

1. INTRODUCTION

The "Web Application for Campus Placement" is a comprehensive solution designed to make the placement process smoother within the college's Training and Placement Cell. This institution-wide platform offers secure logins for Training and Placement officers, students, and recruiters, enabling easy management of crucial information pertaining to placements and career opportunities.

Through this web application, students can conveniently upload their CVs and personal details, while recruiters can post job openings and manage application processes with ease. Real-time updates on selection statuses ensure that students remain informed about their application progress and available opportunities. Additionally, staff members benefit from features such as job posting management, student registration tracking, and monitoring of the selection process, thereby enhancing overall administrative efficiency.

1.1 MOTIVATION

Motivated by the necessity to enhance the efficiency and transparency of the campus placement procedures, the project aims to address the challenges often encountered in traditional placement systems. By providing a centralized platform accessible to all stakeholders involved, including students, recruiters, and placement coordinator, to create a more hassle-free and organized placement experience.

1.2 NEED

The need for such a system arises from the increasing complexity of modern placement processes and the growing demand for straightforward communication and information management. Traditional methods often result in inefficiencies, such as manual tracking of student applications, difficulty in disseminating information about upcoming placements, and limited accessibility to job opportunities for students.

2. SYSTEM STUDY

System study refers to the comprehensive analysis and evaluation of an existing system or process to identify its strengths, weaknesses, and areas for improvement. It involves gathering information about the current system's functionality, user requirements inform the design and development of a proposed solution.

2.1 EXISTING SYSTEM

The existing system relies on manual processes for handling student and recruiter information, often utilizing spreadsheets or basic databases. This manual approach extends to various aspects, including job postings, event updates, and application processes, which lack automation. Consequently, there is limited automation in these critical areas, leading to inefficiencies and delays. Additionally, the system's design fosters minimal interaction and engagement between students and recruiters, hindering effective communication and collaboration. As a result, there is a significant potential for inefficiencies and delays in communication and updates, impacting the overall effectiveness of the placement process.

Disadvantages:

- The reliance on manual processes for managing student and recruiter information using spreadsheets or basic databases can lead to errors, inconsistencies, and data duplication.
- This manual approach increases the likelihood of data entry mistakes and makes it challenging to maintain data integrity.
- The minimal interaction and engagement between students and recruiters hinder effective communication and collaboration.

2.2 PROPOSED SYSTEM

The proposed system introduces several key features to enhance the campus placement process. Firstly, it offers an online application process for students, facilitating easy submission of CVs and management of profiles. This streamlined approach simplifies the application process and improves accessibility for students. Secondly, the system provides real-time updates and communication channels for recruiters, enabling them to post job openings and update selection lists promptly. This enhances efficiency and ensures timely communication with students. Lastly, the system incorporates secure and role-based access control, allowing different user categories, such as students, recruiters, and staff, to access relevant functionalities based on their roles. This ensures data security and privacy while providing tailored access to the system's features, enhancing overall usability and control.

Advantages:

- Streamlined processes through automation, reducing manual effort and errors.
- Enhanced user experience with a user-friendly interface and improved functionalities.
- Better communication and engagement between students and recruiters.
- Data-driven insights for administrators to track placements, student activities, etc.

3. REQUIREMENT ANALYSIS

Requirements analysis, also known as requirements engineering, is the process of figuring out what users want in a new or updated product. These desired features, called requirements, need to be clear, relevant, and detailed. In software, these requirements are often called functional specifications. It's a team effort that requires expertise in hardware, software, and understanding people.

3.1 HARDWARE REQUIREMENT

The basic needs for any software or operating system are the physical resources of a computer, like the processor and memory. For this project, we need an Intel i5 10th Gen processor, at least an 5GB hard disk, and a minimum of 2GB RAM. The software also relies on peripherals like a keyboard, monitor, and mouse, which have specific requirements such as a 15-inch VGA color monitor, a ball or optical mouse, and a keyboard with at least 102 keys.

3.2 SOFTWARE REQUIREMENTS

Software requirements involve specifying what software needs to be installed on a computer for the application to work properly. These requirements aren't usually included with the software installation and must be installed separately beforehand.

3.2.1 OPERATING SYSTEM

Microsoft's Windows is a graphical operating system designed to facilitate various tasks such as file management, software execution, gaming, video playback, and internet connectivity. It caters to both personal and professional computing needs.

3.2.2 PHP

PHP, short for PHP: Hypertext Preprocessor, stands as a prevalent open-source scripting language tailored for web development, seamlessly integrable into HTML.

3.2.3 HTML

HTML, or Hyper Text Markup Language, serves as the fundamental markup language for crafting web documents for browser display, defining content and structure. It often collaborates with technologies like Cascading Style Sheets and JavaScript for enhanced functionality.

3.2.4 CSS

CSS, which stands for Cascading Style Sheets, is a language for defining the visual appearance of documents authored in markup languages like HTML or XML. It plays a crucial role on the web, complementing HTML and JavaScript in enhancing the user experience.

3.2.5 BOOTSTRAP FRAMEWORK

Bootstrap, a robust front-end framework, empowers developers in crafting contemporary websites and web applications. It offers a plethora of HTML and CSS templates for interface elements like buttons and forms, along with support for JavaScript extensions.

3.2.6 MICRO SOFT VISUAL STUDIO CODE

Microsoft Visual Studio Code is a sleek code editor equipped with features like debugging, task management, and version control. It's tailored to streamline the code development process, providing essential tools for swift iteration cycles.

4. SYSTEM DESIGN

Designing software involves identifying different parts of the program and understanding how they relate to each other. This includes figuring out the structure of the software and creating a plan for how it will be documented. When we talk about modularity, we mean breaking the system into smaller pieces so that they interact with each other as little as possible and their connections are clearly defined.

Systems design is about figuring out all the pieces that need to be in a system, like its modules, architecture, and components, as well as how they connect and what data they use. This is done based on what the system needs to do, whether for a business or any other organization.

4.1 LOGICAL DESIGN

Logical design is like coming up with a blueprint for the system, but without getting into the nitty-gritty of how it will actually be built. It's all about mapping out the information the system will use, like what data comes in and goes out, and what processes it needs to support. This involves organizing the data into logical groups called entities and attributes, which are like the building blocks of the system. Data flow diagrams and entity-relationship diagrams are tools used for this part of the design process.

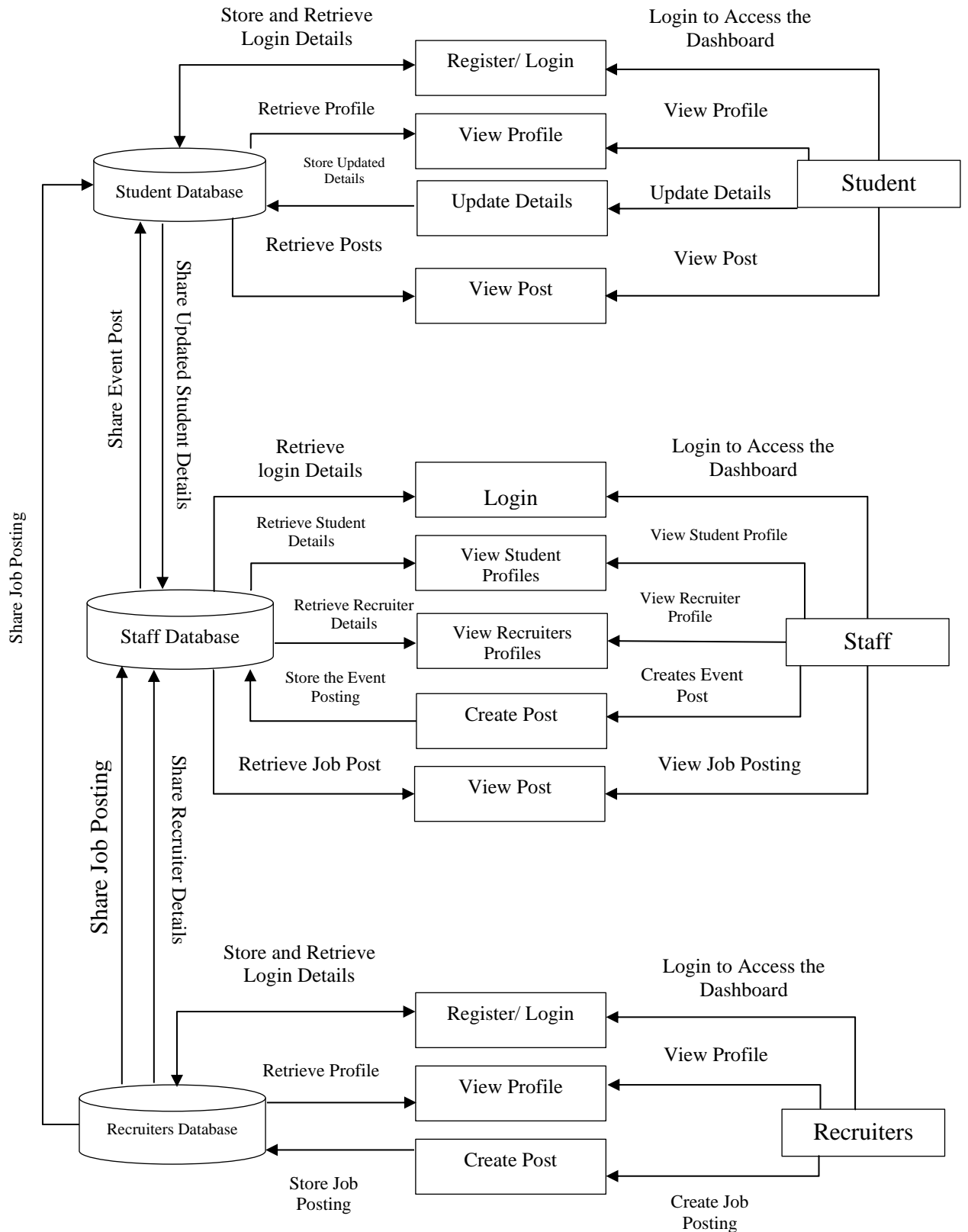


Fig 4.1 Logical Design of Web Application for Campus Placement

4.2 INPUT DESIGN

Input Design is like translating what users want into language computers understand. It's crucial for making sure data is entered correctly and helps guide management in using the computer system effectively.

4.2.1 INPUT DESIGN FOR REGISTRATION

The following input design represents what are inputs are taken from the users while they creating their profile. These data are collected through the register form and stored in the database.

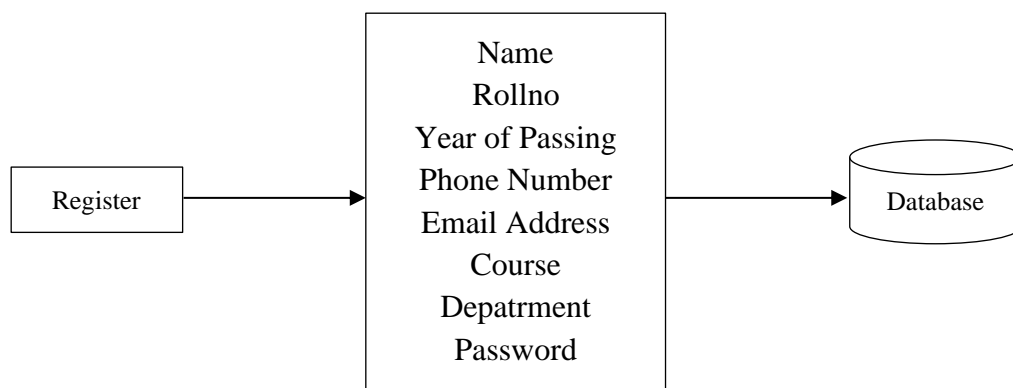


Fig 4.2 Input Design for Student Registration

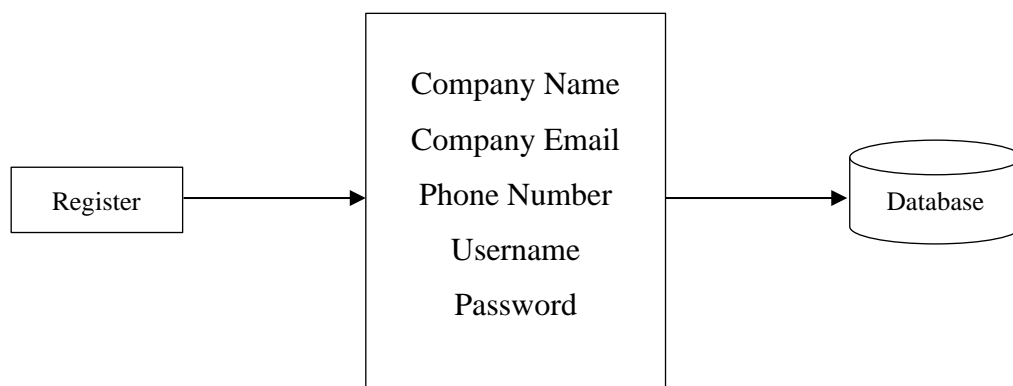


Fig 4.3 Input Design for Recruiter Registration

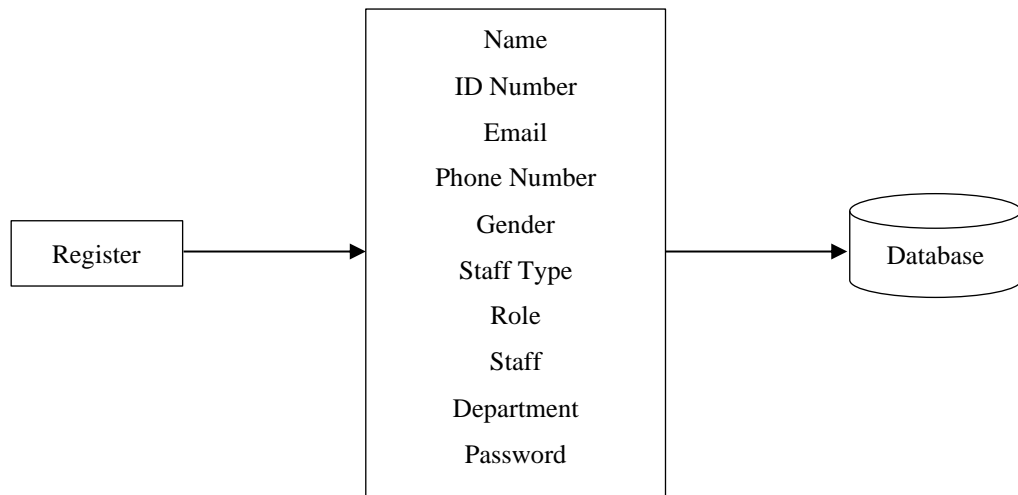


Fig 4.4 Input Design for Staff Registration

4.2.2 INPUT DESIGN FOR UPLOADING POSTING

The following input design represents what are the inputs are taken from the user when they create new post. These data collected from the users through create post form and stored in the database.

