

# VISVESVARAYA TECHNOLOGICAL UNIVERSITY

Jnana Sangama, Santhibastawad Road, Machhe, Belagavi 59001, Karnataka, India



DBMS LABORATORY WITH MINI PROJECT (18CSL58) REPORT

ON

“College Placement Management System”

Submitted in the partial fulfilment of the requirements for the award of the degree of

**BACHELOR OF ENGINEERING**  
**IN**  
**INFORMATION SCIENCE AND ENGINEERING**

For the Academic Year 2022-2023

Submitted by

**ABHAY UMESH HEGDE**  
**ADARSH**

**1JS20IS001**  
**1JS20IS005**

Under the Guidance of

**Dr. Sowmya K N**  
Associate Professor, Dept. of ISE, JSSATEB



2022-2023

DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING

**JSS ACADEMY OF TECHNICAL EDUCATION**

JSS Campus, Dr. Vishnuvardhan Road, Bengaluru-560060

**JSS MAHAVIDYAPEETHA, MYSURU**  
**JSS ACADEMY OF TECHNICAL EDUCATION**

**JSS Campus, Dr Vishnuvardhan Road, Bengaluru-560060**

**DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING**



**CERTIFICATE**

This is to certify that DBMS LABORATORY WITH MINI PROJECT (18CSL58) Report entitled “College Placement Management System” is a Bona-fide work carried out by **ABHAY UMESH HEGDE** [1JS20IS001], **ADARSH** [1JS20IS005] in partial fulfilment for the award of degree of Bachelor of Engineering in Information Science and Engineering of Visvesvaraya Technological University Belagavi during the year 2022- 2023.

**Signature of the Guide**

**Dr Sowmya K N**  
**Associate Professor**  
**Dept. of ISE**  
**JSSATEB**

**Signature of the HOD**

**Dr Rekha P M**  
**Professor & HOD**  
**Dept. of ISE**  
**JSSATEB**

# TABLE OF CONTENTS

ACKNOWLEDGEMENT	5
ABSTRACT	6
CHAPTER <sub>1</sub>	7
INTRODUCTION	7
HISTORY OF DBMS	7
AIMS & OBJECTIVES	8
PROJECT SCOPE AND OVERVIEW	9
CHAPTER 2	11
LITERATURE SURVEY	11
INTRODUCTION	11
NORMALIZATION	15
CHAPTER 3	17
REQUIREMENT SPECIFICATIONS	17
NON FUNCTIONAL REQUIREMENTS	17
FUNCTIONAL REQUIREMENTS	17
CHAPTER 4	18
SYSTEM DESIGN	18
INTRODUCTION TO SYSTEM DESIGN	18
ATTRIBUTES	19
SCHEMA DIAGRAM	20
ER DIAGRAM & WIRE FRAMES	21
CHAPTER 5	23
PROJECT IMPLIMENTATION	23
INTRODUCTION	23
QUERIES	28
TRIGGERS	34
CHAPTER 6	41
SYSTEM TESTING	41
INTRODUCTION	41
TYPES OF TESTING	41
CHAPTER 7	43
RESULTS AND DISCUSSIONS	43
CHAPTER 8	48
CONCLUSION AND FUTURE ENHANCEMENTS	48
CONCLUSION	48
FUTURE ENHANCEMENT	48

CHAPTER 9	49
REFERENCES	49
BOOK REFERENCES	49
WEB REFERENCES	49

## ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mention of the people who made it possible. so with gratitude, we acknowledge all those whose guidance and encouragement crowned our efforts with success.

First and foremost, we would like to thank his **Holiness Jagadguru Sri Shivarathri Deshikendra Mahaswamiji** and **Dr. Bhimasen Soragaon**, Principal, JSSATE, Bangalore for providing an opportunity to carry out the Project Work as a part of our curriculum in the partial fulfilment of the degree course.

We express our sincere gratitude for our beloved Head of the department, **Dr. Rekha P M**, for her co-operation and encouragement at all the moments of our approach.

It is our pleasant duty to place on record our deepest sense of gratitude to our respected guide **Dr. Sowmya K N**, Associate Professor for the constant encouragement, valuable help and assistance in every possible way.

We would like to thank all ISE Department Teachers, **Mrs. Sowmya G** madam, our laboratory instructor and other non-teaching staff for providing us with their valuable guidance and for being there at all stages of our work.

ABHAY UMESH HEGDE [1JS20IS001]

ADARSH [1JS20IS005]

## 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 TO DBMS

### HISTORY OF DBMS:

#### Early Work

Early Work assumed that files were on tape. Access was sequential and the cost of access grew in direct proportion to the size of the file.

#### The emergence of Disks and Indexes

As files grew very large, unaided sequential access was not a good solution. Disks allowed for direct access. Indexes made it possible to keep a list of keys and pointers in a small file that could be searched very quickly. With the key and pointer, the user had direct access to the large, primary file.

#### The emergence of Tree Structures

As indexes also have a sequential flavour, when they grew too much, they also became difficult to manage. The idea of using tree structures to manage the index emerged in the early 60's. However, trees can grow very unevenly as records are added and deleted, resulting in long searches requiring many disk accesses to find a record.

#### Balanced Trees

In 1963, researchers came up with the idea of AVL trees for data in memory. AVL trees, however, did not apply to files because they work well when tree nodes are composed of single records rather than dozens or hundreds of them. In the 1970's came the idea of B-Trees which require an  $O(\log_k N)$  access time where  $N$  is the number of entries in the file and  $k$ , the number of entries indexed in a single block of the B-Tree structure --> B-Trees can guarantee that one can find one file entry among millions of others with only 3 or 4 trips to the disk.

#### Hash Tables

Retrieving entries in 3 or 4 accesses is good, but it does not reach the goal of accessing data with a single request. From early on, Hashing was a good way to reach this goal with files that do not change size greatly over time. Recently, Extendible Dynamic Hashing guarantees one or at most two disk accesses no matter how big a file becomes.

#### 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.

### **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.

### **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.

### **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.

### **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.

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

### 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.

### Disadvantages

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

## CHAPTER 2

# LITERATURE SURVEY

### INTRODUCTION:

To understand Database Management System (DBMS), it is necessary to know about Database and even before that, about Data. Data is any facts or figures which can be recorded and which has a latent meaning. Database is a collection of such data, more accurately it is a collection of related data. For e.g. a database of countries of the world and their calling code. Here, countries and calling code are related to each other and the database serves a purpose, to provide user with correct calling code. A random collection where, data is neither related nor serve a purpose cannot be called as a database. In summary a database is a collection of logically coherent data, which come together to serve a particular purpose and they emulate some aspect of real world.

Software which is used to manage such databases is called DBMS. For e.g. Oracle, IBM DB2, Microsoft Access, Microsoft SQL Server, Sybase. DBMS software is a collection of programs which facilitate in access, retrieval, security, creation and sharing of database among its users or other applications. Primary Goal of DBMS Software is to provide an efficient and convenient way of storing and retrieving data.

Here are four types of database languages and their uses:

#### 1.Data definition language (DDL)

Data definition language (DDL) creates the framework of the database by specifying the database schema, which is the structure that represents the organization of data. Its common uses include the creation and alteration of tables, files, indexes and columns within the database. This language also allows users to rename or drop the existing database or its components. Here's a list of DDL statements:

- 1.CREATE: Creates a new database or object, such as a table, index or column
- 2.ALTER: Changes the structure of the database or object
- 3.DROP: Deletes the database or existing objects
- 4.RENAME: Renames the database or existing objects

## 2.Data manipulation language (DML)

Data manipulation language (DML) provides operations that handle user requests, offering a way to access and manipulate the data that users store within a database. Its common functions include inserting, updating and retrieving data from the database. Here's a list of DML statements

- 1.INSERT: Adds new data to the existing database table
- 2.DELETE: Removes records or rows from the table
- 3.SELECT: Retrieves data from the table or multiple tables

## 3. Data control language (DCL)

Data control language (DCL) controls access to the data that users store within a database. Essentially, this language controls the rights and permissions of the database system. It allows users to grant or revoke privileges to the database. Here's a list of DCL statements:

- 1.GRANT: Gives a user access to the database
- 2.REVOKE: Removes a user's access to the database

## 4. Transaction control language (TCL)

1.Transaction control language (TCL) manages the transactions within a database. Transactions group a set of related tasks into a single, executable task. All the tasks must succeed in order for the transaction to work. Here's a list of TCL statements:

- 2.COMMIT: Carries out a transaction
- 3.ROLLBACK: Restores a transaction if any tasks fail to execute

## MySQL

A relational database organizes data into one or more data tables in which data may be related to each other; these relations help structure the data. SQL is a language programmers use to create, modify and extract data from the relational database, as well as control user access to the database. In addition to relational databases and SQL, an RDBMS like MySQL works with an operating system to implement a relational database in a computer's storage system, manages users,allows for network access and facilitates testing database integrity and creation of backups.



## PHP

PHP code is usually processed on a web server by a PHP interpreter implemented as a module, a daemon or as a Common Gateway Interface (CGI) executable. On a web server, the result of the interpreted and executed PHP code – which may be any type of data, such as generated HTML or binary image data – would form the whole or part of an HTTP response. Various web template systems, web content management systems, and web frameworks exist which can be employed to orchestrate or facilitate the generation of that response. Additionally, PHP can be used for many programming tasks outside the web context, such as standalone graphical applications and robotic drone control. PHP code can also be directly executed from the command line.



