A

Project Report

On

"Campus Placement And Recruitment System"

MCA113-MINI PROJECT

By

"HRISHIKESH SHINDE"

"31231726"

Mini Project work carried out at

Faculty of Commerce and Management

VISHWAKARMA UNIVERSITY

PUNE

April 2024

DECLARATION

I, the undersigned, hereby declare that the project report entitled **“** Campus Placement And Recruitment System**”** written and submitted by me to Vishwakarma University, Pune in partial fulfillment of the requirement for Mini Project (Semester II) as part of two year full-time MCA programme. The project is for academic purposes only. This is the original work of the undersigned and has not been reproduced from any other sources.

Signature:

Name of the Student: HRISHIKESH SHINDE

PRN: 3123015010218

Date:

Place : Vishwakarma University, Pune

Mini Project Completion Certificate

MCA Programme

Batch : 2023-2025

This is to certify that Mr. HRISHIKESH SHINDE. student of Master of Computer Application (MCA) Programme under Faculty of Commerce and Management at Vishwakarma University, Pune, has completed the Mini Project titled **"**Campus Placement And Recruitment System**"** as per university guidelines for the two-year full-time MCA programme.

Faculty Guide (Name):

Faculty Guide (Signature):

ACKNOWLEDGEMENTS

This project would not have been possible without the support of many people. I would like to acknowledge and extend my heartfelt gratitude to the following persons who have made the completion of this project possible.

My Supervisor, **Prof. Yogesh Desale** for his full support and guidance throughout this project. My additional examiner **Gouri KulKarni** for reviewing project and helping me in design. My dean **Dr. Jayashree Vispute** for encouraging and supporting me to pursue the project.

I also thank to Vishwakarma University, Pune team for step by step updating regarding project with nice taxila support.

My team members who have helped me during various phases of the project. Last but not the least; I would like to express my love and gratitude to my beloved family, for their understanding & motivation, through the duration of this project.

ABSTRACT

The "Campus Placement Recruitment System" is a comprehensive and user-centric solution designed to streamline and enhance the process of campus placements for educational institutions and students alike. This innovative system offers a feature-rich platform that connects students, educational institutions, and potential employers, creating a seamless and efficient ecosystem for recruitment activities.

The proposed system enables students to search for upcoming placement opportunities, apply for positions that match their qualifications and interests, and conveniently view the results of placement processes. Additionally, even after graduating, students can access and review placement details, providing ongoing career support. This platform not only simplifies placement processes but also fosters transparency, accessibility, and improved communication between all stakeholders. It stands as a testament to the commitment to empower students in their pursuit of meaningful and rewarding careers.

Administrators can efficiently update placement records, track student progress, and collaborate with industry partners for placement opportunities. The system incorporates robust security measures to safeguard sensitive student information and ensure data integrity.

the system employs object-oriented principles and database connectivity to store, retrieve, and manipulate data efficiently. Its modular design facilitates scalability, allowing for future enhancements and customizations as per specific institutional requirements.

Overall, the The "campus placement and recruitment system. in php serves as a valuable tool for educational institutions to optimize their placement processes, fostering better opportunities for students and strengthening partnerships with recruiting organizations.

Signature of the Student	**Signature of the Supervisor**
Name: HRISHIKESH SHINDE.	**Name: Prof. Yogesh Desale**
Date:	**Date:**
Place: Pune	**Place: Pune**

TABLE OF CONTENTS

List of Figures

Chapter 1:-Introduction

1.1 what is campus placement and Recruitment SysTem?:-

The "Campus Placement Recruitment System" is a comprehensive and user-centric solution designed to streamline and enhance the process of campus placements for educational institutions and students alike. This innovative system offers a feature-rich platform that connects students, educational institutions, and potential employers, creating a seamless and efficient ecosystem for recruitment activities.

In PHP, a campus placement and Recruitment System would typically be developed as a software application using the PHP programming language.