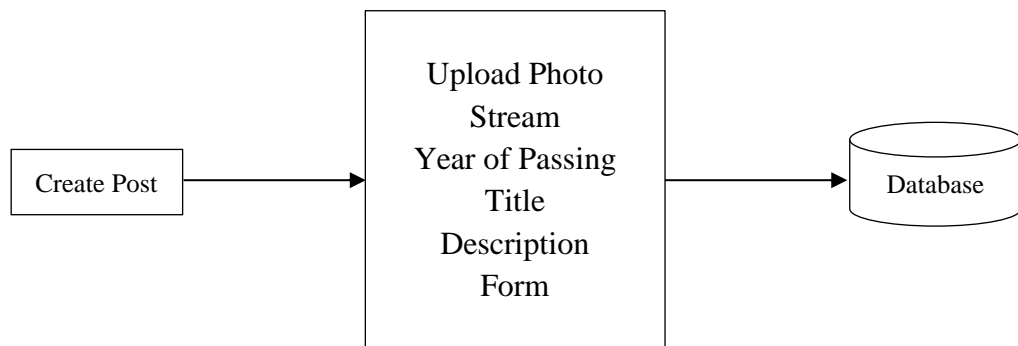


Fig 4.5 Input Design for Recruiter Create Post

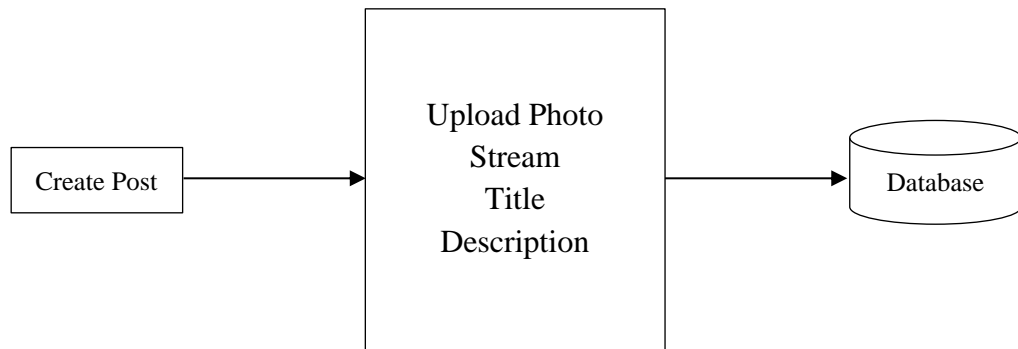


Fig 4.6 Input Design for Staff Create Post

4.2.3 INPUT DESIGN FOR JOB APPLICATION

The following input design represents what are the inputs are taken from the user when they recruiter activate the job application form. These data collected from the student through job application form and stored in the database.

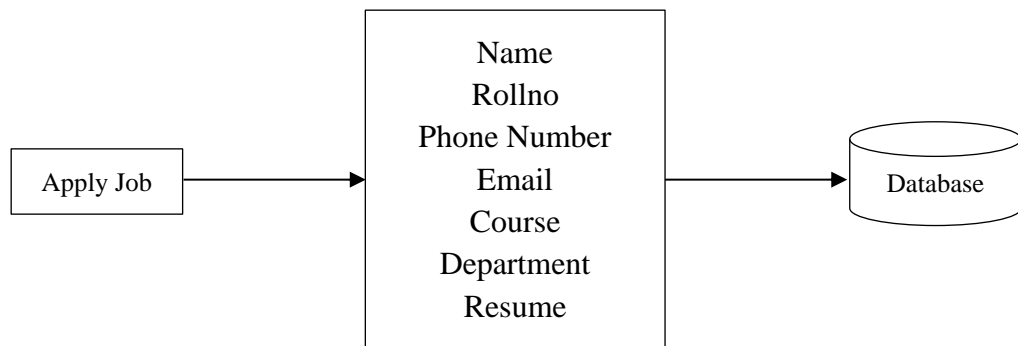


Fig 4.7 Input Design for Job Application

4.3 OUTPUT DESIGN

Output Design is all about figuring out what kind of information the system needs to show and how it should look. It involves planning what reports and visuals are needed and making sure they're easy to understand and control.

4.3.1 OUTPUT DESIGN FOR VIEWING ACCOUNT

The following output design represents what are the data are fetched from database to view profile details of the users.

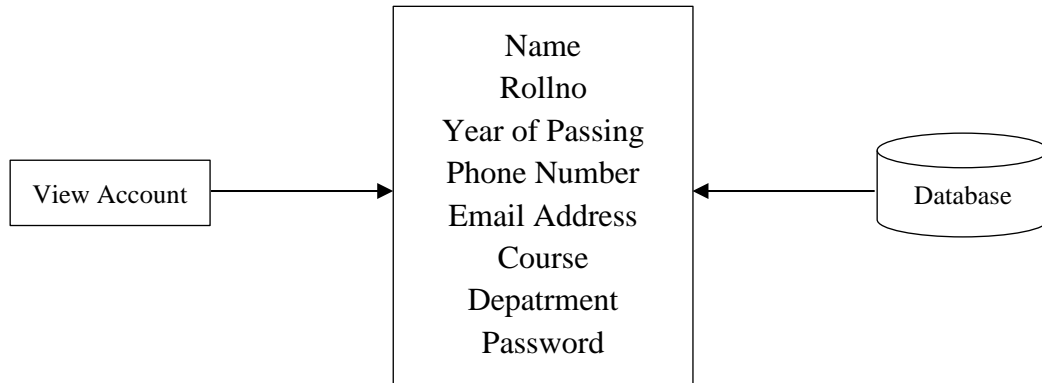


Fig 4.8 Output Design for View Student Account

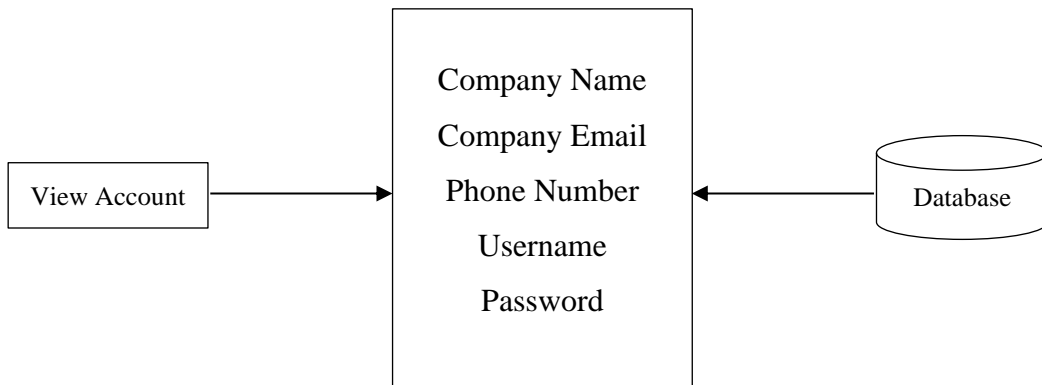


Fig 4.9 Output Design for View Recruiter Account

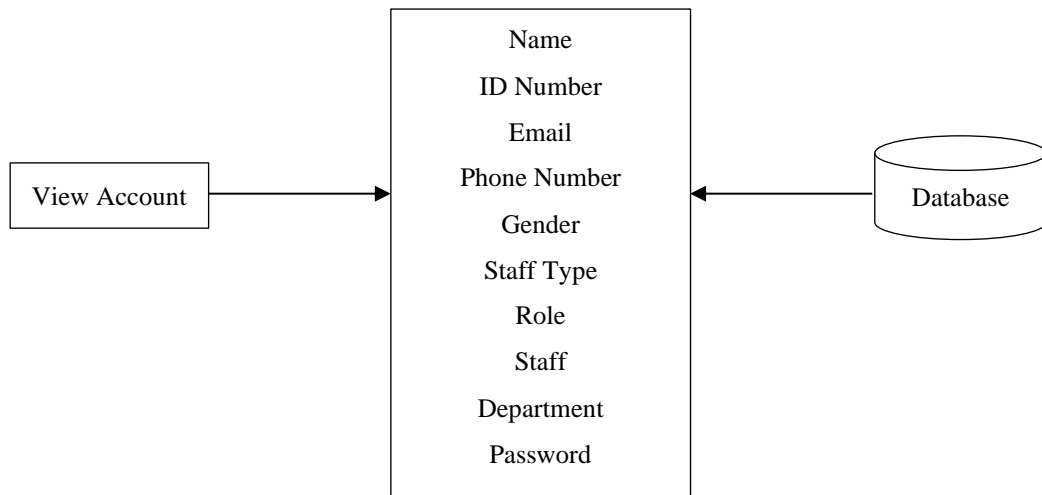


Fig 4.10 Output Design for View Staff Account

4.3.2 OUTPUT DESIGN FOR VIEW POST

The following output design represents what are the data are fetched from the database to view a job post.

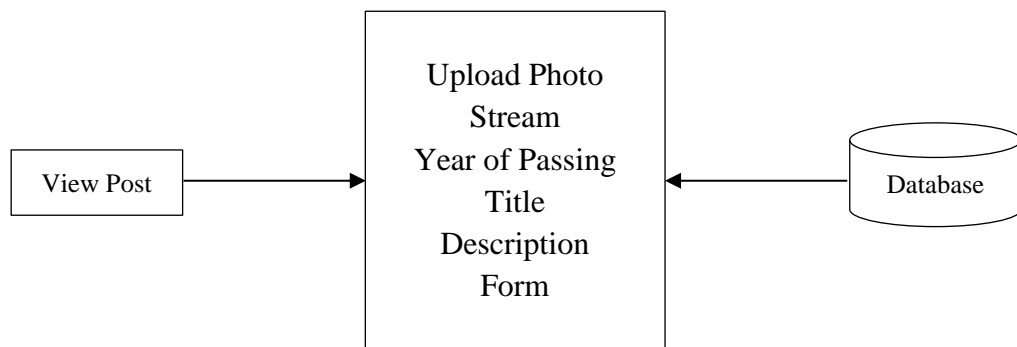


Fig 4.11 Output Design for Recruiter Post

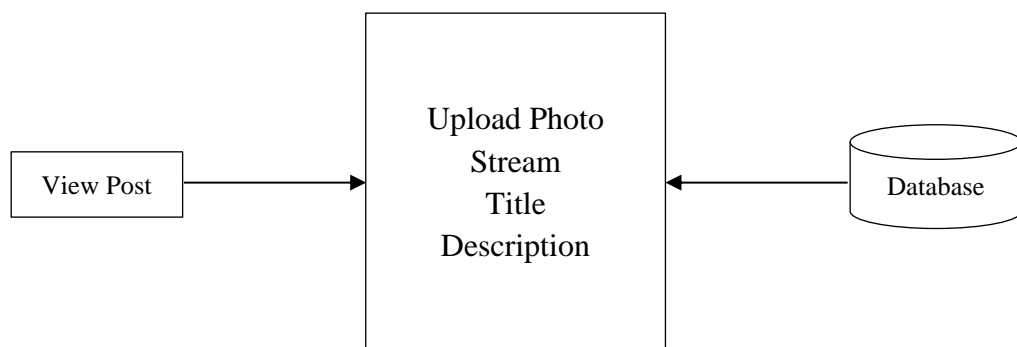


Fig 4.12 Output Design for Staff Post

4.3.3 Output Design for Job Application

The following output design represents what are the data are fetched from database to view a job application.