## CSS

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language such as HTML or XML (including XML dialects such as SVG, MathML or XHTML). CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript. CSS is designed to enable the separation of content and presentation, 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, which reduces complexity and repetition in the structural content; and enable the .css file to be cached to improve the page load speed between the pages that share the file and its formatting.



## JAVASCRIPT

JavaScript often abbreviated as JS, is a programming language that is one of the core technologies of the World Wide Web, alongside HTML and CSS. As of 2022, 98% of websites use JavaScript on the client side for webpage behaviour, often incorporating third-party libraries. All major web browsers have a dedicated JavaScript engine to execute the code on users' devices.

JavaScript is a high-level, often just-in-time compiled language that conforms to the ECMAScript standard. It has dynamic typing, prototype-based object-orientation, and first-class functions. It is multi-paradigm, supporting event-driven, functional, and imperative programming styles. It has application programming interfaces (APIs) for working with text, dates, regular expressions, standard data structures, and the Document Object Model (DOM).



## HTML

The Hyper Text Markup Language or HTML is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript. Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document. HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects such as interactive forms may be embedded into the rendered page. HTML provides a means to create structured documents by denoting

structural semantics for text such as headings, paragraphs, lists, links, quotes, and other items.



## Windows 11

Windows 11 is the latest major release of Microsoft's Windows NT operating system, released in October 2021. It is a free upgrade to its predecessor, Windows 10 (2015), and is available for any Windows 10 devices that meet the new Windows 11 system requirements.

Windows 11 features major changes to the Windows shell influenced by the cancel Windows 10X, including a redesigned Start menu, the replacement of its "live tiles" with a separate "Widgets" panel on the taskbar, the ability to create tiled sets of windows that can be minimized and restored from the taskbar as a group, and new gaming technologies inherited from Xbox Series X and Series S such as Auto HDR and Direct Storage on compatible hardware. Internet Explorer (IE) has been replaced by the Chromium-based Microsoft Edge as the default web browser, like its predecessor, Windows 10, and Microsoft Teams is integrated into the Windows shell. Microsoft also announced plans to allow more flexibility in software that can be distributed via the Microsoft Store and to support Android apps on Windows 11 (including a partnership with Amazon to make its app store available for the function).



## NORMALIZATION:

Database normalization is a database schema design technique, by which an existing schema is modified to minimize redundancy and dependency of data. Normalization involves decomposing a table into

less redundant (and smaller) tables without losing information; defining foreign keys in the old table referencing the primary keys of the new ones. The objective is to isolate data so that additions, deletions, and modifications of an attribute can be made in just one table and then propagated through the rest of the database using the defined foreign keys.

## Database normalization types

### First Normal Form (1NF)

First normal form (1NF) is a property of a relation in a relational database. A relation is in first normal form if the domain of each attribute contains only atomic values, and the value of each attribute contains only a single value from that domain.

Edgar Codd, in a 1971 conference paper, defined a relation in first normal form to be one such that none of the domains of that relation should have elements which are themselves sets.

First normal form enforces these criteria:

1. Eliminate repeating groups in individual tables.
2. Create a separate table for each set of related data.
3. Identify each set of related data with a primary key

### Second Normal Form (2NF)

A table that is in first normal form (1NF) must meet additional criteria if it is to qualify for second normal form. Specifically: a table is in 2NF if it is in 1NF and no non-prime attribute is dependent on any proper subset of any candidate key of the table. A non-prime attribute of a table is an attribute that is not a part of any candidate key of the table. Put simply, a table is in 2NF if it is in 1NF and every non-prime attribute of the table is dependent on the whole of every candidate key.

### Boyce-Codd Normal Form (BCNF)

When a relation has more than one candidate key, anomalies may result even though the relation is in 3NF. 3NF does not deal satisfactorily with the case of a relation with overlapping candidate keys i.e. composite candidate keys with at least one attribute in common. BCNF is based on the concept of a determinant.

### Third Normal Form (3NF)

Third normal form is a normal form used in normalizing a database design to reduce the duplication of data and ensure referential integrity by ensuring that the entity is in second normal form and all the attributes in a table are determined only by the candidate keys of that table and not by any non-prime attributes.

### Fourth Normal Form (4NF)

It is a normal form used in database normalization Introduced by Ronald Fagin in 1977, 4NF is the next level of normalization after Boyce–Codd normal form (BCNF). Whereas the second, third, and Boyce–Codd normal forms are concerned with functional dependencies, 4NF is concerned with a more general type of dependency known as a multivalued dependency

### Fifth Normal Form (5NF)

It is also known as project-join normal form (PJ/NF) is a level of database normalization designed to reduce redundancy in relational databases recording multi-valued facts by isolating semantically related multiple relationships. A table is said to be in the 5NF if and only if every nontrivial join dependency in that table is implied by the candidate keys.

**1.admin** (ADMIN\_ID, ADMIN\_NAME, A\_PASSWORD, POST, EMAIL, CONTACT\_NO, DOB, QUALIFICATION)

1.Primary key is ADMIN\_ID and no other functional dependency is present. Therefore, all values are atomic. Relation is in 1NF.

2.Now, keys = {ADMIN\_ID, (ADMIN\_NAME, DOB), ...} Therefore, there is no partial functional dependency. Relation is in 2NF.

3.Therefore, there is no transitive dependency. Relation is in 3NF.

**2.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)

1. Primary key is STUDENT\_ID and no other functional dependency is there. Therefore, all values are atomic. Relation is in 1NF

2.Now, keys = {STUDENT\_ID, (STUDENT\_NAME, FATHER NAME), (STUDENT \_ID, DOB), ...} Therefore, there is no partial functional dependency. Relation is in 2NF.

3.Therefore, there is no transitive dependency. Relation is in 3NF.

**3.company** (COMPANY\_ID, COMPANY\_NAME, C\_PASSWORD, WEBSITE, ADDRESS, STATUS, COMING\_DATE, APPROVAL) COMPANY\_NAME WEBSITE, ADDRESS

1.Primary key is COMPANY\_ID. Therefore, all values are atomic. Relation is in 1NF.

2.Now, keys = {COMPANY\_ID, (COMPANY\_NAME, WEBSITE), ...} Therefore, there is no partial functional dependency. Relation is in 2NF.

3.Therefore, there is no transitive functional dependency. Relation is in 3NF.



**4.company\_branch** (COMPANY\_NAME, C\_TYPE, BRANCH, MIN\_CGPA, MAX\_ BACKLOGS, MAX\_SALARY, MAX\_STIPEND, JOB\_PROFILE, PLACE\_OF\_POSTING)

1.Primary key is COMPANY\_NAME and no other functional dependency is there. Therefore, all values are atomic. Relation is in 1NF.

2.Now, keys = {(COMPANY\_NAME, C\_TYPE, BRANCH)} Therefore, there is no partial functional dependency. Relation is in 2NF.

3.Therefore, there is transitive functional dependency. Relation is in 3NF.

**5.student\_placement** (STUDENT\_ID, COMPANY\_ID, STUDENT\_NAME, COMPANY\_ NAME, PACKAGE)

1.Primary key STUDENT\_ID and no other functional dependency is there. Therefore, all values are atomic. Relation is in 1NF.

2.Now, keys = {STUDENT\_ID, (STUDENT\_NAME, COMPANY\_ID), ...} Therefore, there is no partial functional dependency. Relation is in 2NF.

3.Therefore, there is transitive functional dependency. Relation is in 3NF.

**6.student\_internship** (STUDENT\_ID, COMPANY\_ID, STUDENT\_NAME, COMPANY\_ NAME, STIPEND)

1.Primary key STUDENT\_ID and no other functional dependency is there. Therefore, all values are atomic. Relation is in 1NF.

2.Now, keys = {STUDENT\_ID, (STUDENT\_NAME, COMPANY\_ID), ...} Therefore, there is no partial functional dependency. Relation is in 2NF.

3.Therefore, there is transitive functional dependency. Relation is in 3NF.

### 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 $\rightarrow$ C_TYPE, BRANCH, MIN_CGPA, MAX_BACKLOGS, MAX_SALARY, MAX_STIPEND, JOB_PROFILE, PLACE_OF_POSTING
Student_placement	STUDENT_ID $\rightarrow$ COMPANY_ID, STUDENT_NAME, COMPANY_NAME, PACKAGE
Student_internship	STUDENT_ID $\rightarrow$ COMPANY_ID, STUDENT_NAME, COMPANY_NAME, STIPEND
Registered_placements	STUDENT_ID $\rightarrow$ STUDENT_NAME, COMPANY_NAME
Registered_interns	STUDENT_ID $\rightarrow$ STUDENT_NAME, COMPANY_NAME

## CHAPTER 3

### REQUIREMENT SPECIFICATIONS

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.

#### System Requirements:

##### 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.

##### Functional Requirements

- **For the students:** o Student can register at training and placement cell. o Can login and view his/her profile. o Can update his/her information that were provided at the time of registration. o Student can view general notifications from T&P cell and can also see the details of companies that are visiting. o A student can apply for the visiting company for internship or placement if he/she is eligible for that.
- **For the companies:** o A company can register at training and placement cell. o Companies add basic details such as the eligibility criteria, max number of backlogs etc.
- **For the admin:** o Can add new admin at training and placement cell. o Add general notifications for the students. o Have access to all the information about students and companies i.e., admin can view or update information of students and company. o Only the admin gives approval to the company after verifying that the company details. o Admin can change the status of the students after receiving final result from the company. o 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 - 12 - Department of ISE JSSATE,BANGLORE 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 4

### SYSTEM DESIGN

#### INTRODUCTION TO SYSTEM DESIGN:

System design is the phase that bridges the gap between problem domain and the existing system in a manageable way. This phase focuses on the solution domain, i.e. “how to implement?” It is the phase where the SRS document is converted into a format that can be implemented and decides how the system will operate. In this phase, the complex activity of system development is divided into several smaller sub-activities, which coordinate with each other to achieve the main objective of system development.