1. A campus placement and recruitment system developed using PHP would typically follow this steps Registration and Profile Creation**:** Students create accounts on the platform by providing necessary information such as personal details, academic records, skills, and resume/CV upload.
2. Job Posting by Companies: Employers register on the platform and post job openings, including details such as job description, required qualifications, skills, and other relevant information.
3. Student Applications: Students browse through the job listings and apply for the positions they're interested in. They may also be able to filter jobs based on criteria such as industry, location, and job type.
4. Shortlisting: Employers review the applications received and shortlist candidates based on their qualifications and suitability for the positions.
5. Communication: The platform facilitates communication between employers and shortlisted candidates for scheduling interviews, sharing additional information, and clarifying doubts.
6. Interviews and Assessments: Employers conduct interviews, tests, and assessments either online through the platform or in-person on the campus, depending on the arrangement.
7. Selection and Offer Letters: Employers select the final candidates and extend job offers through the platform. Offer letters may be generated and sent electronically to the selected candidates.
8. Acceptance: Candidates review the job offers and decide whether to accept or decline them through the platform.
9. Placement and Reporting: Once candidates accept offers, the platform facilitates the placement process, and both parties may track the progress. The system may also generate reports and analytics

for colleges/universities to analyze placement trends.

10. Feedback and Ratings: After the placement process, both employers and candidates may provide feedback and ratings on their experience with the platform and each other.

1.1.2Why we need Training and Placement management system:-

A campus placement and recruitment system serves several important purposes for both students and employers, as well as educational institutions:

1. Efficient Matching: It provides a centralized platform for students to explore job opportunities and for employers to connect with potential candidates. This facilitates a more efficient matching process between job seekers and employers.
2. Early Career Opportunities: For students, campus placement systems offer the chance to secure job offers before completing their education. This helps them kick-start their careers immediately after graduation or post-graduation.
3. Industry Exposure: Students gain exposure to various industries, companies, and job roles through interactions with recruiters during pre-placement talks, interviews, and assessments. This exposure helps them make informed career decisions.
4. Industry-Ready Skills: The recruitment process often emphasizes the skills and competencies required by the industry. This encourages students to develop relevant skills and knowledge during their academic tenure, making them more employable.
5. Streamlined Process: For employers, campus placement systems streamline the recruitment process by providing access to a pool of pre-screened candidates who meet their criteria. This saves time and resources compared to traditional recruitment methods.
6. Talent Acquisition: Employers can identify and attract top talent from reputed educational institutions. This helps them build a strong workforce and maintain a competitive edge in the market.
7. Partnerships with Institutions: Campus placement systems foster partnerships between educational institutions and companies, leading to collaboration on various initiatives such as curriculum development, internships, research projects, and industry-academic exchanges.
8. Tracking Placement Trends: Educational institutions can track placement trends and outcomes to evaluate the effectiveness of their academic programs, identify areas for improvement, and make data-driven decisions to better prepare students for the job market.

Overall, campus placement and recruitment systems play a crucial role in bridging the gap between academia and industry, facilitating the transition of students from education to employment, and meeting the talent needs of organizations.

1.2 Objective:-

The main objective of the project on campus Placement and recruitment System is to manage the details of Training and Placement of Students and their eligibility criteria. It manages all the information about Placement Recruitment Cell. The purpose of the project is to build an program to reduce the manual work for managing the Training, Placement, Placement Cell of Students. It tracks all the details about the Students and their eligibility criteria.

1.3 Scope:-

It may help collecting perfect management in details. In a very short time, the collection will be obvious, simple and sensible. It will help a person to know the management of passed year perfectly. It also helps in current all works relative to campus placement recruitment System. It will be also reduced the cost of collecting the management & collection procedure will go on smoothly. The aim of project is to we have tried to computerize various processes of campus placement recruitment System

1 In computer system the person has to fill the various forms & number of copies of the forms can be easily generated at a time.