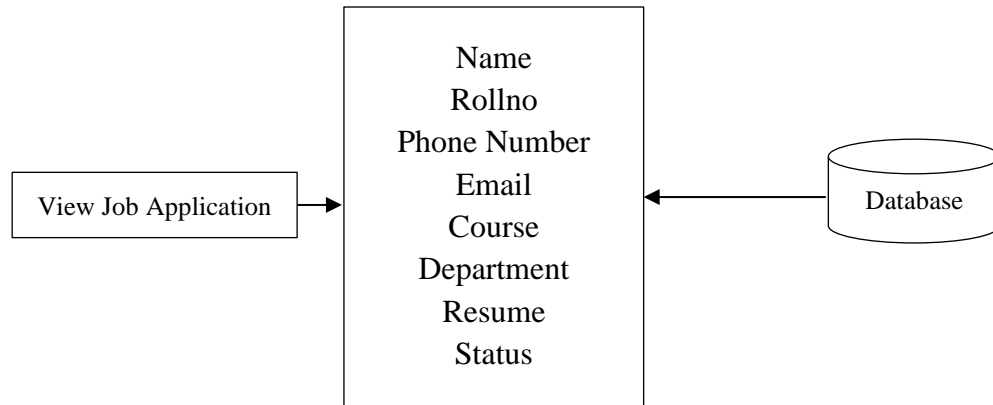


Fig 4.13 Output Design for Job Application

4.4 DATABASE DESIGN

Database design is basically about creating and organizing a system to manage data effectively. By doing this well, you save money on maintenance, ensure data is consistent, and use storage space efficiently. To design a good database, you need to understand what data you have, how it's related, and what rules it needs to follow. The goal is to create both a logical model (focused on data requirements) and a physical model (how data is stored) that work well together. Think of it like planning a house: you need to figure out what rooms you need and how they'll be arranged before you start building.

4.4.1 DATAFLOW DIAGRAM

A Data Flow Diagram (DFD) serves as a classic visual representation of how information moves within a system. A well-organized DFD can effectively illustrate the system's requirements graphically, whether the system is manual, automated, or a mix of both. It outlines how data enters and exits the system, identifies points of information alteration, and indicates data storage locations.

The primary goal of a DFD is to delineate the scope and limitations of the entire system. It functions as a communication tool between system analysts and stakeholders, providing a starting point for system redesign. Sometimes referred to as a data flow

graph or bubble chart, a DFD illustrates the sequential transformations data undergoes within the system, leading to the generation of output.

In complex systems, data transformation typically occurs through multiple steps before resulting in output. The DFD aims to capture these transformations, depicting how input data evolves through various processes (or bubbles) to generate output data.

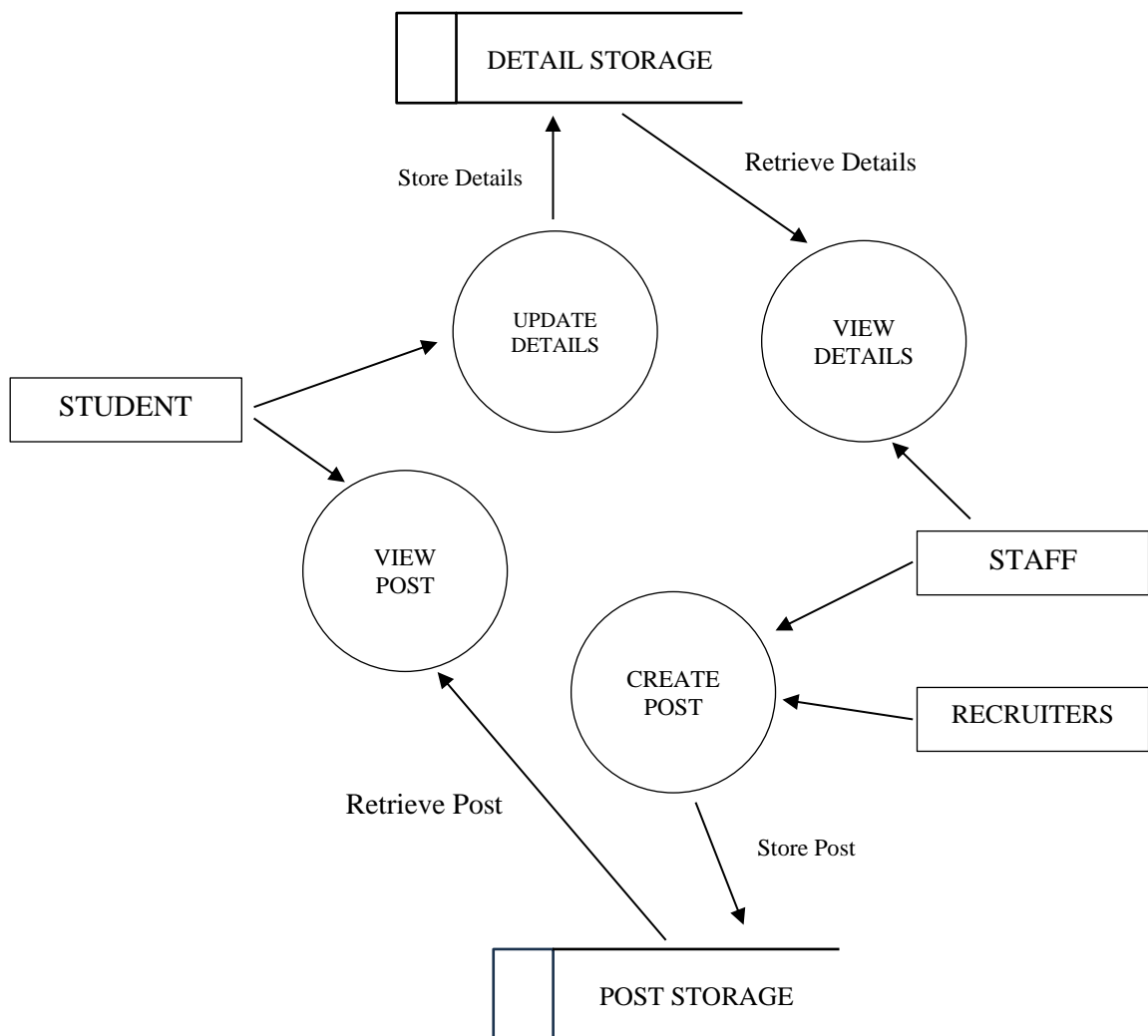


Fig 4.14 Dataflow diagram for Web Application for Campus Placement

4.4.2 TABLE DESIGN

A widely used data structure is a database table, comprising rows and columns, akin to a two-dimensional array. In this structure, each value is associated with a particular index, resembling a list of values within the array.

4.4.2.1 Student Registration Table

The student registration table is used for storing student details such as Name, Rollno, Year of Passing, Mobile number, Email address, Course, Department and Password. Field Id is a primary key used to identify each student.

Table 4.1 Student Registration Table

Field name	Data Type	Null	Key
name	Varchar (30)	No	
rollno	Int (10)	No	PRIMARY_KEY
yop	Int (10)	No	
phone	Int (10)	No	
email	Varchar (30)	No	
course	Varchar (30)	No	
depatrment	Varchar (30)	No	
password	Varchar (30)	No	

4.4.2.2 Recruiter Registration Table

The recruiter registration table is used for storing recruiter details such as Company Name, Company Email, Phone number, Username and Password. Field Id is a primary key used to identify each recruiter.

Table 4.2 Recruiter Registration Table

Field name	Data Type	Null	Key
userid	Int (10)	No	PRIMARY_KEY
cname	Varchar (30)	No	
cemail	Varchar (30)	No	
cphone	Int (10)	No	
username	Varchar (30)	No	
password	Varchar (30)	No	

4.4.2.3 Staff Registration Table

The staff registration table is used for storing staff details such as Name, IDnum, Email, Phone Number, Gender, Staff Type, Role, Staff, Department and Password. Field Id is a primary key used to identify each recruiter.

Table 4.3 Staff Registration Table

Field name	Data Type	Null	Key
userid	Int (10)	No	PRIMARY_KEY
name	Varchar (30)	No	
idnum	Int (10)	No	
email	Varchar (30)	No	
phone	Int (10)	No	
gender	Varchar (30)	No	
stype	Varchar (30)	No	
role	Varchar (30)	No	
staff	Varchar (30)	No	
department	Varchar (30)	No	
password	Varchar (30)	No	

4.4.2.4 Recruiter Post Table

The recruiter posts table is used for storing data of posts details such as User ID of the Recruiter, Title, Stream, Year of Passing, Description, Form, Photo_Path and Posted date. In this table post_id is a primary key to identify each post.

Table 4.4 Recruiter Post Table

Field name	Data Type	Null	Key
userid	Int (10)	No	
postid	Int (30)	No	PRIMARY_KEY
username	Varchar (30)	No	
stream	Varchar (30)	No	
title	Varchar (30)	No	
description	Varchar (30)	No	
form	Varchar (30)	No	
photo_path	Varchar (255)	No	

4.4.2.5 Staff Post Table

The staff posts table is used for storing data of posts details such as User ID of the Staff, Title, Stream, Description, Photo_Path and Posted date. In this table post_id is a primary key to identify each post.

Table 4.5 Staff Post Table

Field name	Data Type	Null	Key
idnum	Int (10)	No	
postid	Int (30)	No	PRIMARY_KEY
stream	Varchar (30)	No	
title	Varchar (30)	No	
description	Varchar (30)	No	
photo_path	Varchar (255)	No	

4.4.2.6 Job Application Table

The job application table is used to get the data from students such as Name, Rollno, Phone Number, Email, Course, Department, Percentage and Resume. In this table applid is a primary key to identify each job application.

Table 4.6 Job Application Table

Field name	Data Type	Null	Key
applid	Int (10)	No	PRIMARY_KEY
name	Varchar (30)	No	
rollno	Int (10)	No	
phone	Int (10)	No	
email	Varchar (30)	No	
course	Varchar (30)	No	
depatrment	Varchar (30)	No	
per	Int (5)	No	
file_path	Varchar (255)	No	

4.4.3 ER DIAGRAM

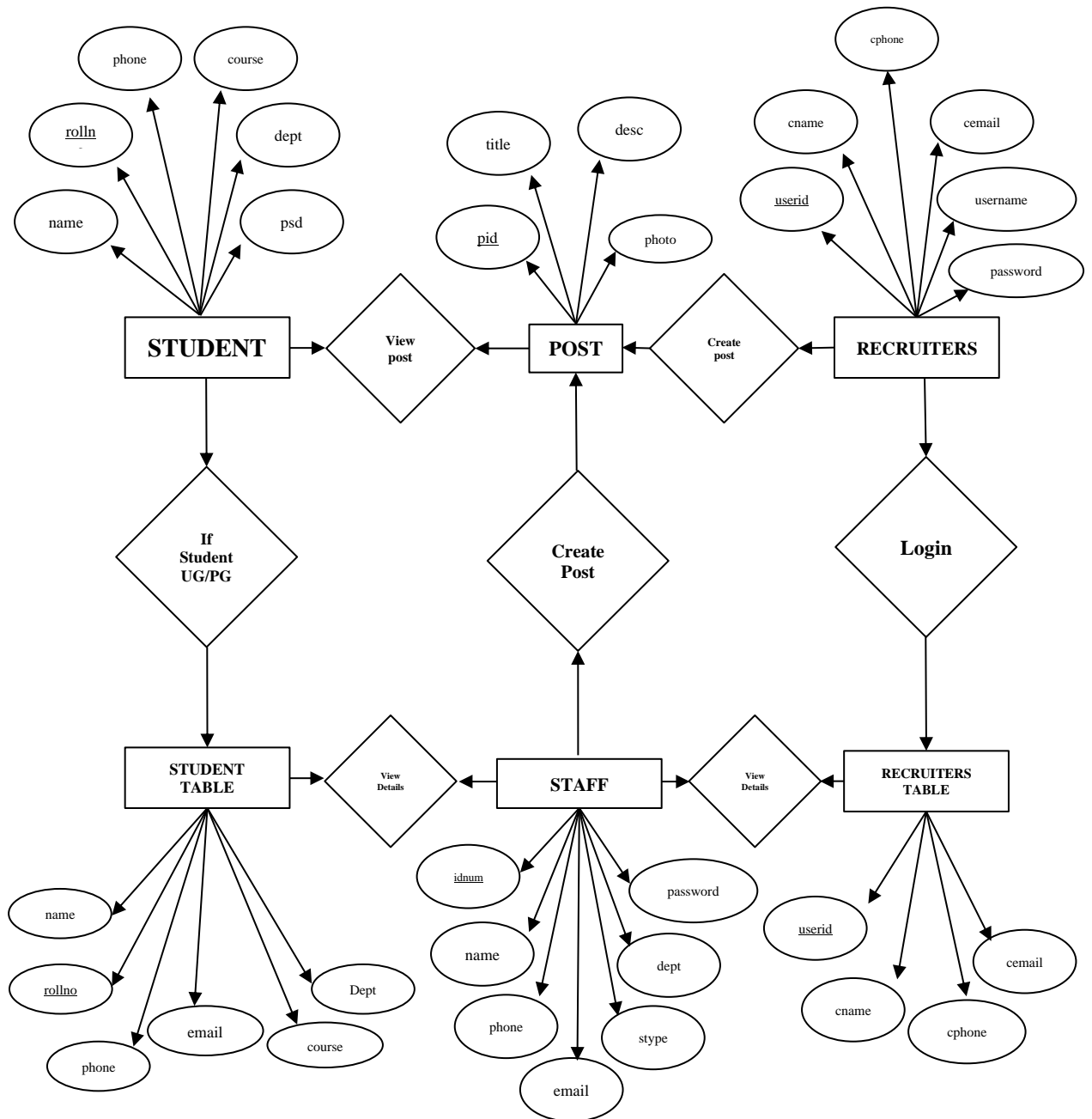


Fig 4.15 ER Diagram for Web Application for Campus Placement

4.5 PROGRAM DESIGN

Program design is the process of conceptualizing, planning, and structuring a software program to meet specific requirements or solve particular problems effectively.