Systems design is the process or art of defining the architecture, components modules, interfaces, and data for a system to satisfy specified requirements. One could see it as the application of systems theory to product development. Database design can be generally defined as a collection of tasks or processes that enhance the designing, development, implementation, and maintenance of enterprise data management system. Designing a proper database reduces the maintenance cost thereby improving data consistency and the cost-effective measures are greatly influenced in terms of disk storage space. Therefore, there has to be a brilliant concept of designing a database. The designer should follow the constraints and decide how the elements correlate and what kind of data must be stored.

Database Design is a collection of processes that facilitate the designing, development, implementation and maintenance of enterprise data management systems. Properly designed database are easy to maintain, improves data consistency and are cost effective in terms of disk storage space. The database designer decides how the data elements correlate and what data must be stored.

The main objectives of database designing are to produce logical and physical designs models of the proposed database system.

#### ATTRIBUTES:

Attributes define the properties of a data object and take on one of three different characteristics.

They can be used to:

- Name an instance of data object.
- Describe the instance.

The following figure shows the Relational model for College Placement Management System.

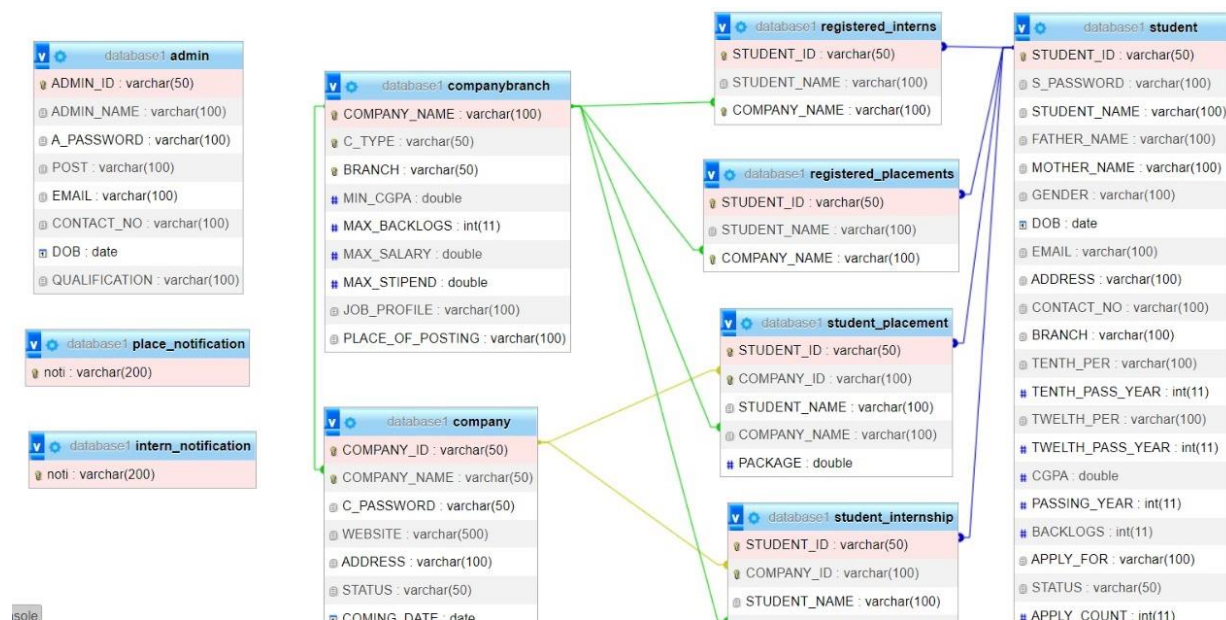


Fig 4.1 Displays the relation

Relational Model was proposed by E.F. Codd to model data in the form of relations or tables. After designing the conceptual model of Database using ER diagram, we need to convert the conceptual model in the relational model which can be implemented using any RDBMS languages like Oracle SQL, MySQL etc. So we will see what Relational Model is. Relational Model represents how data is stored in Relational Databases. A relational database stores data in the form of relations (tables).

## SCHEMA DIAGRAM:

A database schema is the skeleton structure that represents the logical view of the entire database. A database schema defines its entities and the relationship among them. It contains a descriptive detail of the database, which can be depicted by means of schema diagrams.

A schema diagram contains entities and the attributes that will define that schema. It 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.

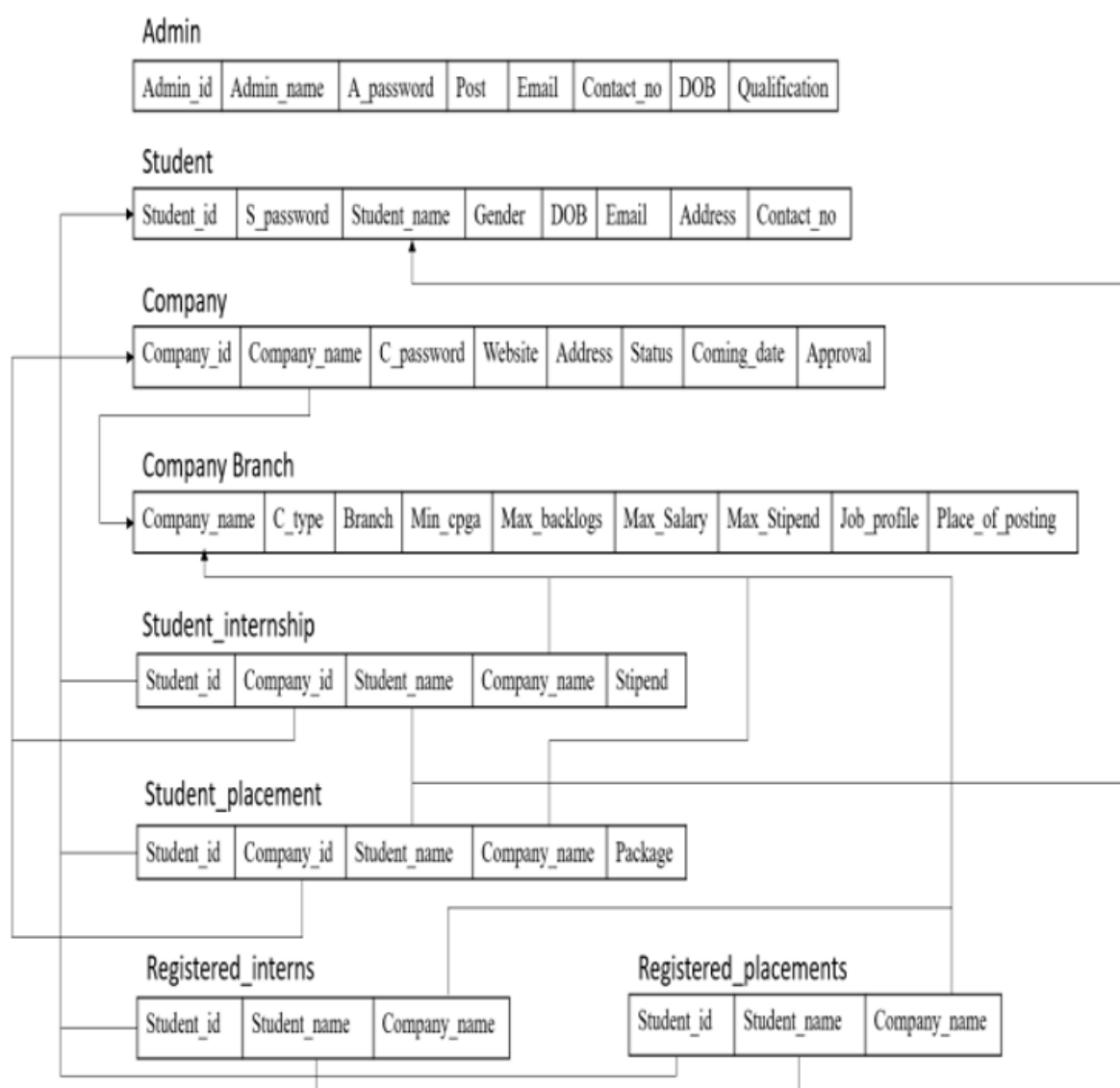


Fig 4.2 schema for College Placement Management System

A database schema is a structure that represents the logical storage of the data in a database. It represents the organization of data and provides information about the relationships between the tables in a given database. In this topic, we will understand more about database schema and its types. The states of a created conceptual schema are transformed into an explicit mapping, the database schema. This describes how real-world entities are modelled in the database. All the various table used are described in the following schema. The necessary Primary key's and the corresponding Foreign keys are also represented.

## E-R DIAGRAM:

An Entity Relationship (ER) Diagram is a type of flowchart that illustrates how “entities” such as people, objects or concepts relate to each other within a system. ER Diagrams are most often used to design or debug relational databases in the fields of software engineering, business information systems, education and research. Also known as ERDs or ER Models, they use a defined set of symbols such as rectangles, diamonds, ovals and connecting lines to depict the interconnectedness of entities, relationships and their attributes.