2 In computer system, it is not necessary to create the manifest but we can directly print it, which saves our time.

3 To assist the staff in capturing the effort spent on their respective working areas.

4 To utilize resources in an efficient manner by increasing their productivity.

5 The system generates types of information that can be used for various purposes.

6 It satisfy the user requirement.

7 Be easy to understand by the user and operator.

8 Be easy to operate.

9 Have a good user interface.

10 Be expandable.

Chapter 2:-System Architecture

2.1 System Design:-

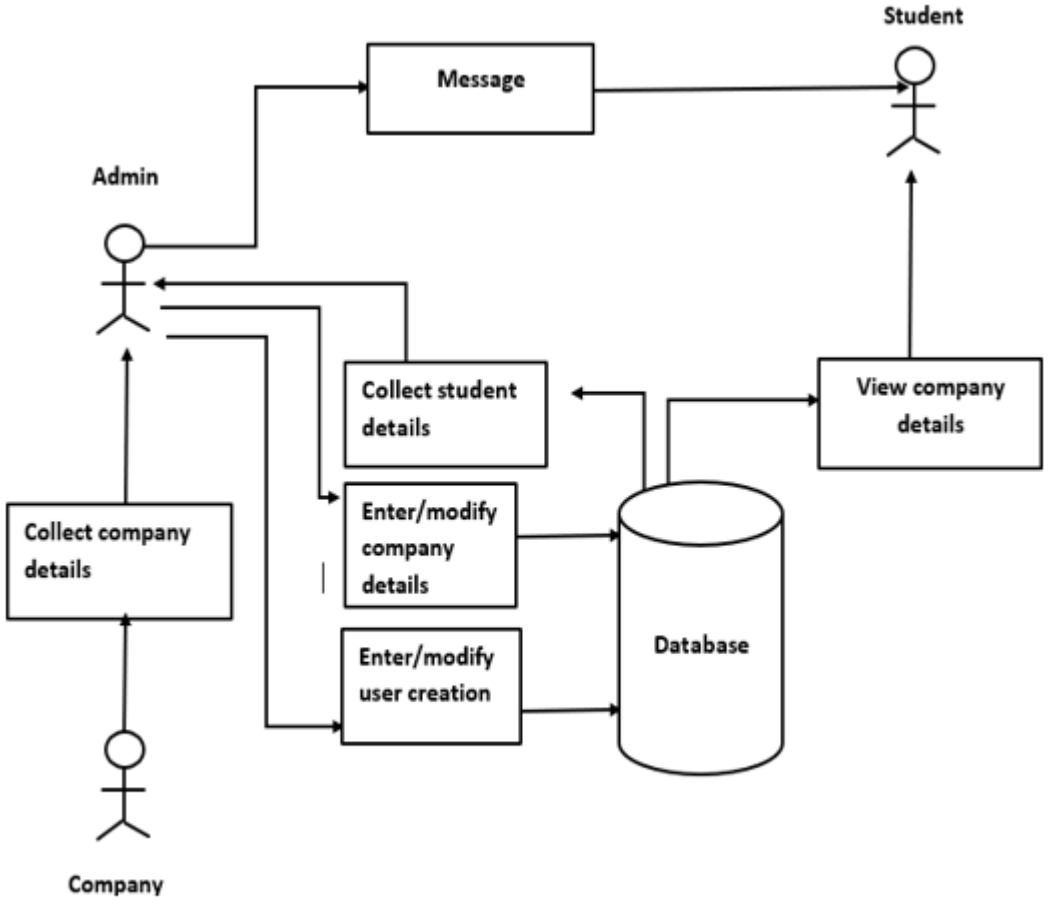
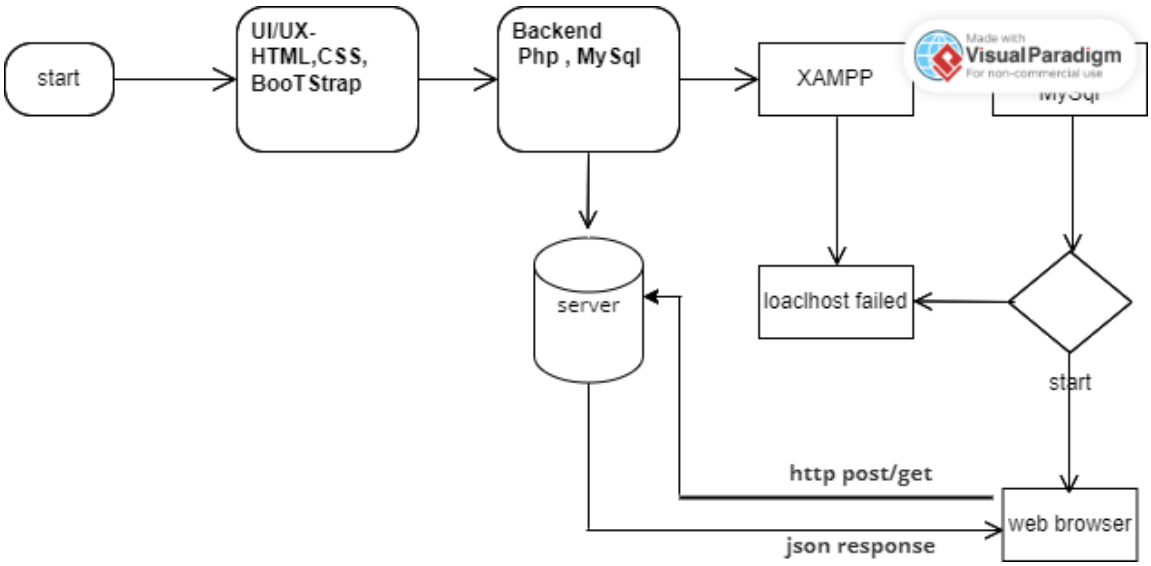