4.5.1 MODULE DESCRIPTION

The "Web Application for Campus Placement" comprises several interconnected modules, each serving a specific function within the overall system. These modules work together seamlessly to facilitate the efficient management of campus placements.

1. Student
2. Recruiter
3. Staff

4.5.1.1 Student

- The student module enables students to create and manage their profiles.
- Students can upload their resumes/CVs, update personal information, and specify their career preferences.
- They can view and apply to job, posted by recruiters.

4.5.1.2 Recruiter

- Recruiters have access to this module to post job openings, manage applications, and communicate with students.
- They can create detailed job postings, specify job requirements, and track application statuses.
- Recruiters can also view student profiles and shortlist candidates for interviews.

4.5.1.3 Staff

- This module it's mostly dedicated to placement officers responsible for overseeing the placement process.
- Placement officers can manage job postings, monitor student applications, and coordinate communication between students and recruiters.

5. SYSTEM DEVELOPMENT

Systems development involves the systematic process of conceptualizing, designing, testing, and implementing new software applications or programs. This encompasses activities such as internally developing customized systems, creating database systems, or procuring third-party software solutions. To ensure smooth functioning, written standards and procedures are crucial for guiding all information systems processing functions. The management of an organization plays a pivotal role in defining and implementing these standards, as well as adopting an appropriate system development life cycle methodology to govern the entire process of developing, acquiring, implementing, and maintaining computerized information systems and associated technology.

5.1 PROGRAM DEVELOPMENT

Program Development constitutes a methodical approach to creating high-quality software. It offers a structured framework for breaking down the complexities of program development into manageable phases, ensuring each step is successfully completed before progressing to the next. The initial phase involves defining the problem, often led by system analysts who deliver program specifications to programmers. Following the identification and removal of syntax errors, the program undergoes execution. However, the output may still be erroneous due to logical errors in the program's design, which are mistakes made during problem-solving.

Coding entails translating algorithms into a specific programming language, with structured programming emphasizing the use of well-defined control structures. Adherence to language rules is imperative, as any violation can lead to errors that must be rectified before advancing to subsequent stages. During deployment, the program is installed at the user's site and monitored until user approval is obtained. Even after completion, software requires regular maintenance and evaluation, involving tasks such as error correction and software updates performed by the programming team.

5.1.1 SOURCE CODE

5.1.1.1 Login.html

```
<html lang="en" data-bs-theme="">
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="shortcut icon" href="/Img/Career.png" type="image/png">
<link rel="stylesheet" href="/Assets/css/style.css">
<link rel="stylesheet" href="/Assets/css/bootstrap.min.css">
<title>Training and Placement</title>
</head>
<body>
<section id="top">
<!-- Back to Top Button -->
</section>
<!-- nav -->
<header>
<nav class="navbar navbar-expand-lg border-bottom shadow">
<div class="container text-center">
<a class="navbar-brand justify-content-start" href="index.html">Training and
Placement</a>
<button class="navbar-toggler" type="button" data-bs-toggle="collapse"
data-bs-target="#navbarNavDropdown" data-bs-trigger="focus" aria-
controls="navbarNavDropdown"
aria-expanded="false" aria-label="Toggle navigation">
<span class="navbar-toggler-icon"></span>
</button>
<div class="collapse navbar-collapse justify-content-end"
id="navbarNavDropdown">
<ul class="navbar-nav">
<li class="nav-item">
<a class="nav-link text-dark" aria-current="page" href="index.html">Home</a>
</li>
<li class="nav-item dropdown">
<a class="nav-link dropdown-toggle text-dark" href="#" role="button"
data-bs-toggle="dropdown" aria-expanded="false">
Login
</a>
<ul class="dropdown-menu" style="font-size: large;">
<li><a class="dropdown-item" href="stafflogin.html">Staff</a></li>
<li><a class="dropdown-item" href="recruiterlogin.html">Recruiter</a></li>
<li><a class="dropdown-item" href="studentlogin.html">Students</a></li>
</ul>
</li>
</ul>
</div>
</div>
</nav>
</header>
<!-- nav -->
```

```

<!-- Form design -->
<section id="stlogin">
<div class="container py-5 h-100 ">
<div class="row justify-content-center align-items-center h-100">
<div class="col-12 col-lg-9 col-xl-7">
<div class="card shadow-2-strong card-registration" style="border-radius: 15px;">
<div class="card-body p-4 p-md-5">
<h3 class="mb-4 pb-2 pb-md-0 mb-md-5">Student Login</h3>
<form method="post" action="studentlogin.php">
<div class="form-floating mb-3">
<input type="text" class="form-control" name="rollno" id="floatingPassword"
placeholder="Roll Number" required>
<label for="floatingPassword">Roll Number</label>
</div>
<div class="form-floating mb-3">
<input type="password" name="password" class="form-control"
id="floatingPassword"
placeholder="Password">
<label for="floatingPassword">Password</label>
</div>
<div class="row mb-3">
<div class="col">
<h6 class="text-start">Don't have an account yet? <a href="studentregister.html"
class="text-danger" style="text-decoration: none;">Sign Up</a></h6>
</div>
</div>
<div class="d-flex gap-3">
<button type="submit" class="btn btn-dark w-50">Login</button>
<button type="reset" class="btn btn-dark w-50">Clear</button>
</div>
</form>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
<!-- Form design -->
<!-- footer -->
<section id="footer">
<div class="container-fluid px-5 bg-light">
<div class="row row-cols-1 row-cols-sm-2 row-cols-md-4 py-5 my-5 text-start">
<div class="col mb-3">
<a href="/" class="d-flex align-items-center mb-3 link-dark text-decoration-none">
<h5 class="ci me-2" width="40" height="32">Training and Placement</h5>
</a>
<p class="text-muted">&copy; 2022</p>
</div>
<div class="col">
<h5>For any Enquiry</h5>

```

```

<ul class="nav flex-column">
<li class="nav-item mb-2"><b>Address</b></li>
<li class="nav-item mb-2">
<p>Bishop Heber College, <br>Tiruchirappalli - 620 017, <br>Tamil Nadu,
India.</p>
</li>
<li class="nav-item mb-2"><b>Phone And Email</b></li>
<li class="nav-item mb-2">
<p>9786088211 <br>heberplacement@bhc.edu.in </p>
</li>
<li class="nav-item mb-4">
<p><b>Working Hours:</b> 9:00 AM to 5:00 PM</p>
</li>
</ul>
</div>
<div class="col">
<h5>For any Enquiry</h5>
<ul class="nav flex-column">
<iframe class="w-100 h-75 mb-4 rounded-2"
src="https://www.google.com/maps/embed?pb=!1m14!1m8!1m3!1d7837.911300304
496!2d78.673077!3d10.814706!3m2!1i1024!2i768!4f13.1!3m3!1m2!1s0x3baaf5766c
36af8f%3A0x6b032561715070a3!2sBishop%20Heber%20College!5e0!3m2!1sen!2s
us!4v1700985869646!5m2!1sen!2sus"
style="border:0;" allowfullscreen="" loading="lazy" referrerpolicy="no-referrer-
when-downgrade"></iframe>
</ul>
</div>
<div class="col">
<h5 class="">Logins</h5>
<ul class="nav flex-column">
<li class="nav-item mb-2"><a href="stafflogin.html" class="nav-link p-0 text-
muted">Staff</a></li>
<li class="nav-item mb-2"><a href="recruiterlogin.html" class="nav-link p-0 text-
muted">Recruiter</a></li>
<li class="nav-item mb-2"><a href="studentlogin.html" class="nav-link p-0 text-
muted">Student</a></li>
</ul>
</div>
</footer>
</div>
</section>
<!-- footer -->
<!-- libraries -->
<script src="./Assets/js/jquery.min.js"></script>
<script src="./Assets/js/bootstrap.min.js"></script>
<script src="./Assets/js/popper.min.js"></script>
<script
src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.11.8/dist/umd/popper.min.js"></
script>

```

```

<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.min.js"></script>
<!-- script -->
<script>
// let triggerElement = document.getElementById("display");
// let popover = new bootstrap.Popover(triggerElement, {
// trigger:'focus',
// });
const popoverTriggerList = document.querySelectorAll("[data-bs-toggle='popover']")
const popoverList = [...popoverTriggerList].map(popoverTriggerEl => new
bootstrap.Popover(popoverTriggerEl, { trigger: 'focus' }))
</script>
</body>
</html>
</body>
</html>

```

5.1.1.2 Studentmain.php

```

<?php
session_start();
require_once 'connectdb.php';
if (!isset($_SESSION['rollno'])) {
header('Location: studentlogin.html');
exit;
}
@$rollno = $_SESSION["rollno"];
$sql = "SELECT * FROM studentlogin where rollno='$rollno'";
$profile = $conn->query($sql);
?>
<!DOCTYPE html>
<html lang="en" data-bs-theme="">
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet" href="./Assets/css/style.css">
<link rel="stylesheet" href="./Assets/css/bootstrap.min.css">
<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-
icons@1.11.3/font/bootstrap-icons.min.css">
<title>Training and Placement</title>
</head>
<script src="https://code.jquery.com/jquery-3.7.1.min.js" integrity="sha256-
/JqT3SQfawRcv/BIHPTthkBvs0OEvtFFmqPF/lyI/Cxo="
crossorigin="anonymous"></script>
<script>
$(function() {
$("#contact").load("contactus.php");
});
</script>
<style>
/* Styles for the full-screen preview */
.preview-container {

```



```

display: none;
position: fixed;
top: 0;
left: 0;
width: 100%;
height: 100%;
background-color: rgba(0, 0, 0, 0.9);
z-index: 999;
overflow: auto;
}
.preview-content {
display: flex;
justify-content: center;
align-items: center;
height: 100%;
}
.preview-image {
max-width: 90%;
max-height: 90%;
margin: auto;
}
</style>
<body>
<?php
while ($row = mysqli_fetch_assoc($profile)) {
?>
<!-- nav -->
<header>
<nav class="navbar navbar-expand-lg border-bottom shadow-lg">
<div class="container text-center">
<a class="navbar-brand justify-content-start" href="index.html" onclick="return
confirm('Do You Want To Logout..?')">Training and Placement</a>
<button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-
target="#navbarNavDropdown" data-bs-trigger="focus" aria-
controls="navbarNavDropdown" aria-expanded="false" aria-label="Toggle
navigation">
<span class="navbar-toggler-icon"></span>
</button>
<div class="collapse navbar-collapse justify-content-end"
id="navbarNavDropdown">
<ul class="navbar-nav">
<li class="nav-item">
<a class="nav-link text-dark" aria-current="page"
href="./studentmainpg.php">Home</a>
</li>
<li class="nav-item">
<a class="nav-link text-dark" href="#post">Notice</a>
</li>
<li class="nav-item">
<a class="nav-link text-dark" href="./studentpgform.php">Forms</a>

```

```

</li>
<li class="nav-item">
<a class="nav-link text-dark navbar-nav" type="button" data-bs-toggle="offcanvas"
data-bs-target="#offcanvasScrolling" aria-controls="offcanvasScrolling"><i class="bi
bi-person-circle"></i><b class="mx-2"><?php echo $row["name"]; ?></b></a>
</li>
<li class="nav-item">
<a class="nav-link text-dark" href="/" onclick="return confirm('Do You Want To
Logout..?')">Logout</a>
</li>
</ul>
</div>
</div>
</nav>
</header>
<!-- nav -->
<!-- Offcanvas Profile -->
<div class="offcanvas offcanvas-start w-100" data-bs-scroll="true" data-bs-
backdrop="false" tabindex="-1" id="offcanvasScrolling" aria-
labelledby="offcanvasScrollingLabel">
<div class="offcanvas-header">
<h5 class="offcanvas-title" id="offcanvasScrollingLabel"></h5>
<button type="button" class="btn-close" data-bs-dismiss="offcanvas" aria-
label="Close"></button>
</div>
<div class="offcanvas-header">
<h1>Profile</h1>
</div>
<div class="offcanvas-body">
<table class="table">
<thead>
<tr>
<th scope="col"> </th>
<th scope="col"> </th>
<th scope="col"> </th>
</tr>
</thead>
<tbody>
<tr>
<th scope="row">
<td>Name</td>
<td><?php echo $row["name"]; ?></td>
</th>
</tr>
<tr>
<th scope="row">
<td>Roll No</td>
<td><?php echo $row["rollno"]; ?></td>
</th>
</tr>

```

```

<tr>
<th scope="row">
<td>Course</td>
<td><?php echo $row["course"]; ?></td>
</th>
</tr>
<tr>
<th scope="row">
<td>Department</td>
<td><?php echo $row["dept"]; ?></td>
</th>
</tr>
<tr>
<th scope="row">
<td>Phone</td>
<td><?php echo $row["phone"]; ?></td>
</th>
</tr>
<tr>
<th scope="row">
<td>Email</td>
<td><?php echo $row["email"]; ?></td>
</th>
</tr>
</tbody>
</table>
<?php
}
?>
<a class="btn btn-secondary w-100" href="/pgprofile.php" role="button">View Full
Details</a>
</div>
</div>
<div class="container mt-5 mb-5">
<div class="d-flex justify-content-center">
<div class="btn-group" role="group" aria-label="Post Type">
<button type="button" class="btn btn-outline-secondary active"
id="recruiterBtn">Recruiters</button>
<button type="button" class="btn btn-outline-secondary" id="staffBtn">Training and
Placement</button>
</div>
</div>
</div>
<?php
// Retrieve job post data along with the respective user profiles
$jobPostQuery = "SELECT rp.*, rf.*, rp.stream AS stream FROM recruiterpost rp
INNER JOIN recruiterform rf ON rp.userid = rf.userid WHERE rp.stream = 'For PG'
OR rp.stream = 'Both UG and PG'";
$result = $conn->query($jobPostQuery);
?>