They mirror grammatical structure, with entities as nouns and relationships as verbs. ER diagrams are related to data structure diagrams (DSDs), which focus on the relationships of elements within entities instead of relationships between entities themselves. ER diagrams also are often used in conjunction with data flow diagrams (DFDs), which map out the flow of information for processes or systems.

Rectangle: Represents Entity sets.

Ellipses: Attributes

Diamonds: Relationship Set

Lines: They link attributes to Entity Sets and Entity sets to Relationship Set

Double Ellipses: Multivalued Attributes

Dashed Ellipses: Derived Attributes

Double Rectangles: Weak Entity Sets

Double Lines: Total participation of an entity in a relationship set



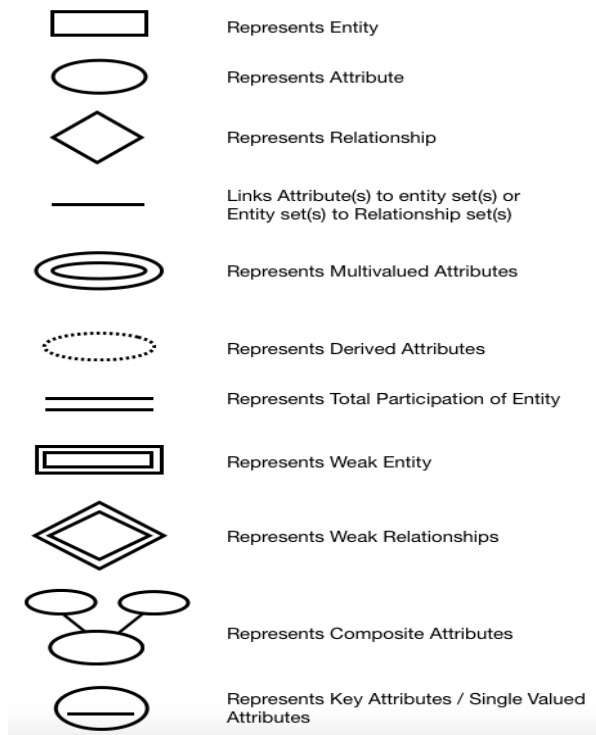


Fig 4.3 entity symbols

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.

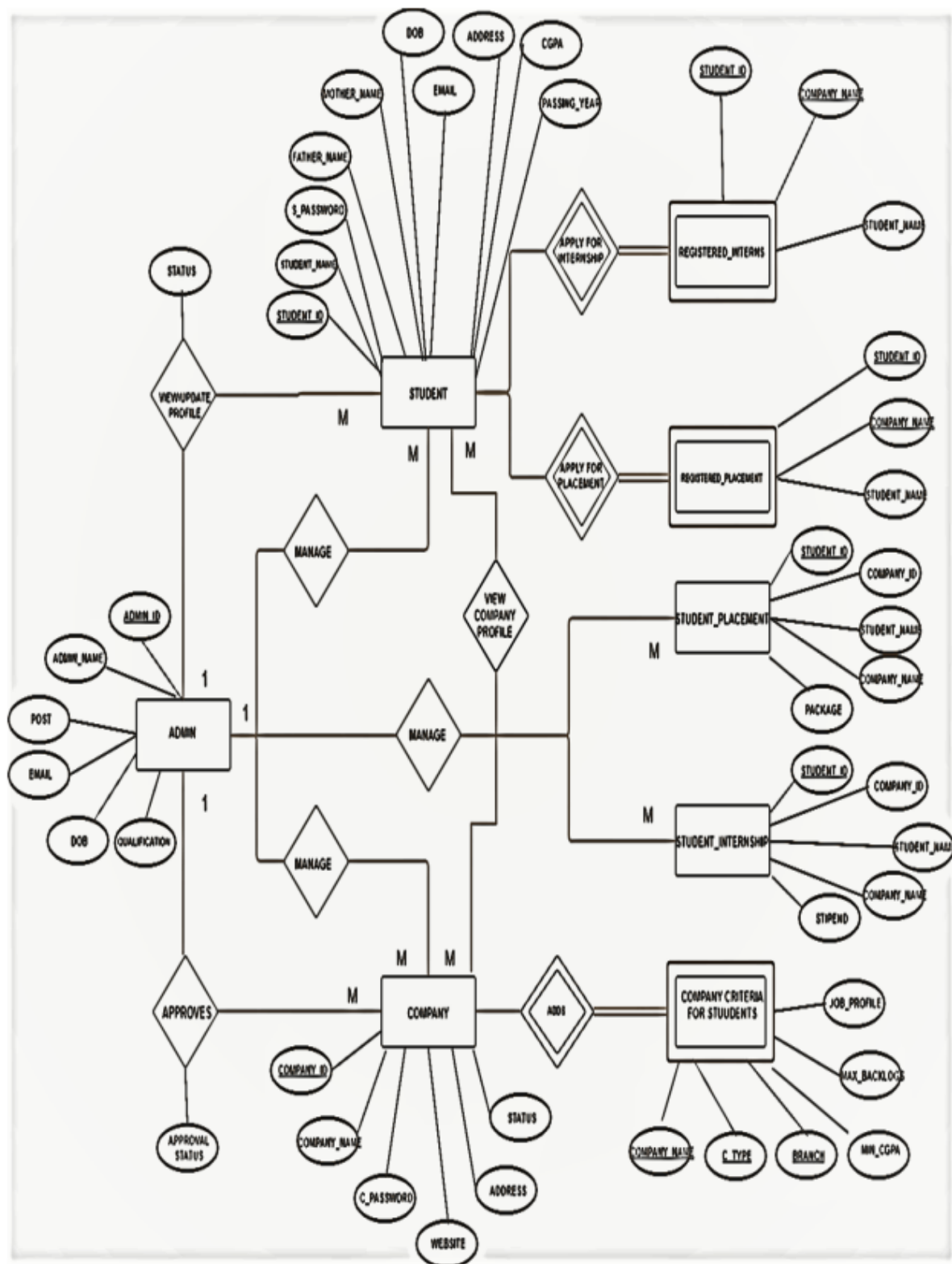


Fig 4.3 E-R diagram for College Placement Management System

Each row of a relation/table represents a record, and each column represents an attribute of data. The Structured Query Language (SQL) is used to manipulate relational databases. The design of a relational database is composed of four stages, where the data are model into a set of related tables.

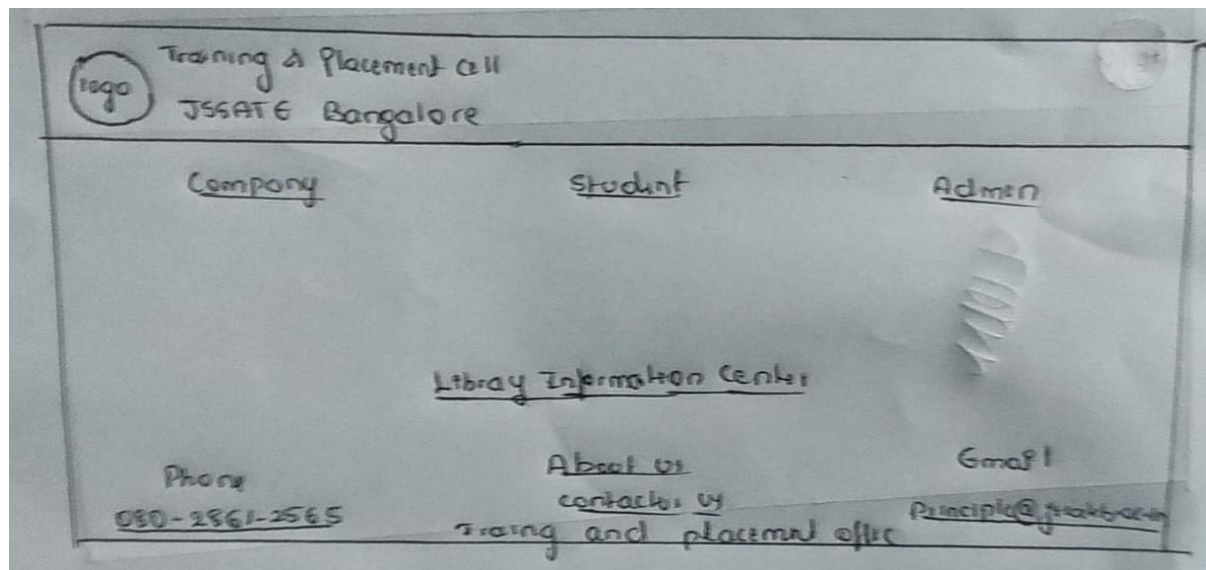
WIRE FRAMES:

Fig 4.4 Home page

Fig shows the home page of the college placement management project. Home pages shows the following three categories: student, company and admin. And also here we can get the information of library of our college.

