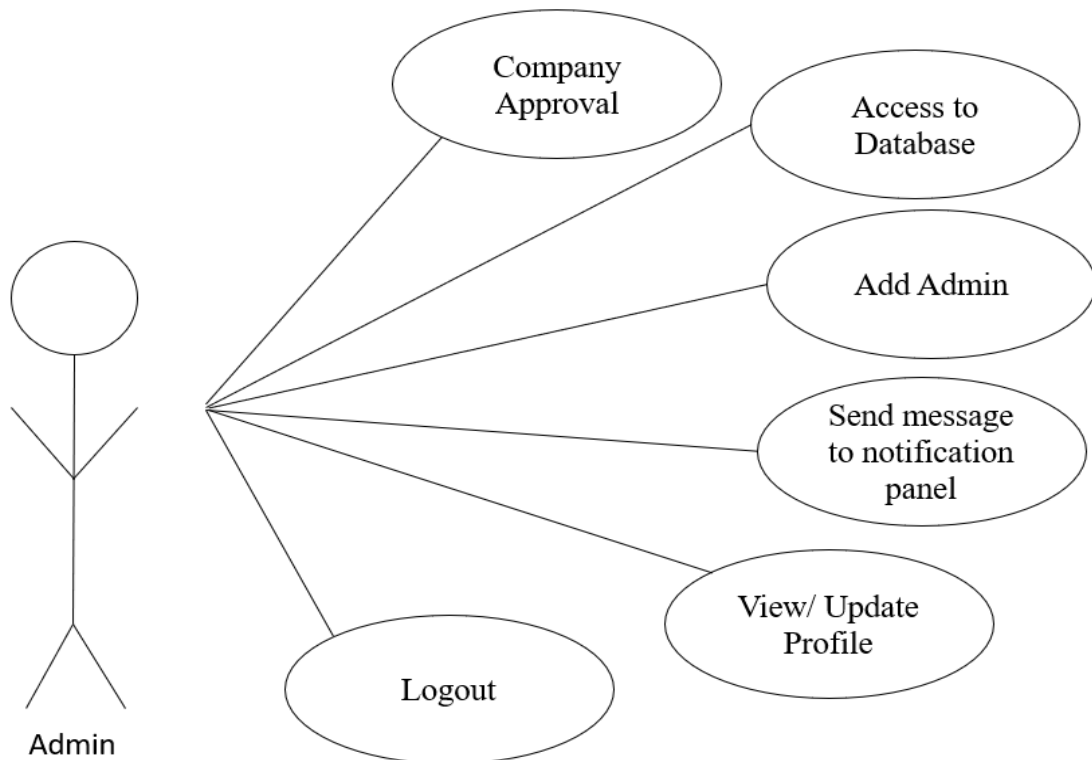


Fig 6.1.3: Admin use case diagram

6.2 Validation

- The user id and password cannot be blank while logging into the site.
- In email id field, '@' character is mandatory.
- In modify password page, user has to specify user id, old password as well as the new password.
- In edit user or new account page, some fields are mandatory like login id, password, student id, company id, admin id, etc.

CHAPTER 7

CONCLUSION AND FUTURE SCOPE

Presently this application is designed to be very User Friendly. Many features are enhanced to the College placement management system. With this application most of the TPO's time is saved. The features of the system can be further enhanced in many ways. The documentation that has enclosed can enable even a person with minimum knowledge to understand it well.

Future Scope:

Presently, this application is run on local host. More features can be added to improvise it before launching on public platform.