```

```

<!-- Post -->
<section id="post" style="display: block;">
<div class="container mt-5">
<div class="d-flex flex-wrap gap-2 mb-4 justify-content-center" id="data-container">
<?php
if ($result && $result->num_rows > 0) {
$modalCounter = 0; // Initialize a counter for unique modal IDs
while ($row = $result->fetch_assoc()) {
$modalCounter++; // Increment the counter for each iteration
$modalID = "viewcompany" . $modalCounter; // Unique modal ID for each job post
?>
<div class="card mb-3 p-2 text-center" style="width: 350px; height:475px;">
<div class="header">
" alt="" class="float-start mt-1 mx-3"
style="float: left; width: 40px; height: 40px; border-radius: 40px; overflow: hidden;"
class="btn btn-primary" data-bs-toggle="modal" data-bs-target="#<?php echo
$modalID; ?>">
<p class="p-2 m-1 text-start pointer-event " data-bs-toggle="modal" data-bs-
target="#<?php echo $modalID; ?>"><?php echo $row["cname"]; ?> <small><?php
echo $row["city"]; ?></small></p>
</div>
<div class="card-body">
" class="card-img-top img-fluid "
style="height:200px; object-fit:contain; object-position:center;" id="cardpreview"
alt="" onclick="previewImage(this);">
<div class="preview-container" onclick="closePreview()">
<div class="preview-content">
<img src="" class="preview-image" id="fullImage" alt="Full Image Preview">
</div>
</div>
<h5 class="card-title m-3">Job title</h5>
<p class="card-text"><?php echo $row["title"]; ?></p>
<button class="btn btn-dark" data-bs-toggle="modal" data-bs-
target="#ViewJob<?php echo $modalID; ?>">View Full Details</button>
<p class="card-text mt-3"><small class="text-body-secondary"
id="lastUpdated">Posted on: <?php echo $row["insert_time"]; ?></small></p>
</div>
</div>
<!-- Modal for company profile -->
<div class="modal fade" id="<?php echo $modalID; ?>" tabindex="-1" aria-
labelledby="exampleModalLabel" aria-hidden="true">
<div class="modal-dialog modal-dialog-scrollable modal-fullscreen">
<div class="modal-content">
<div class="modal-header">
<h1 class="modal-title fs-5" id="exampleModalLabel">Company Profile</h1>
<button type="button" class="btn-close" data-bs-dismiss="modal" aria-
label="Close"></button>
</div>
<div class="modal-body">
<div class="container mt-5 mb-5 rounded-2 shadow">

```

```

<!-- Cover Photo -->
<div class="row">
<div class="col-12">
" class="img-fluid mt-3 w-100
rounded-4 " alt="" style="width: 800px; height: 250px; object-fit: cover;">
</div>
</div>
<!-- Profile Photo -->
<div class="row p-2">
<div class="col-lg-4 col-12">
<div class="m-5">
<center>" class="img-fluid rounded-
2 shadow mt-4" style="width: 300px; height: 300px; object-fit: cover;"></center>
</div>
</div>
<!-- Company Profile -->
<div class="col-lg-8 col-12 p-4">
<div class="row p-4">
<div class="col-lg-6 col-12">
<h4 class="mb-3"><i class="bi bi-building-fill"> Company Name</i></h4>
<h6 class="mb-5"><?php echo $row["cname"]; ?></h6>
</div>
<div class="col-lg-6 col-12">
<h4 class="mb-3"><i class="bi bi-geo-alt-fill"> Company Address </i></h4>
<address class="mb-5">
<?php echo $row["caddress"]; ?>,
<?php echo $row["city"]; ?> - <?php echo $row["pincode"]; ?>,
<?php echo $row["state"]; ?>,
<?php echo $row["country"]; ?>.</address>
</div>
<div class="col-lg-6 col-12">
<h4 class="mb-3"><i class="bi bi-envelope-fill"> Company Email</i></h4>
<h6 class="mb-5"><?php echo $row["cemail"]; ?></h6>
</div>
<div class="col-lg-6 col-12">
<h4 class="mb-3"><i class="bi bi-telephone-fill"> Company Website</i></h4>
<h6 class="mb-5"><a href="<?php echo $row["cweb"]; ?>" class="text-decoration-
none text-info">View Website</a></h6>
</div>
<div class="col-lg-6 col-12">
<h4 class="mb-3"><i class="bi bi-person-fill"> HR Name</i></h4>
<h6 class="mb-5"><?php echo $row["hrname"]; ?></h6>
</div>
<div class="col-lg-6 col-12">
<h4 class="mb-3"><i class="bi bi-envelope-fill"> HR Email</i></h4>
<h6 class="mb-5"><?php echo $row["hremail"]; ?></h6>
</div>
</div>
</div>
</div>
</div>

```

```

</div>
</div>
<div class="modal-footer">
<button type="button" class="btn btn-secondary" data-bs-
dismiss="modal">Close</button>
</div>
</div>
</div>
</div>
<!-- Job details -->
<div class="modal fade" id="ViewJob<?php echo $modalID; ?>" tabindex="-1" aria-
labelledby="exampleModalLabel" aria-hidden="true">
<div class="modal-dialog modal-dialog-scrollable modal-fullscreen">
<div class="modal-content">
<div class="modal-header">
<h1 class="modal-title fs-5" id="exampleModalLabel">Job Details</h1>
<button type="button" class="btn-close" data-bs-dismiss="modal" aria-
label="Close"></button>
</div>
<div class="modal-body">
<div class="container text-start">
<div class="row ">
<div class="col-md-6 d-flex justify-content-center">
" class="img-fluid rounded-2 shadow
mt-4" id="dynamicImage" style="max-width: 100%; max-height: 70vh; object-fit:
cover;">
</div>
<div class="col-md-6 align-self-center">
<h4 class="text-start mt-3">Job Title</h4>
<span><?php echo $row["title"]; ?></span>
<h4 class="text-start mt-3">Job Description</h4>
<p class="" style="text-align: justify;"><?php echo $row["description"]; ?></p>
<h4 class="text-start mt-3">Eligibility criteria</h4>
<span>Eligibility <?php echo $row["stream"]; ?></span>
<p class="mt-5"><small class="text-body-secondary" id="lastUpdated">Posted on:
<?php echo $row["insert_time"]; ?></small></p>
<hr>
<!-- job form -->
<?php
// Check if the user has already applied for the job
$checkAppliedQuery = "SELECT * FROM jobapplication WHERE rollno = '$rollno'
AND postid = " . $row['postid'];
$checkResult = $conn->query($checkAppliedQuery);
// If the user has already applied, do not display the form
if ($checkResult && $checkResult->num_rows > 0) {
// User has already applied, do not display the form
// Optionally, you can display a message indicating that the user has already applied
echo "<b class='text-danger'>You have already applied for this job.</b>";
} else {
// User has not applied for the job, display the form

```

```

if ($row['form'] == 'Activate') {
// Display the form design
echo '<button type="button" class="btn btn-dark mb-3" data-bs-toggle="offcanvas"
data-bs-target="#Form' . $modalID . '">Fill Form</button>';
} else {
// Do not display the form
// echo "<p>Form Not Activated for $modalID</p>";
}
}
?>
<!-- Form design on OFFCANVAS -->
<div class="offcanvas offcanvas-top h-100 " data-bs-backdrop="static" tabindex="-1"
id="Form<?php echo $modalID; ?>" aria-labelledby="staticBackdropLabel">
<div class="offcanvas-header">
<h5 class="offcanvas-title" id="staticBackdropLabel">Application</h5>
<button type="button" class="btn-close" data-bs-dismiss="offcanvas" aria-
label="Close"></button>
</div>
<div class="offcanvas-body">
<div class="container">
<form method="POST" class="w-100 rounded-1 m-3 p-4 border bg-white"
action="jobappl.php" enctype="multipart/form-data">
<input required name="userid" type="text" class="form-control" placeholder=""
value="<?php echo $row["userid"]; ?>" hidden />
<input required name="postid" type="text" class="form-control" placeholder=""
value="<?php echo $row["postid"]; ?>" hidden />
<span class="form-label d-block">Company name</span>
<input required name="cname" type="text" class="form-control mb-4"
placeholder="" value="<?php echo $row["cname"]; ?>" readonly />
<span class="form-label d-block">Job Title</span>
<input required name="title" type="text" class="form-control mb-4" placeholder=""
value="<?php echo $row["title"]; ?>" readonly />
<?php
$sql = "SELECT * FROM studentlogin where rollno='$rollno'";
$profile = $conn->query($sql);
while ($row = mysqli_fetch_assoc($profile)) {
?>
<span class="form-label d-block">Your name</span>
<input required name="name" type="text" class="form-control" placeholder="Name"
value="<?php echo $row["name"]; ?>" />
<span class="form-label d-block mt-3">Roll Number</span>
<input required name="rollno" type="text" class="form-control"
placeholder="RollNumber" value="<?php echo $row["rollno"]; ?>" />
<span class="form-label d-block mt-3">Contact</span>
<input required name="phone" type="text" class="form-control" placeholder="Phone
Number" value="<?php echo $row["phone"]; ?>" />
<span class="form-label d-block mt-3">Email</span>
<input required name="email" type="email" class="form-control"
placeholder="Email ID" value="<?php echo $row["email"]; ?>" />
<span class="form-label d-block mt-3">Stream</span>

```

```

<input required name="course" type="text" class="form-control" placeholder="UG
OR PG" value="<?php echo $row["course"]; ?>" />
<span class="form-label d-block mt-3">Department</span>
<input required name="dept" type="text" class="form-control"
placeholder="MSCIT,MSCCS" value="<?php echo $row["dept"]; ?>" />
<?php
}
?>
<span class="form-label d-block mt-3">Semester Percentage</span>
<input required name="per" type="text" class="form-control" placeholder="Enter the
Percentage Till Concluded Semester" />
<span class="form-label d-block mt-3">Your CV</span>
<input required name="user_file1" type="file" class="form-control" />
<div class="d-flex gap-3">
<button type="submit" class="btn btn-dark w-50 mt-3 rounded-3">Submit</button>
<button type="reset" class="btn btn-dark w-50 mt-3 rounded-3">Clear</button>
</div>
</form>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
<div class="modal-footer">
<button type="button" class="btn btn-secondary" data-bs-
dismiss="modal">Close</button>
</div>
</div>
</div>
</div>
<?php
}
} else {
echo "<p>No job posts available</p>";
}
?>
</div>
</div>
</section>
<?php
// Retrieve job post data along with the respective user profiles
$PostQuery = "SELECT sp.*, sf.*, sp.stream AS stream FROM staffpost sp INNER
JOIN staffform sf ON sp.idnum = sf.idnum WHERE sp.stream = 'For PG' OR
sp.stream = 'Both UG and PG'";
$result = $conn->query($PostQuery);
?>
<!-- Staff Post -->
<section id="staffpost" style="display: none;">

```



```

<div class="d-flex flex-wrap gap-2 mb-4 justify-content-center" id="data-container">
<?php
if ($result && $result->num_rows > 0) {
    $modalCounter = 0; // Initialize a counter for unique modal IDs
    while ($row = $result->fetch_assoc()) {
        $modalCounter++; // Increment the counter for each iteration
        $modalID = "viewstaff" . $modalCounter; // Unique modal ID for each job post
        ?>
        <div class="card mb-3 p-2 text-center" style="width: 350px; height:475px;">
        <div class="header">
        " alt="" class="float-start mt-1 mx-3"
        style="float: left; width: 40px; height: 40px; border-radius: 40px; overflow: hidden;"
        class="btn btn-primary" data-bs-toggle="modal" data-bs-target="#<?php echo
        $modalID; ?>">
        <p class="p-2 m-1 text-start pointer-event " data-bs-toggle="modal" data-bs-
        target="#<?php echo $modalID; ?>"><?php echo $row["name"]; ?> <small><?php
        echo $row["role"]; ?></small></p>
        </div>
        <div class="card-body">
        " class="card-img-top img-fluid "
        style="height:200px; object-fit:contain; object-position:center;" id="cardpreview2"
        alt="">
        <h5 class="card-title m-3">Title</h5>
        <p class="card-text"><?php echo $row["title"]; ?></p>
        <button class="btn btn-dark" data-bs-toggle="modal" data-bs-
        target="#ViewJob<?php echo $modalID; ?>">View Full Details</button>
        <p class="card-text mt-3"><small class="text-body-secondary"
        id="lastUpdated">Posted on: <?php echo $row["insert_time"]; ?></small></p>
        </div>
        </div>
        <!-- Modal for Staff Profile -->
        <div class="modal fade" id="<?php echo $modalID; ?>" tabindex="-1" aria-
        labelledby="exampleModalLabel" aria-hidden="true">
        <div class="modal-dialog modal-dialog-scrollable modal-fullscreen">
        <div class="modal-content">
        <div class="modal-header">
        <h1 class="modal-title fs-5" id="exampleModalLabel">Staff Profile</h1>
        <button type="button" class="btn-close" data-bs-dismiss="modal" aria-
        label="Close"></button>
        </div>
        <div class="modal-body">
        <div class="container mt-5 mb-5 rounded-2 shadow">
        <!-- Profile Photo -->
        <div class="row p-2">
        <div class="col-lg-4 col-12">
        <div class="m-5">
        <center>" class="img-fluid rounded-
        2 shadow mt-4" style="width: 300px; height: 300px; object-fit: cover;"></center>
        </div>
        </div>