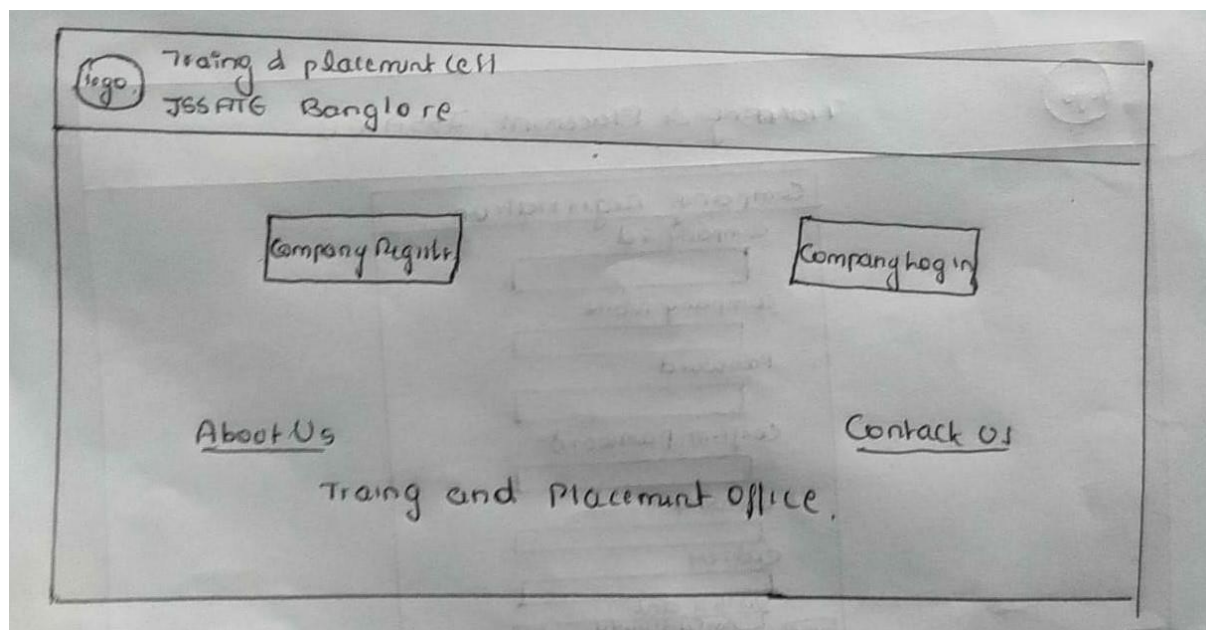


Fig 4.5 view of company index page

Fig shows the company index page of the college placement management project. In this page there are two options: Company register and login. This page can only be used by the company itself.

Hand-drawn sketch of the company registration page. The header includes a "logo" in a circle and the text "Training & Placement, JSSATEB". The main form is titled "Company Registration" and contains fields for "Company id.", "Company Name", "Password", "Confirm password", "Website", "Address", and "Visiting date". A "Submit" button is at the bottom left.

Fig 4.6 company registration page

Fig shows the company registration page of the college placement management project. This registration form only for company.

Hand-drawn sketch of the admin index page. The header includes a "logo" in a circle and the text "Training & Placement cell JSSATE, Bangalore". The main content area has buttons for "Admin Register" and "Admin login". At the bottom, there are links for "About us" and "Contact us", and the text "Training & Placement office".

Fig 4.7 index page of admin

Fig shows the admin index page of the college placement management project. Here we can see two options: Admin register and admin login.



Fig 4.8 company login page

Fig shows the company login page of the college placement management project. This page can be used after filling the registration form only.

Fig 4.9 company login page

Fig shows the student index page of the college placement management project. Here we can see three options: student register, student login for placement, and student login for internship.

The sketch shows a web page titled "Training & Placement, JSSATE" with a logo on the left. The main content area is titled "Student Log for Internship". It contains four input fields: "Student Name", "Student Id", "Apply for [ ] Internship", and "Password". Below these fields is a "Sign In" button. The fields are arranged vertically and are enclosed in a rectangular box.

Fig 4.9.1 student login page

Fig shows the student login for internship page of the college placement management project. Here student can login the page for apply internship.

The sketch shows a web page titled "Training & Placement JSSATE" with a logo on the left. The main content area is titled "Admin Login". It contains three input fields: "Admin Name", "Admin Id", and "Password". Below these fields is a "Sign In" button. The fields are arranged vertically and are enclosed in a rectangular box.

Fig 4.9.2 admin login page

Fig shows the admin login page of the college placement management project. Here admin can login the page only. After sign in of this page admin can get the next page.

## CHAPTER 5

### PROJECT IMPLEMENTATION

#### INTRODUCTION:

Implementation is the stage where the theoretical design is turned into a working system. The most crucial stage in achieving a new successful system and in giving confidence on the new system for the users that it will work effectively.

The system can be implemented only after thorough testing is done and if it is found to work according to the specification. It involves careful planning, investigating of the current system and its constraints on implementation, design of methods to achieve the change and an evaluation of change over methods a part from planning. Two major tasks of preparing the implementation are education and training of the users and testing of the system.

The following codes will ensure the complete implementation of our design and the project.

#### CREATING TABLES:

##### **TABLE ADMIN**

```
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=Inn DEFAULT CHARSET=utf8;
```

```
SELECT * FROM admin;
```


























	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1	ADMIN_ID 	varchar(50)	utf8_general_ci		No	None			 Change  Drop  More
<input type="checkbox"/>	2	ADMIN_NAME	varchar(100)	utf8_general_ci		No	None			 Change  Drop  More
<input type="checkbox"/>	3	A_PASSWORD	varchar(100)	utf8_general_ci		No	None			 Change  Drop  More
<input type="checkbox"/>	4	POST	varchar(100)	utf8_general_ci		No	None			 Change  Drop  More
<input type="checkbox"/>	5	EMAIL	varchar(100)	utf8_general_ci		No	None			 Change  Drop  More
<input type="checkbox"/>	6	CONTACT_NO	varchar(100)	utf8_general_ci		No	None			 Change  Drop  More
<input type="checkbox"/>	7	DOB	date			No	None			 Change  Drop  More
<input type="checkbox"/>	8	QUALIFICATION	varchar(100)	utf8_general_ci		No	None			 Change  Drop  More

Fig 5.1 structure of admin table

Snapshot represent the attributes of admin table, where the admin\_ id is primary key. The table is used to store data related to admin as shown in figure.

## TABLE STUDENT

```

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;

```

```
SELECT * FROM student;
```

<input type="checkbox"/>	1	<b>STUDENT_ID</b>	varchar(50)	utf8_general_ci	No	None	Change	Drop	More
<input type="checkbox"/>	2	<b>S_PASSWORD</b>	varchar(100)	utf8_general_ci	No	None	Change	Drop	More
<input type="checkbox"/>	3	<b>STUDENT_NAME</b>	varchar(100)	utf8_general_ci	No	None	Change	Drop	More
<input type="checkbox"/>	4	<b>FATHER_NAME</b>	varchar(100)	utf8_general_ci	No	None	Change	Drop	More
<input type="checkbox"/>	5	<b>MOTHER_NAME</b>	varchar(100)	utf8_general_ci	No	None	Change	Drop	More
<input type="checkbox"/>	6	<b>GENDER</b>	varchar(100)	utf8_general_ci	No	None	Change	Drop	More
<input type="checkbox"/>	7	<b>DOB</b>	date		No	None	Change	Drop	More
<input type="checkbox"/>	8	<b>EMAIL</b>	varchar(100)	utf8_general_ci	No	None	Change	Drop	More
<input type="checkbox"/>	9	<b>ADDRESS</b>	varchar(100)	utf8_general_ci	No	None	Change	Drop	More
<input type="checkbox"/>	10	<b>CONTACT_NO</b>	varchar(100)	utf8_general_ci	No	None	Change	Drop	More
<input type="checkbox"/>	11	<b>BRANCH</b>	varchar(100)	utf8_general_ci	No	None	Change	Drop	More
<input type="checkbox"/>	12	<b>TENTH_PER</b>	varchar(100)	utf8_general_ci	No	None	Change	Drop	More
<input type="checkbox"/>	13	<b>TENTH_PASS_YEAR</b>	int(11)		No	None	Change	Drop	More
<input type="checkbox"/>	14	<b>TWELTH_PER</b>	varchar(100)	utf8_general_ci	No	None	Change	Drop	More

Fig 5.2 structure of student table

Snapshot represent the attributes of student table, where the student\_ id is primary key. The table is used to store data related to student as shown in figure.

## TABLE COMPANY

```

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;

```

```
SELECT * FROM company;
```

<input type="checkbox"/>	1	<b>COMPANY_ID</b>	varchar(50)	utf8_general_ci	No	None	Change	Drop	More
<input type="checkbox"/>	2	<b>COMPANY_NAME</b>	varchar(50)	utf8_general_ci	No	None	Change	Drop	More
<input type="checkbox"/>	3	<b>C_PASSWORD</b>	varchar(50)	utf8_general_ci	No	None	Change	Drop	More
<input type="checkbox"/>	4	<b>WEBSITE</b>	varchar(500)	utf8_general_ci	No	None	Change	Drop	More
<input type="checkbox"/>	5	<b>ADDRESS</b>	varchar(100)	utf8_general_ci	No	None	Change	Drop	More
<input type="checkbox"/>	6	<b>STATUS</b>	varchar(50)	utf8_general_ci	Yes	visiting	Change	Drop	More
<input type="checkbox"/>	7	<b>COMING_DATE</b>	date		No	None	Change	Drop	More
<input type="checkbox"/>	8	<b>APPROVAL</b>	varchar(50)	utf8_general_ci	Yes	not approved	Change	Drop	More

Fig 5.3 structure of company table

Snapshot represent the attributes of company table, where the company\_ id is primary key. The table is used to store data related to company as shown in figure.