Fig 1: System Architecture



Workflow Diagram

2.2 Technological Stack:-

2.2.1 PHP programming language:-

PHP will serve as the primary programming language for developing the backend logic of the system. use PHP to handle user authentication, database interactions, business logic, and server-side processing

2.2.2 MYSQL Database:-

- MySQL is a relational database management system
- MySQL is open-source
- MySQL is free
- MySQL is ideal for both small and large applications
- MySQL is very fast, reliable, scalable, and easy to use
- MySQL is cross-platform
- MySQL is compliant with the ANSI SQL standard
- MySQL was first released in 1995
- MySQL is developed, distributed, and supported by Oracle Corporation
- MySQL is named after co-founder Monty Widenius's daughter: My.

This document describes how to create a simple web application that connects to a MySQL database server.

MySQL is a popular open source database management system commonly used in web applications due to its speed, flexibility and reliability. MySQL employs SQL, or *Structured Query Language*, for accessing and processing data contained in databases.

This tutorial is a continuation from the Connecting to a MySQL Database tutorial and assumes that you have already created a MySQL database named placement_database, which you have registered a connection for in the xampp. The table data used in that tutorial is contained in connection.sql and is also required for this tutorial. This SQL file creates two tables, Subject and Counselor, then populates them with sample data.

2.3 System Modules:-

2.3.1 User Interface Design:-

- Admin Module: Used for managing the student details.
- Placement Staff Module : Used for managing the details of Placement Cell.
- User module : Used for search,apply and view result of placement.

Designing the user interface (UI) for a campus placement and recruitment system involves creating an intuitive, user-friendly experience that caters to the needs of students, employers, and administrators. Here are some key considerations and components to include in the UI design:

1. User Roles and Authentication: Implement a login/signup interface for different user roles, such as students, employers, and administrators. Ensure secure authentication mechanisms, such as username/password or social login options, to protect user accounts.
2. Main Dashboard:- A welcoming interface displaying key functionalities (e.g., Home, campus drive, login, contact us etc.).
3. Navigation Menu: Include a clear and intuitive navigation menu to allow users to easily access different sections of the platform. Organize menu items logically, such as Home, Jobs, Events, Profile, and Help, to help users navigate the system efficiently.
4. Job Listings: Create a visually appealing and informative job listings page where students can browse and search for job opportunities. Include filters and sorting options to refine search results based on criteria like industry, location, job type, and company size.
5. Job Details: When users click on a job listing, display detailed information about the job, including the company name, job title, job description, required qualifications, application

deadline, and contact information. Allow users to apply for jobs directly from the job details page.