```

```

<!-- Staff Profile -->
<div class="col-lg-8 col-12 p-4">
<div class="row p-4">
<div class="col-md-6 col-12 mb-3">
<h3>Name</h3>
<p><?php echo $row["name"]; ?></p>
</div>
<div class="col-md-6 col-12 mb-3">
<h3>ID Number</h3>
<p><?php echo $row["idnum"]; ?></p>
</div>
<div class="col-md-6 col-12 mb-3">
<h3>Aided or Self-Financed</h3>
<p><?php echo $row["staff"]; ?></p>
</div>
<div class="col-md-6 col-12 mb-3">
<h3>Teaching or Non-Teaching</h3>
<p><?php echo $row["sttype"]; ?></p>
</div>
<div class="col-md-6 col-12 mb-3">
<h3>Role</h3>
<p><?php echo $row["role"]; ?></p>
</div>
<div class="col-md-6 col-12 mb-3">
<h3>Department</h3>
<p><?php echo $row["dept"]; ?></p>
</div>
<div class="col-md-6 col-12 mb-3">
<h3>Phone Number</h3>
<p><?php echo $row["phone"]; ?></p>
</div>
<div class="col-md-6 col-12 mb-3">
<h3>Email</h3>
<p><?php echo $row["email"]; ?></p>
</div>
</div>
</div>
</div>
</div>
<div class="modal-footer">
<button type="button" class="btn btn-secondary" data-bs-dismiss="modal">Close</button>
</div>
</div>
</div>
</div>
<!-- Event details -->
<div class="modal fade" id="ViewJob<?php echo $modalID; ?>" tabindex="-1" aria-labelledby="exampleModalLabel" aria-hidden="true">

```

```

<div class="modal-dialog modal-dialog-scrollable modal-fullscreen">
<div class="modal-content">
<div class="modal-header">
<h1 class="modal-title fs-5" id="exampleModalLabel">Event Details</h1>
<button type="button" class="btn-close" data-bs-dismiss="modal" aria-
label="Close"></button>
</div>
<div class="modal-body">
<div class="container text-start">
<div class="row ">
<div class="col-md-6 d-flex justify-content-center">
" class="img-fluid rounded-2 shadow
mt-4" id="dynamicImage" style="max-width: 100%; max-height: 70vh; object-fit:
cover;">
</div>
<div class="col-md-6 align-self-center">
<h4 class="text-start mt-3">Title</h4>
<span><?php echo $row["title"]; ?></span>
<h4 class="text-start mt-3">Description</h4>
<p class="" style="text-align: justify;"><?php echo $row["description"]; ?></p>
<h4 class="text-start mt-3">Eligibility criteria</h4>
<span><?php echo $row["stream"]; ?> can attend</span>
<p class="mt-5"><small class="text-body-secondary" id="lastUpdated">Posted on:
<?php echo $row["insert_time"]; ?></small></p>
<hr>
</div>
</div>
</div>
</div>
</div>
<div class="modal-footer">
<button type="button" class="btn btn-secondary" data-bs-
dismiss="modal">Close</button>
</div>
</div>
</div>
</div>
<?php
}
} else {
echo "<p>No posts available</p>";
}
?>
</div>
</div>
</section>
<!-- Contactus -->
<section id="contact">
</section>
<!-- Contactus -->
<!-- libraries -->

```

```

<script src="./Assets/js/jquery.min.js"></script>
<script src="./Assets/js/bootstrap.min.js"></script>
<script src="./Assets/js/popper.min.js"></script>
<script src="./Assets/js/departement.js"></script>
<script
src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.11.8/dist/umd/popper.min.js"></
script>
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.min.js"></script>
<!-- script -->
<script>
// Full Image Preview
function previewImage(img) {
// Show the preview container
var previewContainer = document.querySelector('.preview-container');
previewContainer.style.display = 'block';
// Update the preview image source with the clicked image source
var fullImage = document.getElementById('fullImage');
fullImage.src = img.src;
}
function closePreview() {
// Hide the preview container
var previewContainer = document.querySelector('.preview-container');
previewContainer.style.display = 'none';
}
// image ratio
window.onload = function() {
var img = document.getElementById('dynamicImage');
img.onload = function() {
if (img.naturalWidth > img.naturalHeight) {
img.style.width = 'auto';
img.style.height = '100%';
} else {
img.style.width = '100%';
img.style.height = 'auto';
}
};
};
const popoverTriggerList = document.querySelectorAll('[data-bs-toggle="popover"]')
const popoverList = [...popoverTriggerList].map(popoverTriggerEl => new
bootstrap.Popover(popoverTriggerEl, {
trigger: 'focus'
}))
// JavaScript to handle changing the selected category and toggle visibility of sections
document.getElementById('recruiterBtn').addEventListener('click', function() {
document.getElementById('post').style.display = 'block';
document.getElementById('staffpost').style.display = 'none';
document.getElementById('recruiterBtn').classList.add('active');
document.getElementById('staffBtn').classList.remove('active');
});

```

```
document.getElementById('staffBtn').addEventListener('click', function() {  
document.getElementById('post').style.display = 'none';  
document.getElementById('staffpost').style.display = 'block';  
document.getElementById('recruiterBtn').classList.remove('active');  
document.getElementById('staffBtn').classList.add('active');  
});  
</script>  
</body>  
</html>
```

6. SYSTEM TESTING

System testing, conducted on a fully integrated system, assesses its adherence to specified requirements with both functional and non-functional aspects examined. Carried out by specialized or independent testers, this process involves identifying and rectifying errors caused by improper inputs. Ultimately, system testing serves as validation that the system operates as intended and offers users reassurance of its functionality.

6.1 PROJECT TESTING

Testing comprises various tests aimed at thoroughly exercising computer-based systems. While each test serves a distinct purpose, collectively they ensure proper integration of all system elements and their allocated functions. The testing process involves verifying whether the developed system aligns with the actual requirements and objectives.

6.1.1 UNIT TESTING

Unit testing involves testing individual modules to detect any issues, conducted by the developer. Its primary objective is to isolate system units, identify defects, and resolve them. Usually executed by software developers, unit tests ensure code adherence to its design and intended behavior, aiming to validate the correctness of each program component.

The screenshot shows a 'Student Register' form with the following fields: Name, Roll Number, Phone Number, Email address, Select Course (dropdown), Select Department (dropdown), Year of Passing, and Password. Below the Password field is a note: 'Your password must be 8-20 characters long, contain letters and numbers, and must not contain spaces, special characters, or emoji.' At the bottom, there is a link 'Already have an account? Sign In' and two buttons: 'Register' and 'Clear'.

Fig 6.1 Unit Testing on Registration Page

6.1.2 INTEGRATION TESTING

Integration testing, occurring after unit testing, constitutes the second phase of the software testing process. This phase involves testing groups of units or individual components of the software together. The primary objective of integration testing is to reveal defects during the interaction between integrated components or units. Unlike unit testing, which focuses on testing modules individually, integration testing combines these modules to assess their functionality. Given that software is typically developed with multiple modules coded by different programmers, integration testing aims to verify the accuracy of communication among all these modules.

The screenshot shows a 'Student Login' form with the following fields: Roll Number and Password. Below the Password field is a link 'Don't have an account yet? Sign Up'. At the bottom, there are two buttons: 'Login' and 'Clear'.

Fig 6.2 Integration Testing on Login Page

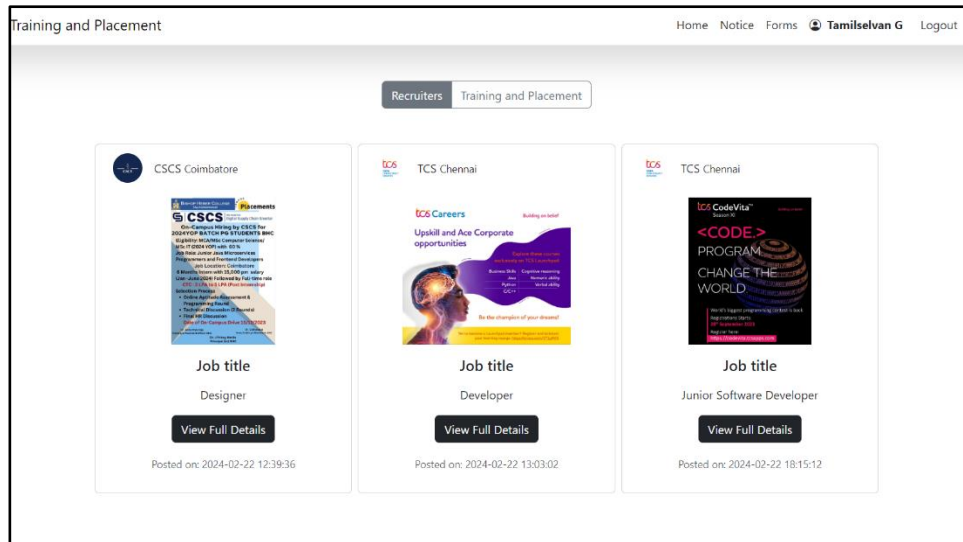


Fig 6.3 Integration Testing on Student Home Page

6.2 VALIDATION TESTING

In software engineering, validation testing serves to ascertain whether the current system adheres to its specified requirements, executes its designated functions, and aligns with the organization's objectives and needs. This testing method holds significant importance, particularly for aspiring top-tier software testers. Following the validation testing stage, the software verification and validation process prioritizes verification testing.

The screenshot shows the 'Student Register' form. The form contains the following fields and elements:

- Name:** A text input field with a blue border, indicating it is the active field.
- Roll Number:** A text input field.
- Phone:** A text input field with a validation error message: 'Please fill out this field.'
- Email address:** A text input field.
- Select Course *:** A dropdown menu with 'Select Course' as the selected option.
- Select Department *:** A dropdown menu with 'Select Department' as the selected option.
- Year of Passing:** A text input field.
- Password:** A text input field.
- Password Hint:** A small text note below the password field: 'Your password must be 8-20 characters long, contain letters and numbers, and must not contain spaces, special characters, or emoji.'
- Already have an account? Sign In:** A link to the login page.
- Register:** A button to submit the form.
- Clear:** A button to reset the form.

Fig 6.4 Validation Testing on Registration Page

Table 6.1 Validation Testing on Registration page

Field name	Field Type	Validation Test
Name	text	Required Field Validator
Rollno	number	Required Field Validator
Mobile Number	number	Regular Expression Validator
		Required Field Validator
Email Address	email	Regular Expression Validator
		Required Field Validator
Course	select	Required Field Validator
Department	select	Required Field Validator
Year of Passing	text	Required Field Validator
Password	password	Required Field Validator

Fig 6.5 Validation Testing on Create Post Page**Table 6.2** Validation Testing on Create Post Page

Field name	Field Type	Validation Test
Stream	select	Required Field Validator
Year of Passing	number	Required Field Validator
Job Title	text	Required Field Validator
Job Description	text	Required Field Validator
From	select	Required Field Validator

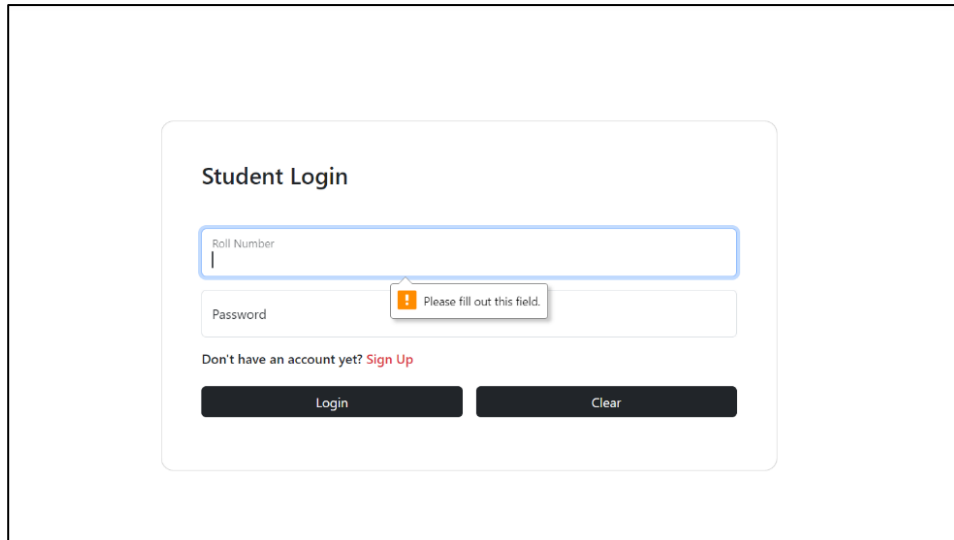


Fig 6.6 Validation Testing on Login Page

Table 6.3 Validation Testing on Login Page

Field name	Field Type	Validation Test
Rollno	number	Required Field Validator
Password	password	Required Field Validator