## TABLE COMPANY BRANCH

```

CREATE TABLE IF NOT EXISTS company branch
(
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;

```

```
SELECT * FROM company branch;
```

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1 <b>COMPANY_NAME</b>	varchar(100)	utf8_general_ci		No	None			Change  Drop  More
<input type="checkbox"/>	2 <b>C_TYPE</b>	varchar(50)	utf8_general_ci		No	None			Change  Drop  More
<input type="checkbox"/>	3 <b>BRANCH</b>	varchar(50)	utf8_general_ci		No	None			Change  Drop  More
<input type="checkbox"/>	4 <b>MIN_CGPA</b>	double			Yes	NULL			Change  Drop  More
<input type="checkbox"/>	5 <b>MAX_BACKLOGS</b>	int(11)			Yes	0			Change  Drop  More
<input type="checkbox"/>	6 <b>MAX_SALARY</b>	double			Yes	NULL			Change  Drop  More
<input type="checkbox"/>	7 <b>MAX_STIPEND</b>	double			Yes	NULL			Change  Drop  More
<input type="checkbox"/>	8 <b>JOB_PROFILE</b>	varchar(100)	utf8_general_ci		Yes	NULL			Change  Drop  More
<input type="checkbox"/>	9 <b>PLACE_OF_POSTING</b>	varchar(100)	utf8_general_ci		Yes	NULL			Change  Drop  More

Fig 5.4 structure of company branch

Snapshot represent the attributes of company branch table, where the company\_ name is primary key. The table is used to store data related to company branch details as shown in figure.

## TABLE STUDENT PLACEMENT

```
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=Inn DEFAULT CHARSET=utf8;
```

```
SELECT* FROM student placement;
```

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1	STUDENT_ID 	varchar(50)	utf8_general_ci		No	None			 Change  Drop  More
<input type="checkbox"/>	2	COMPANY_ID 	varchar(100)	utf8_general_ci		No	None			 Change  Drop  More
<input type="checkbox"/>	3	STUDENT_NAME	varchar(100)	utf8_general_ci		No	None			 Change  Drop  More
<input type="checkbox"/>	4	COMPANY_NAME 	varchar(100)	utf8_general_ci		No	None			 Change  Drop  More
<input type="checkbox"/>	5	PACKAGE	double			No	None			 Change  Drop  More

Fig 5.5 structure of student placement table

Snapshot represent the attributes of student placement table, where primary key not present. The table is used to store data related to student placement details as shown in figure.

## TABLE STUDENT INTERNSHIP

```
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=Inn DEFAULT CHARSET=utf8;
```

```
SELECT * FROM student_internship;
```

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1	STUDENT_ID 	varchar(50)	utf8_general_ci		No	None			 Change  Drop  More
<input type="checkbox"/>	2	COMPANY_ID 	varchar(100)	utf8_general_ci		No	None			 Change  Drop  More
<input type="checkbox"/>	3	STUDENT_NAME	varchar(100)	utf8_general_ci		No	None			 Change  Drop  More
<input type="checkbox"/>	4	COMPANY_NAME 	varchar(100)	utf8_general_ci		No	None			 Change  Drop  More
<input type="checkbox"/>	5	STIPEND	double			No	None			 Change  Drop  More

Fig 5.6 structure of student internship table

Snapshot represent the attributes of student internship table, where primary key not present. The table is used to store data related to student internship details as shown in figure.

**TABLE REGISTERED INTERNS**

```
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;
```

```
SELECT * FROM registered interns;
```

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1 STUDENT_ID	varchar(50)	utf8_general_ci		No	None			Change  Drop  More
<input type="checkbox"/>	2 STUDENT_NAME	varchar(100)	utf8_general_ci		No	None			Change  Drop  More
<input type="checkbox"/>	3 COMPANY_NAME	varchar(100)	utf8_general_ci		No	None			Change  Drop  More

Fig 5.7 structure of registered interns table

Snapshot represent the attributes of registered interns table, where primary key not present. The table is used to store data related to registered interns details as shown in figure.

**TABLE REGISTERED PLACEMENT**

```
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=Inn DEFAULT CHARSET=utf8;
```

```
SELECT * FROM registered placement;
```

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1 STUDENT_ID	varchar(50)	utf8_general_ci		No	None			Change  Drop  More
<input type="checkbox"/>	2 STUDENT_NAME	varchar(100)	utf8_general_ci		No	None			Change  Drop  More
<input type="checkbox"/>	3 COMPANY_NAME	varchar(100)	utf8_general_ci		No	None			Change  Drop  More

Fig 5.8 structure of registered placement table

Snapshot represent the attributes of registered placement table, where primary key not present. The table is used to store data related to registered placement details as shown in figure.

## TABLE INTERN NOTIFICATION

```
CREATE TABLE IF NOT EXISTS intern_ notification
```

```
(
noti varchar(200),
PRIMARY KEY (noti)
) ENGINE=Inn DEFAULT CHARSET=utf8;
```

```
SELECT * FROM intern notification;
```

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	noti	varchar(200)	utf8_general_ci		No	None			Change Drop More

With selected: Browse Change Drop Primary Unique Index Spatial Fulltext

Fig 4.9 structure of intern notification table

Snapshot represent the attributes of interns notification table, where primary key not present. The table is used to store data related to interns notification details as shown in figure.

## TABLE PLACEMENT NOTIFICATION

```
CREATE TABLE IF NOT EXISTS place_ notification
```

```
(
Noti varchar(200),
PRIMARY KEY (noti)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
SELECT * FROM placement notification;
```

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	noti	varchar(200)	utf8_general_ci		No	None			Change Drop More

With selected: Browse Change Drop Primary Unique Index Spatial Fulltext

Fig 4.9.1 structure of placement notification table

Snapshot represent the attributes of placement notification table, where primary key not present. The table is used to store data related to placement notification details as shown in figure.

## QUERIES

The below mentioned are all the queries used to perform various tasks in MySQL such as insert, delete, update, triggers and procedures.

```
$query = "INSERT INTO admin
```

```
(ADMIN_ID, ADMIN_NAME ,A_PASSWORD,POST,EMAIL,CONTACT_NO,DOB,QUALIFICATI  
ON)
```

```
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);
```

## TRIGGERS

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

### Syntax:

```
create trigger [trigger_name]
```

```
[before | after]
```

```
{insert | update | delete}
```

```
on [table_name]
```

```
[for each row]
```

```
[trigger_body]
```

Triggers used : The logs that gets activated whenever the user insert data to the student table

1: Trigger name: on insert

Table: student

Time: after

Event: insert

```
INSERT INTO logs VALUES (null, NEW. rid, 'Student Inserted', NOW())
```



## CHAPTER 6

### SYSTEM TESTING

#### INTRODUCTION:

Testing plays a vital role in the success of the system. System testing makes a logical assumption that if all parts of the system are correct, the goal will be successfully achieved. Once program code has been developed, testing begins. The testing process focuses on the logical internals of the software, ensuring that all statements have been tested, and on the functional externals, that is conducted tests to uncover errors and ensure that defined input will produce actual results that agree with required results. Broadly speaking, there are at least three levels of testing: unit testing, integration testing, and system testing.

#### TYPES OF TESTING:

##### Unit testing

Unit testing refers to tests that verify the functionality of a specific section of code, usually at the function level. In an object-oriented environment, this is usually at the class level, and the minimal unit tests include the constructors and destructors. These types of tests are usually written by developers as they work on code (white-box-style), to ensure that the specific function is working as expected. One function might have multiple tests, to catch corner cases or other branches in the code. Unit testing alone cannot verify the functionality of a piece of software, but rather is used to ensure that the building blocks of the software work independently from each other.

Unit testing is a software development process that involves a synchronized application of a broad spectrum of defect prevention and detection strategies in order to reduce software development risks, time, and costs. It is performed by the software developer or engineer during the construction phase of the software development lifecycle. Unit testing aims to eliminate construction errors before code is promoted to additional testing; this strategy is intended to increase the quality of the resulting software as well as the efficiency of the overall development process.

Table 1: unit testing for validation

Sl.no	Test Case	Expected Result	Test Result
1	Enter valid email-id and password and usn	Software should display the index page of student	Successful

	click login in student login page		
2	Enter invalid email-id or password or user name and click login	Software should display invalid email-id or password.	successful

Fig 6.1 unit testing for validation

## Integration Testing

Integration testing is any type of software testing that seeks to verify the interfaces between components against a software design. Software components may be integrated in an iterative way or all together ("big bang"). Normally the former is considered a better practice since it allows interface issues to be located more quickly and fixed. Integration testing works to expose defects in the interfaces and interaction between integrated components (modules). Progressively larger groups of tested software components corresponding to elements of the architectural design are integrated and tested until the software works as a system.

Table 2: Integrating testing for test cases

Sl.no	Test Case Objective	Test Case Description	Expected Result	Output
1	Check the interface link between the Login and company module	Enter the login credentials (email-id and password) and click on login	To be Directed to company index page.	Successful
2	Check the interface link between the company index page and logout module	When present on the company index page, If the user of company wants to logout click on logout	To be directed to the home page of the placement website.	Successful
3	Check the interface link	After clicking on add or edit button	After entering the valid details of company, a	Successful

	when the company wants to add details of company.	the details to add or edit is shown in the page	message arrives informing that he/she has successfully added or edited.	
4	Check whether if student want to update the date, who already entered.	When a student want to update profile then student should click update button..	After updating profile, message shows like data added successfully.	Successful

Fig 6.2 shows the integration testing

## System Testing

System testing tests a completely integrated system to verify that the system meets its requirements. For example, a system test might involve testing a logon interface, then creating and editing an entry, plus sending or printing results, followed by summary processing or deletion (or archiving) of entries, then logoff.

Table 3: test cases for system testing

Sl.no	Test case	Expected Result	Test Result
1.	On clicking company	Renders the page to company index page	successful
2.	On clicking library information centre	Renders the page to college website	successful
3.	On clicking student	Renders the page to student index page	successful
4	On clicking about	Renders the page to college website	successful

Fig 6.3 shows the system testing

## CHAPTER 7

# RESULTS AND DISCUSSIONS

### 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.

### Home page:

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.

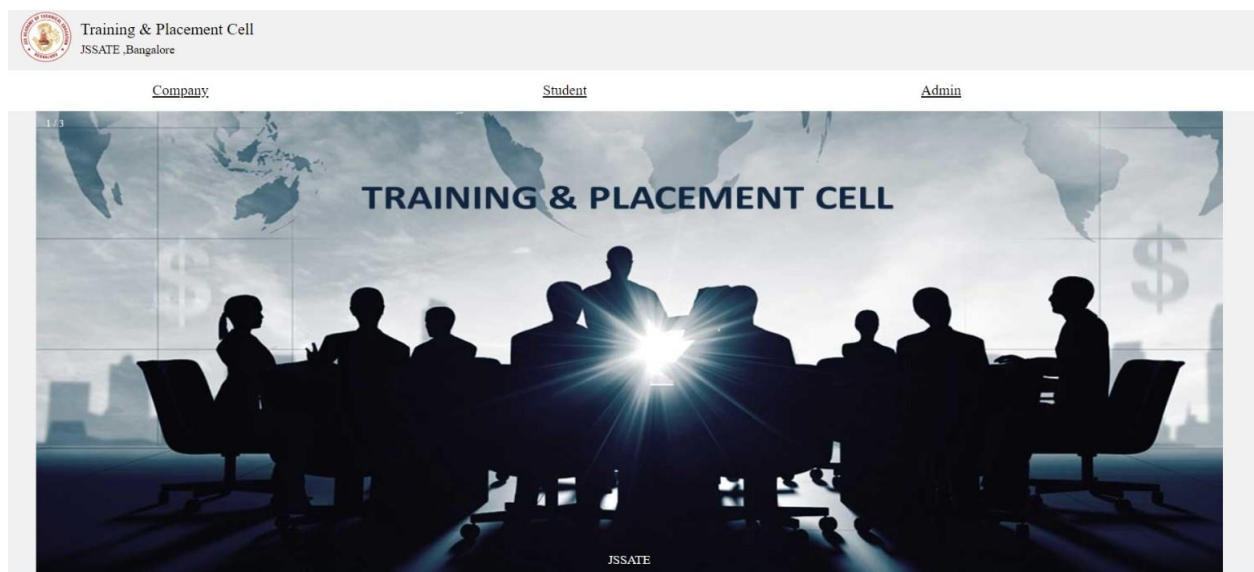


Fig 7.1 Home page

### Company Category:

The company category has two options-

- ☐ Company Register.
- ☐ Company Login.

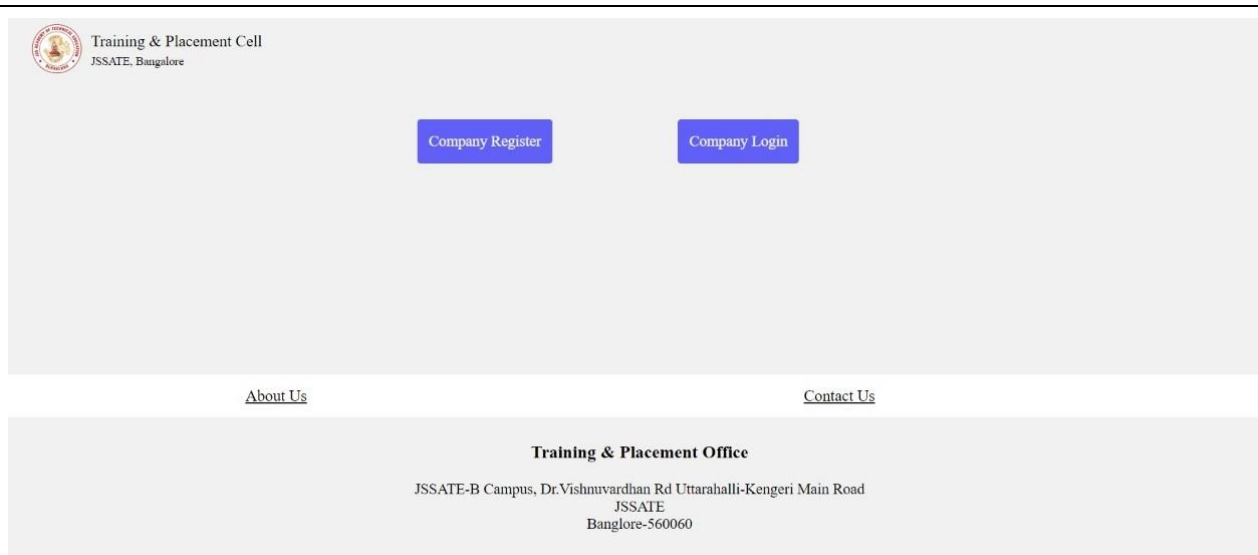


Fig 7.2 company page

The screenshot shows the 'Company Registration' form. The form has a green header with the title 'Company Registration'. Below the header, there are several input fields for registration details: 'Company Id', 'Company Name', 'Password', 'Confirm Password', 'Website', 'Address', and 'Visiting Date'. The 'Visiting Date' field includes a date picker icon. At the bottom of the form, there is a green 'Sign up' button.

Fig 7.3 company registration



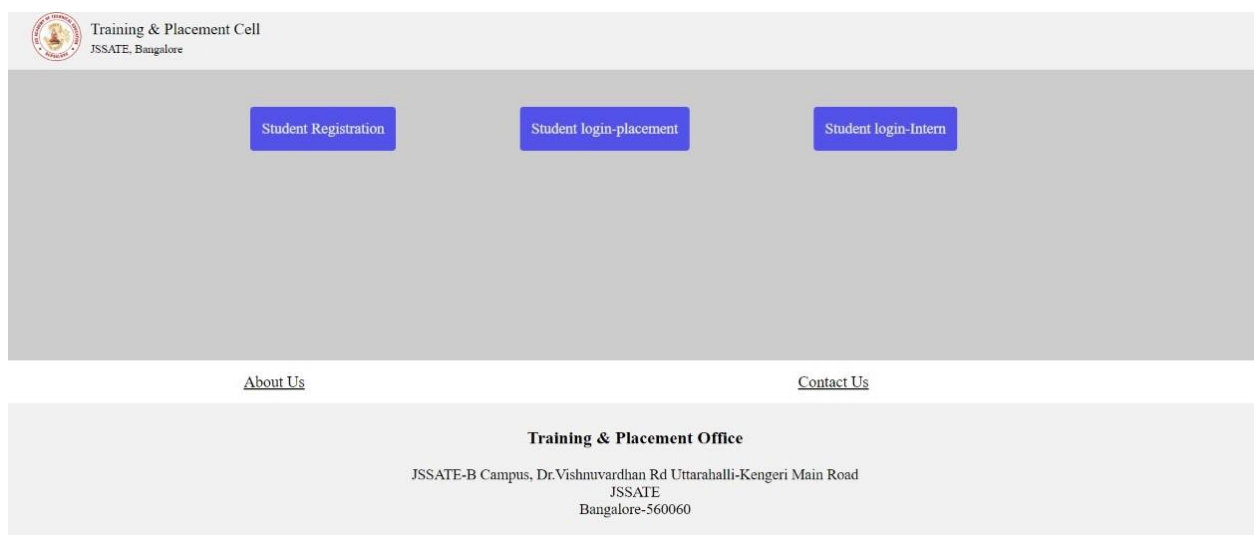
The image shows a 'Company Login' form with a green header. It contains three input fields for 'Company Name', 'Company Id', and 'Password'. Below the fields is a green 'Sign in' button and a link for 'Not enrolled yet? Sign up'.

Fig 7.4 company login

### b) Student Category:

The student category has three options-

- ☐ Student Register.
- ☐ Student login-placement.
- ☐ Student login-internship.



The image shows the 'Student index page' for the Training & Placement Cell of JSSATE, Bangalore. It features a header with the cell's name and logo. Below the header are three blue buttons: 'Student Registration', 'Student login-placement', and 'Student login-Intern'. At the bottom, there are links for 'About Us' and 'Contact Us', and a footer section containing the office name and address.