7. SYSTEM IMPLEMENTATION

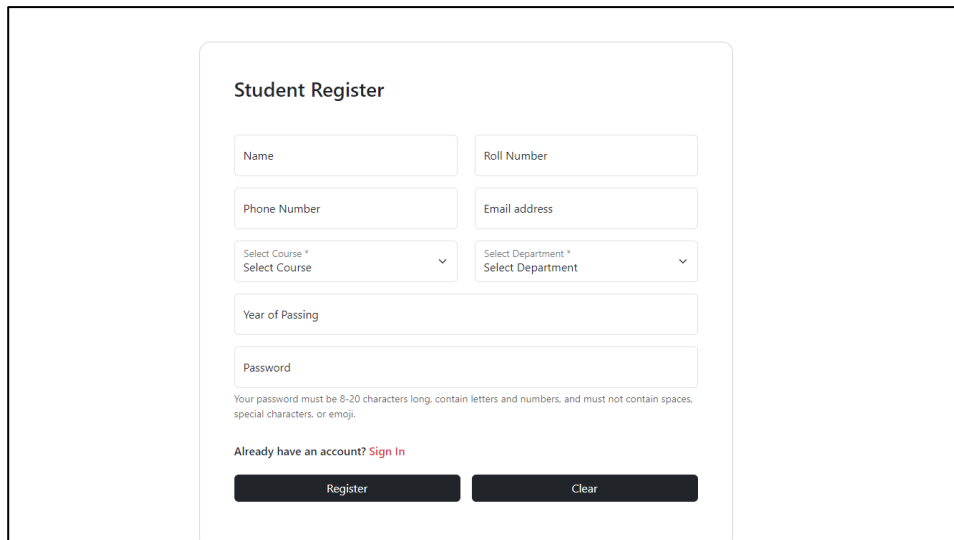
Implementation stands as the pivotal phase in realizing a successful system and instilling user confidence in its workability and effectiveness. It involves the adaptation of an application to supplant an existing one, typically manageable when substantial system changes are absent. This stage transitions the theoretical design into a functional system through meticulous planning, evaluation of the current system's constraints, development of changeover strategies, and assessment of their effectiveness. Importantly, the conversion process must seamlessly integrate with organizational operations to prevent disruption, underscoring its criticality in ensuring the system's success and user assurance.

7.1 SOFTWARE DEMONSTRATION

Demo software refers to a trial edition of a software application that permits users to utilize it at no cost as they contemplate purchasing it. For instance, such software might restrict certain functionalities, such as saving files, allowing users to explore its functionality without the ability to retain their work.

7.1.1 STUDENT REGISTRATION PAGE

Through the registration page students can create their profile using details like name, rollno, year of passing, mobile number, email address, course, department and password.

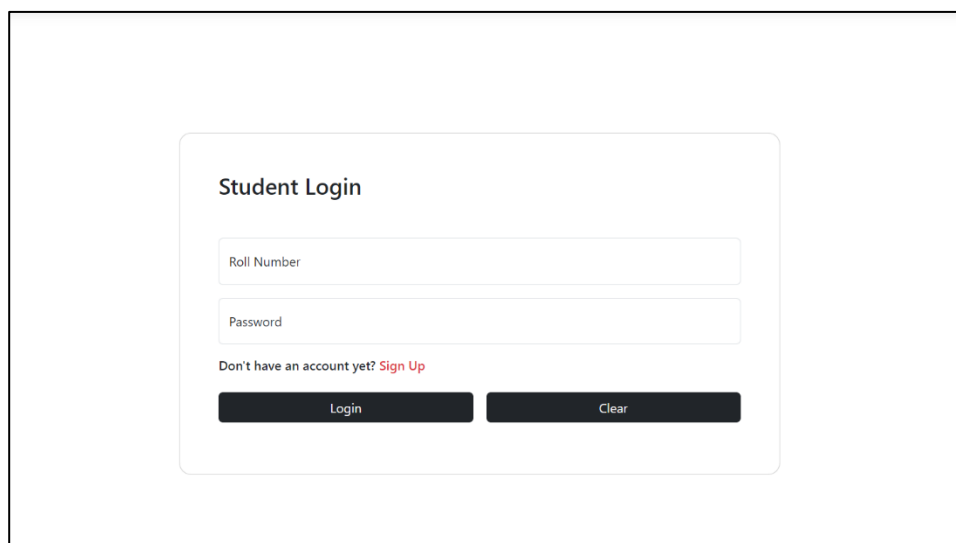


The image shows a 'Student Register' form. It contains the following fields: 'Name', 'Roll Number', 'Phone Number', 'Email address', 'Select Course *' (a dropdown menu), 'Select Department *' (a dropdown menu), 'Year of Passing', and 'Password'. Below the password field, there is a note: 'Your password must be 8-20 characters long, contain letters and numbers, and must not contain spaces, special characters, or emoji.' At the bottom, there is a link 'Already have an account? Sign In' and two buttons: 'Register' and 'Clear'.

Fig 7.1 Student Registration Page

7.1.2 STUDENT LOGIN PAGE

After creating their profile successfully students can login to the application using their rollno and password.



The image shows a 'Student Login' form. It contains the following fields: 'Roll Number' and 'Password'. Below the password field, there is a link 'Don't have an account yet? Sign Up'. At the bottom, there are two buttons: 'Login' and 'Clear'.

Fig 7.2 Student Login Page

7.1.3 STUDENT HOME PAGE

If the entered credentials are true in the login page it redirected to student home page.

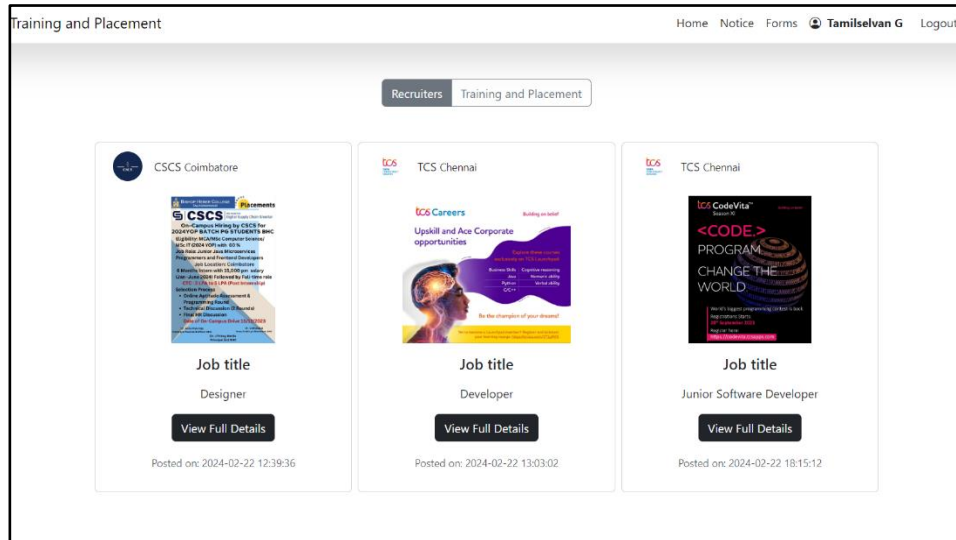


Fig 7.3 Student Home Page

7.1.4 STUDENT PROFILE PAGE

By clicking on the profile picture students can access profile page. Through my profile page users can update their profile details.

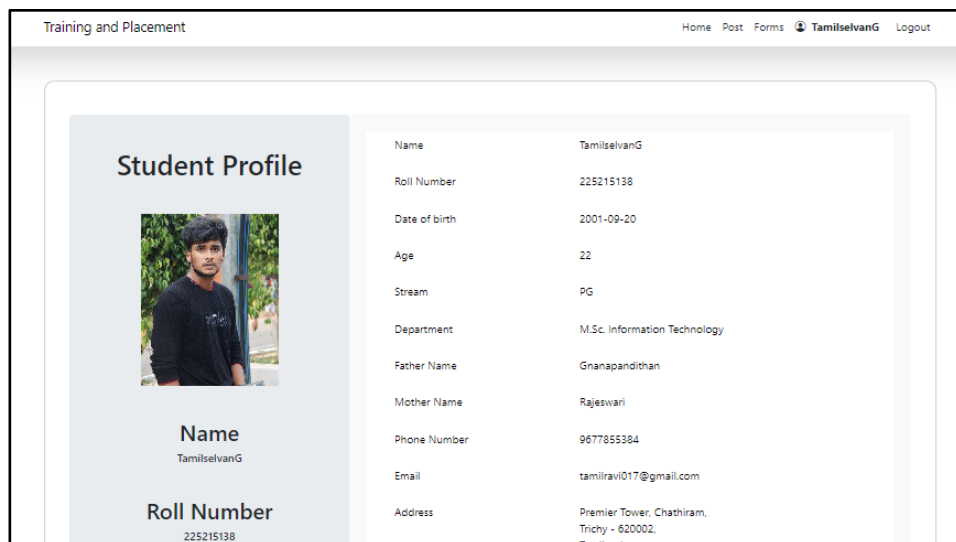
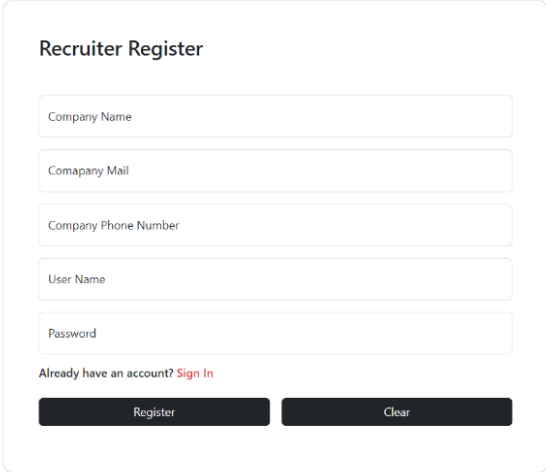


Fig 7.4 Student Profile Page

7.1.5. RECRUITER REGISTRATION PAGE

Through the registration page recruiters can create their profile using details like company name, company email, phone number, username and password.

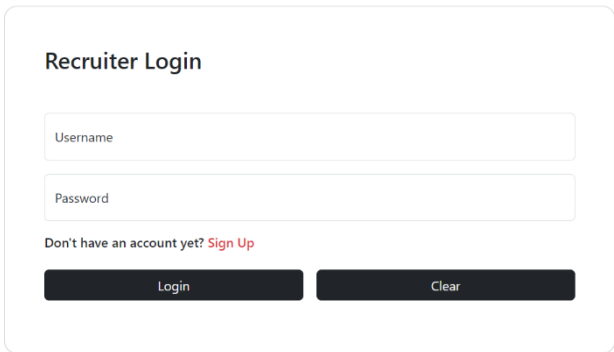


The image shows a web form titled "Recruiter Register". It contains five input fields: "Company Name", "Comapany Mail" (note the typo), "Company Phone Number", "User Name", and "Password". Below the fields is a link that says "Already have an account? Sign In". At the bottom are two buttons: "Register" and "Clear".

Fig 7.5 Recruiter Registration Page

7.1.6 RECRUITER LOGIN PAGE

After creating their profile successfully recruiters can login to the application using their username and password.



The image shows a web form titled "Recruiter Login". It contains two input fields: "Username" and "Password". Below the fields is a link that says "Don't have an account yet? Sign Up". At the bottom are two buttons: "Login" and "Clear".

Fig 7.6 Recruiter Login Page

7.1.7 RECRUITER HOME PAGE

If the entered credentials are true in the login page it redirected to recruiter home page.



Fig 7.7 Recruiter Home Page

7.1.8 RECRUITER PROFILE PAGE

By clicking on the profile recruiter can access my profile page. Through profile page recruiters can update their profile details, create post, view post and manage job application.

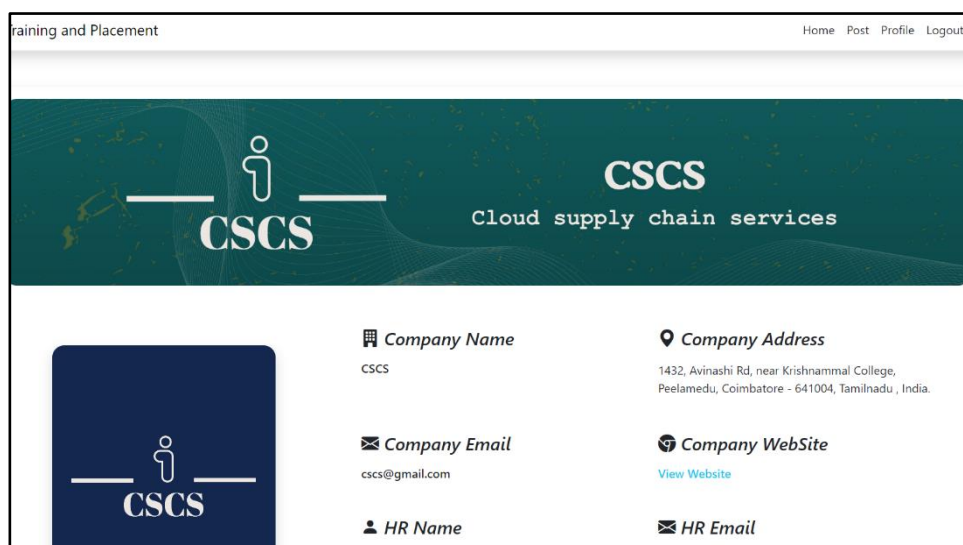


Fig 7.8 Recruiter Profile Page

7.1.9 RECRUITER CREATE POST PAGE

Through the new post recruiters can create their post and publish by using user id of the recruiter, title, stream, year of passing, description, form, photo_path and posted date.