Fig 7.5 student index page

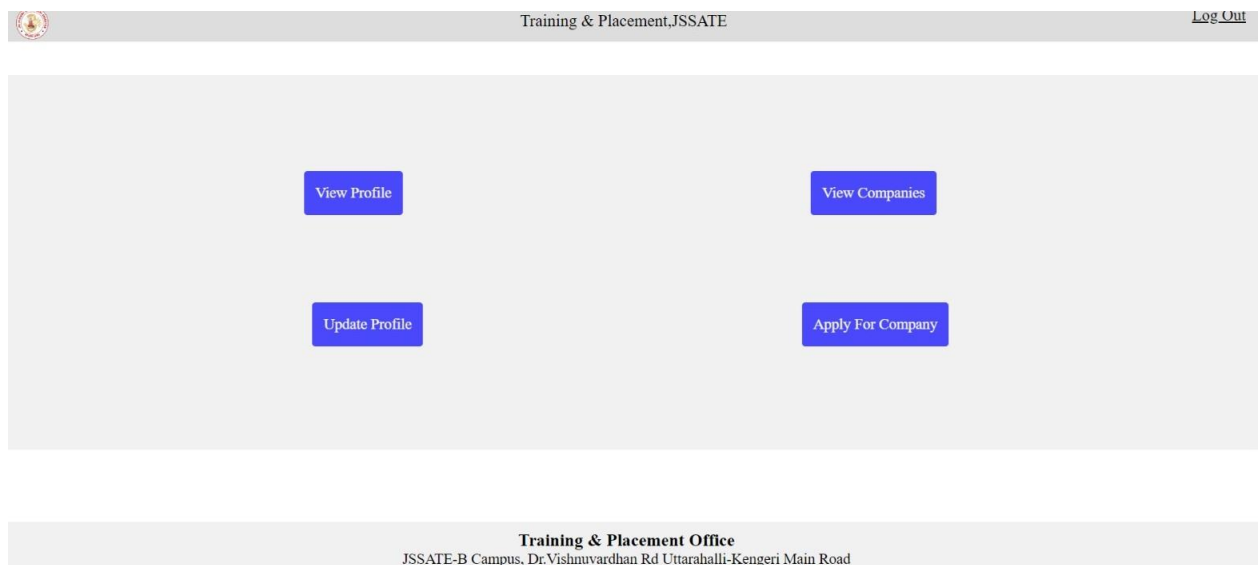
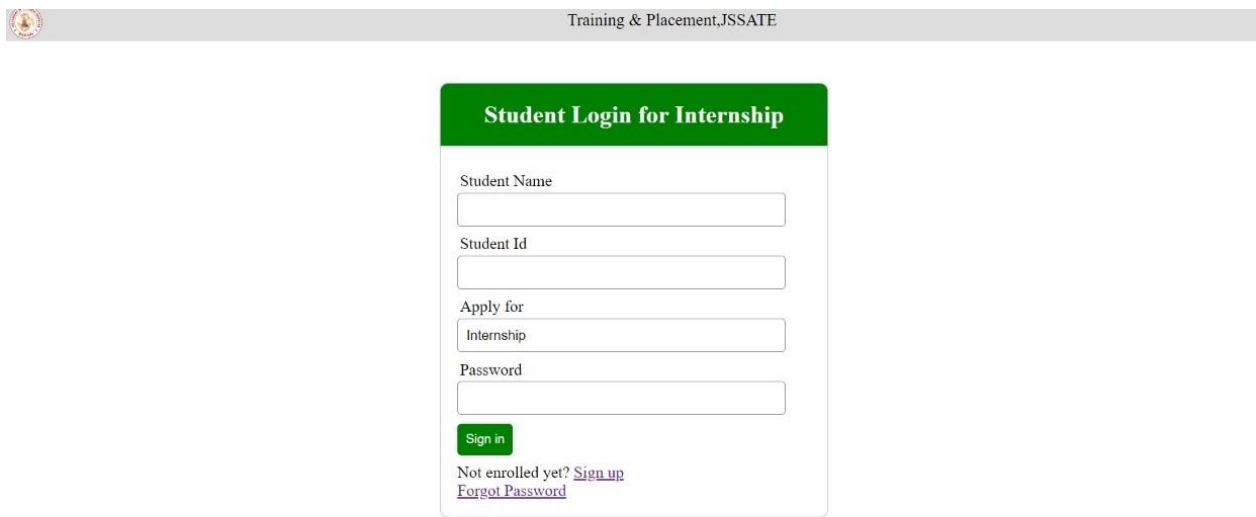


Fig 7.6 view page for after student login

The screenshot displays a login form titled "Student Log in for Placement" in a green header. The form contains the following fields and elements: "Student Name" with a text input field, "Student Id" with a text input field, "Apply for" with a dropdown menu showing "Placement", and "Password" with a text input field. Below these fields is a green "Sign in" button. At the bottom of the form, there is a link "Not enrolled yet? Sign up" and a link "Forgot Password". The entire form is set against a light gray background with a header bar at the top showing a logo and the text "Training & Placement,JSSATE".

Fig 7.7 student login for placement





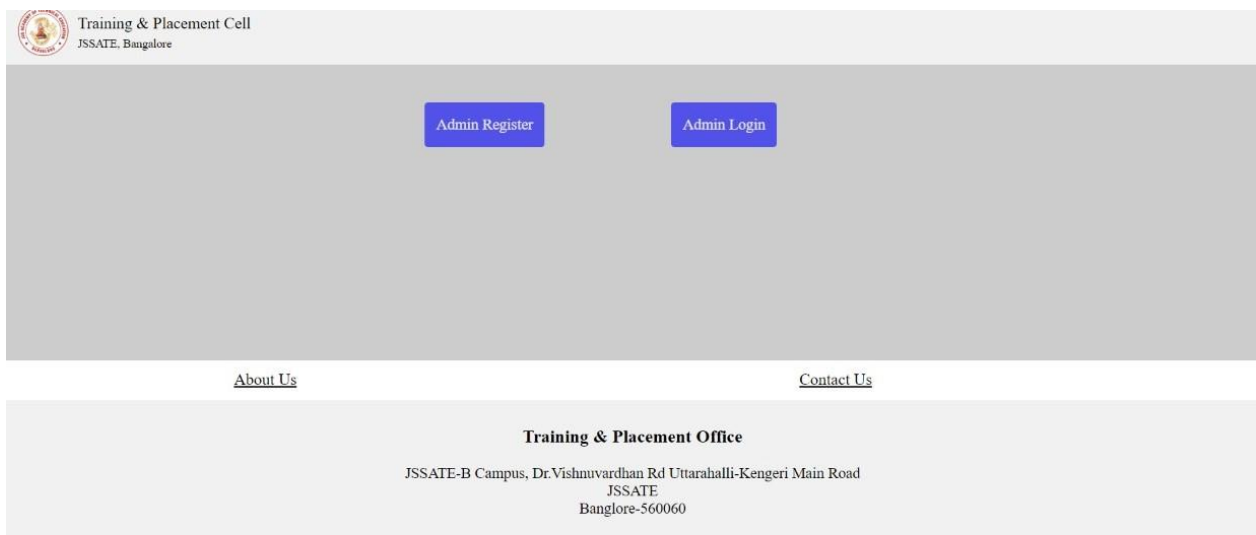
The screenshot shows a web page for the 'Training & Placement, JSSATE'. At the top left is a small circular logo. The main content is a green-bordered box titled 'Student Login for Internship'. Inside this box, there are four input fields: 'Student Name', 'Student Id', 'Apply for' (with 'Internship' selected), and 'Password'. Below these fields is a green 'Sign in' button. At the bottom of the box, there is a link 'Not enrolled yet? [Sign up](#)' and another link '[Forgot Password](#)'.

Fig 7.8 student login for internship

## c) Admin Category:


The admin category shows two option-

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



The screenshot shows the 'Admin' page of the 'Training & Placement Cell, JSSATE, Bangalore'. The page has a light gray background. At the top left is a circular logo. Below the logo, there are two blue buttons: 'Admin Register' and 'Admin Login'. At the bottom of the page, there is a footer section with the text 'Training & Placement Office' and 'JSSATE-B Campus, Dr.Vishnuvardhan Rd Uttarahalli-Kengeri Main Road JSSATE Bangalore-560060'. There are also two links: '[About Us](#)' and '[Contact Us](#)'.

Fig 7.9 admin page

 Training & Placement,JSSATE

### Admin Registration

Admin Id

Admin Name

Email


Post

Contact Number

Date of birth  
 ☐

Qualification

Fig 7.10 admin registration page

 Training & Placement,JSSATE

### Admin Login

Admin Name


Admin Id

Password

Not enrolled yet? [Sign up](#)

Fig 7.11 admin login page

Student Profile



Student Name : susha

Student Id : 100

Status : -

D.O.B : 2000-01-03

Gender : Female

Email : sushiga2000@gmail.com

Address: Hibi

Contact Number : 9818083254

Branch: CSE

10th % : 96

10th pass year : 2013

12th % : 85

12th pass year : 2015

CGPA : 8.7

Final Year : 2019

Backlogs : 0

Apply For : Internship

Fig 7.12 view profile

### Update Profile

Student name  
renuka

Date of birth  
05-05-2000

Email  
rashmipai2000@gmail.com

Address  
Hubli

Contact Number  
8618015254

10th percentage  
96

12th percentage  
88

CGPA  
8.7

Current Password

New Password

[Update](#)

Updated the details [Go Back](#)

Fig 7.13 view page of update profile

## CHAPTER 8

### CONCLUSION AND FUTURE ENHANCEMENTS

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.

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

## CHAPTER 9

## REFERENCES

### BOOK REFERENCES

- 1.Database systems Models, Languages, Design and Application Programming, Ramez Elmasri and Shamkant B. Navathe, 7th Edition, 2017, Pearson.
- 2.An Introduction to Database Systems” by Bipin Desai.
- 3.Database System Concepts” by Abraham Silberschatz and S Sudarshan.
- 4.“Database Management Systems” by Raghu Ramakrishnan.

### WEB REFERENCES

- 1.W3schools
- 2.<https://stackoverflow.com/questions>
- 3.[www.tutorialspoint.com](http://www.tutorialspoint.com)
- 4.[www.javatpoint.com](http://www.javatpoint.com)
- 5.<https://www.php.net/docs.php>
- 6.<https://dev.mysql.com/doc/>