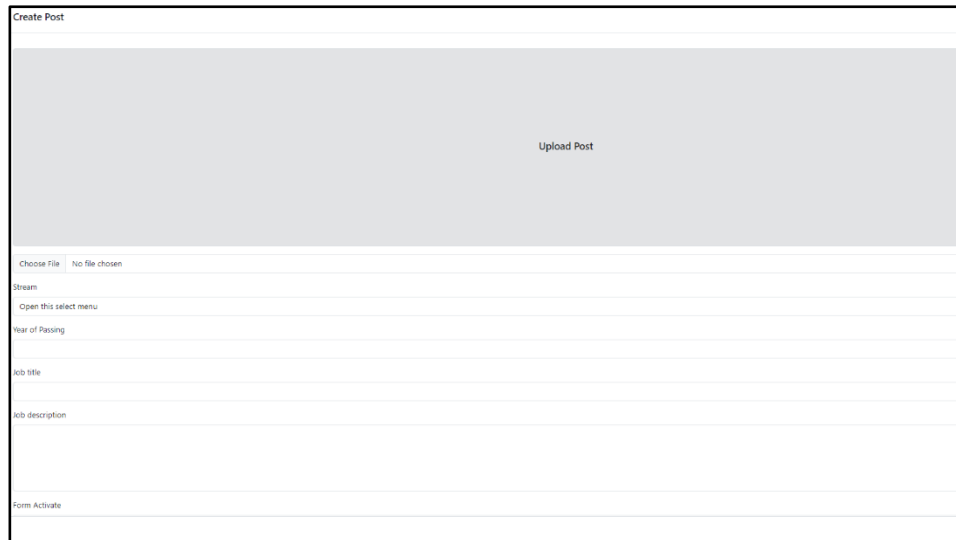
The image shows a web form titled "Create Post". At the top is a large grey rectangular area with the text "Upload Post" centered. Below this are several input fields: a file selection area with "Choose File" and "No file chosen" text, a "Stream" dropdown menu with the placeholder "Open this select menu", a "Year of Passing" text input, an "Job title" text input, a "Job description" text input, and a "Form Activate" checkbox at the bottom.

Fig 7.9 Recruiter Create Post Page

7.1.10 STAFF LOGIN PAGE

After creating their profile successfully staff can login to the application using their id-number and password.

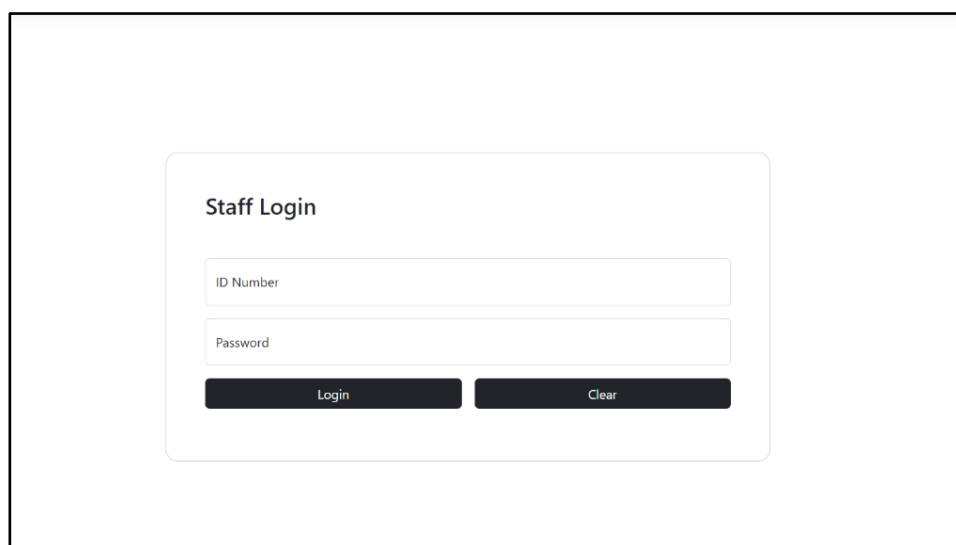
The image shows a "Staff Login" form. It has a title "Staff Login" at the top. Below the title are two input fields: "ID Number" and "Password". At the bottom of the form are two buttons: "Login" and "Clear".

Fig 7.10 Staff Login Page

7.1.11 STAFF HOME PAGE

If the entered credentials are true in the login page it redirected to staff home page.

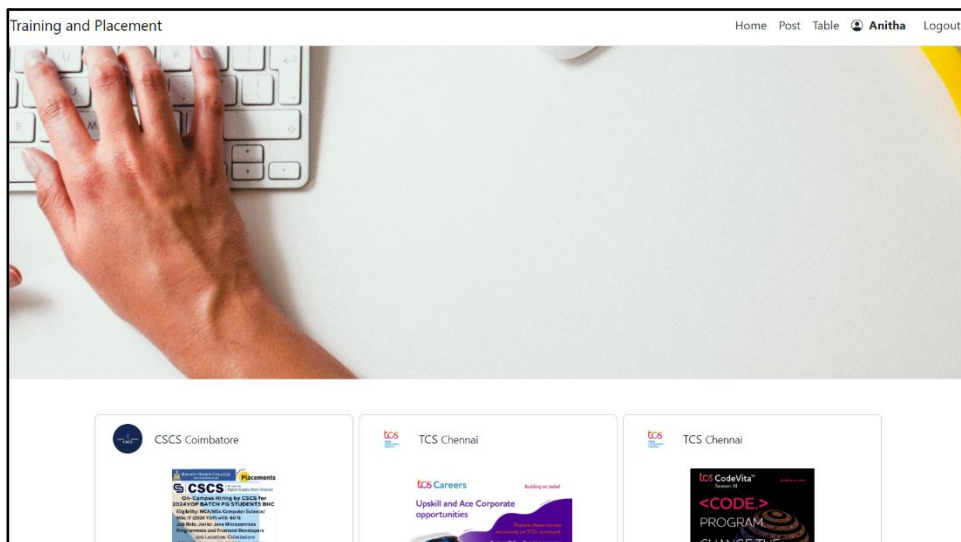


Fig 7.11 Staff Home page

7.1.12 STAFF PROFILE PAGE

By clicking on the name staff can access staff profile page. Through profile page users can update their profile details, create post, view post.

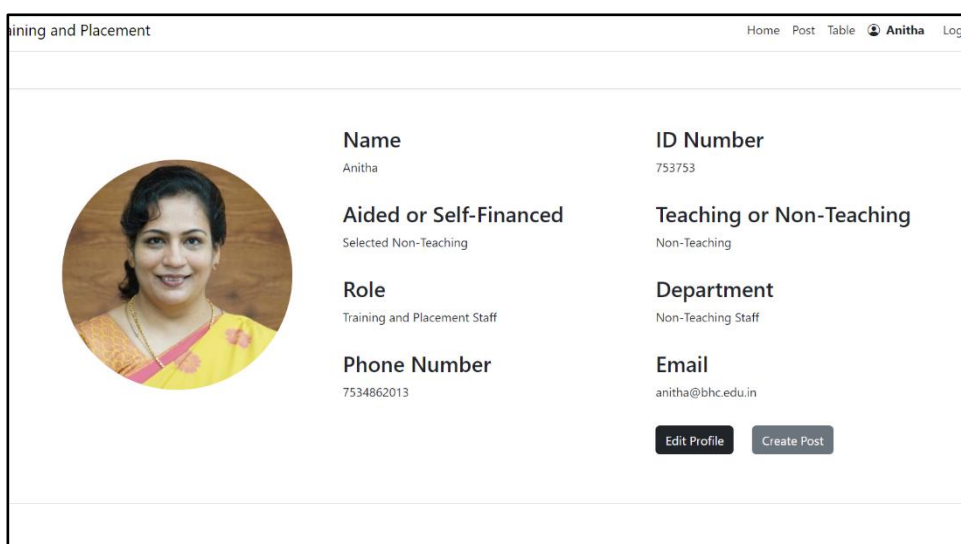


Fig 7.12 Staff Profile Page

7.1.13 STAFF CREATE POST

Through the new post staff can create their post and publish by using user id of the recruiter, title, stream, year of passing, description, form, photo_path and posted date.

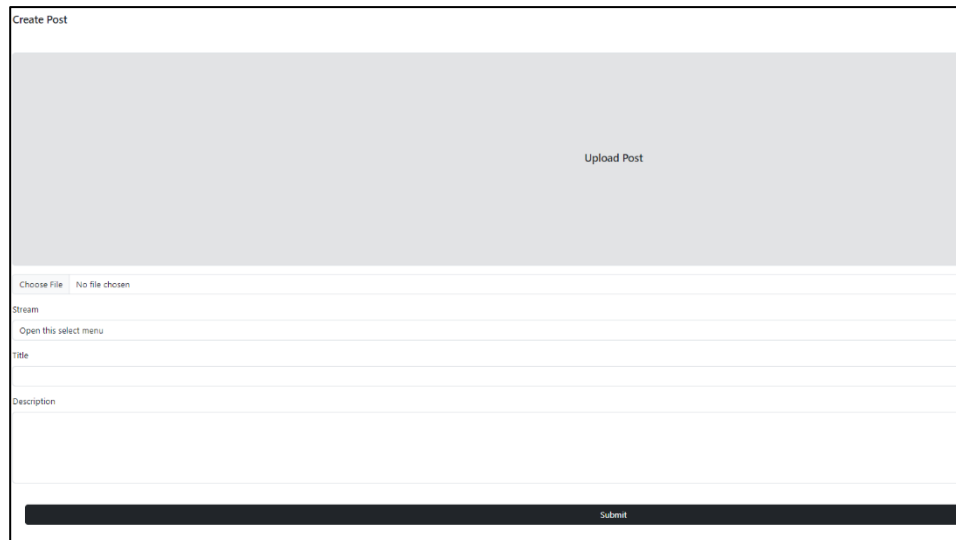
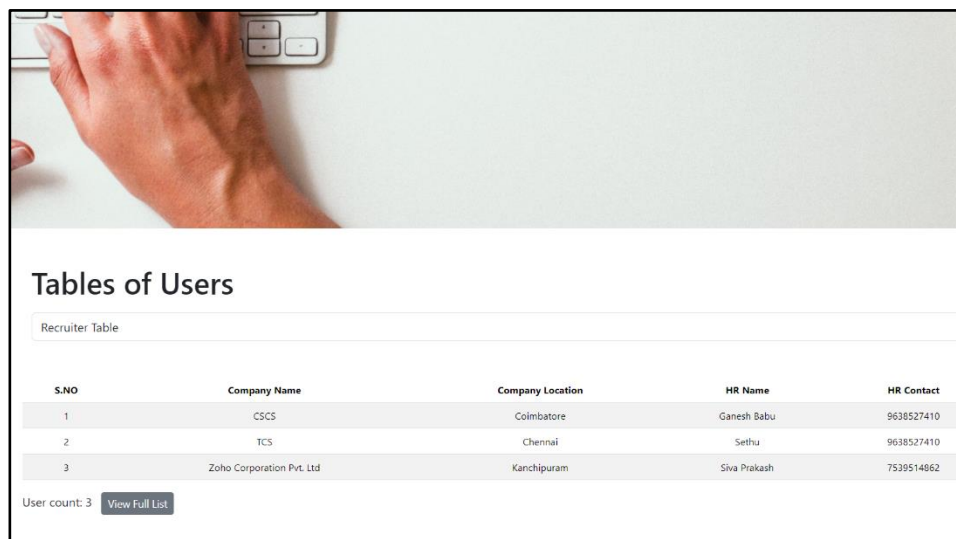


Fig 7.13 Staff Create Post

7.1.14 STAFF MANAGE PAGE

Staff can view recruiter and student details.



S.NO	Company Name	Company Location	HR Name	HR Contact
1	CSCS	Coimbatore	Ganesh Babu	9638527410
2	TCS	Chennai	Sethu	9638527410
3	Zoho Corporation Pvt. Ltd	Kanchipuram	Siva Prakash	7539514662

User count: 3 [View Full List](#)

Fig 7.14 Staff Manage Page

8. CONCLUSION

The "Web Application for Campus Placement" represents a significant advancement in the management of campus placements within our college. By transitioning from manual, fragmented processes to a centralized, automated platform, we aim to address the inefficiencies and challenges inherent in the existing system.

Through the implementation of various modules such as user authentication, student, recruiter, staff, dashboard, and reporting, the proposed system offers a comprehensive solution for managing all aspects of the placement process.

In summary, the "Web Application for Campus Placement" represents a modern, efficient, and user-friendly solution tailored to meet the evolving needs of our college's placement ecosystem. Through its implementation, anticipate for transformative impact on placement processes, ultimately facilitating better outcomes for students and recruiters alike.

BIBLIOGRAPHY

Book References:

1. Core PHP programming By Leon Atkinson, Zeev Suraski, Third Edition -2004
2. PHP Object-Oriented Solutions by David Powers-2008
3. PHP ALL SCRIPT Zeev Suraski, Third Edition, -2004
4. Jon Duckett, “HTML and CSS: Design and Built Web Sites”, First Edition, Wiley,2011.
5. Vikram Vaswani, “The Complete Reference on MySql”, First Edition, McGrawHill Education, 2017.

Web References:

1. www.stackoverflow.com
2. www.youtube.com
3. www.geekforgeeks.com
4. www.bootstrap.com
5. <https://chat.openai.com/>