6. Profile Management: Enable users to manage their profiles effectively by providing options to update personal information, upload resumes, add skills and qualifications, and set preferences for job alerts and notifications.
7. Application Tracking: Implement a feature for users to track the status of their job applications. Students should be able to see which jobs they've applied for, the status of each application (e.g., pending, shortlisted, rejected), and any actions required from their end.
8. Recruitment Events: Display upcoming recruitment events, such as job fairs, career workshops, and company presentations, in a calendar or list view. Allow users to register for events, view event details, and add events to their calendars.
9. Messaging and Communication: Incorporate messaging functionality to facilitate communication between students and employers. Allow users to send messages, schedule interviews, and exchange documents securely within the platform.
10. Admin Panel: Design an intuitive admin panel for administrators to manage system settings, user accounts, job postings, event listings, and other administrative tasks. Provide tools for generating reports and analytics to track platform usage and performance.

2.3.2 Search and Navigation:-

Search Functionality:-

1. Implement Search Criteria:

Search by Student:-

Allow searching by student name, ID, batch, etc.

Search by Company:-

Enable search by company name, industry, location, etc.

Search by Date Range:-

For reports based on date-specific data (e.g., placement dates).

2. User Interface Elements:-

Search Bar:

Provide a dedicated search bar at the top of the report section for users to enter search queries.

Filtering Options:-

Dropdowns, checkboxes, or radio buttons for refining search criteria (e.g., date range selectors, student/company selectors).

3. Backend Implementation:-

Implement backend logic to process search queries and fetch relevant data from the database.
Use SQL queries or ORM (Object-Relational Mapping) techniques to retrieve filtered data.

4. Display Search Results:-

Display search results in a clear and organized manner in the report section's display area.
Update the displayed reports dynamically as users apply search filters or keywords.

Navigation:-

1. Menu and Navigation Bar:-

Sidebar or Top Navigation Bar:

Clearly labeled options for different types of reports (placement statistics, student placement reports, etc.).

Breadcrumb Navigation:-

Use breadcrumb trails to show users their current location within the report section.

2. Report Category Selection:-

Allow users to select report categories from a dropdown or list, making it easy to navigate to a specific type of report.

3. Clickable Elements:-

Make report titles or categories clickable to navigate users directly to specific reports or sections.

4. Back and Forward Functionality:-

Include navigation options that allow users to move back to the previous report or category and forward again if needed.

5. Responsive Design:-

Ensure navigation elements are responsive and adaptable for different screen sizes and devices.

6. Logical Flow:-

Design the navigation to follow a logical flow, guiding users from broader categories to specific reports with ease.

Integration:-

1. Backend-UI Integration:-

Ensure the search functionality and navigation elements are seamlessly integrated into the user interface using Php backend logic and frontend components (HTML,CSS,JAVA SCRIPT).

2. Testing:-

Thoroughly test the search and navigation functionalities to ensure they work smoothly, handle edge cases, and provide accurate results.

Implementing these search and navigation features will enhance the user experience and efficiency of navigating through various reports in your Training and Placement Management System.

2.4 Hardware and Software Requirements:-

Software Requirements:-

Hardware Requirements:-

Processor : Intel 5

Installed memory (RAM) : 4 GB

Hard Disk : 500 GB

Operating System : Windows

Software Requirements: -

Front End: HTML5, CSS3, Bootstrap

Back End: PHP 8.1, MYSQL

Control End: Angular Java Script

Android Tools:

IDE: Android Studio

Android Emulator

xampp-win64-8.1

PHP Tools:

xampp-win64-8.1

Chapter 3:- Implementation:-

3.1 Development Environment Setup:-

The Implementation chapter commences with a crucial section dedicated to configuring the development environment, with a primary focus on the VS code and XAMPP SERVER for the coding and building phases of the Campus Placement recruitment system.

3.1.1 VS CODE Installation:-

Download VS Code:

Visit the Visual Studio Code website.

Click on the "Download for Windows" button.

Run the Installer:

Once the download is complete, locate the downloaded file (usually in your Downloads folder).

Double-click on the installer file (e.g., VSCodeSetup.exe) to run it.

Follow the Installation Wizard:

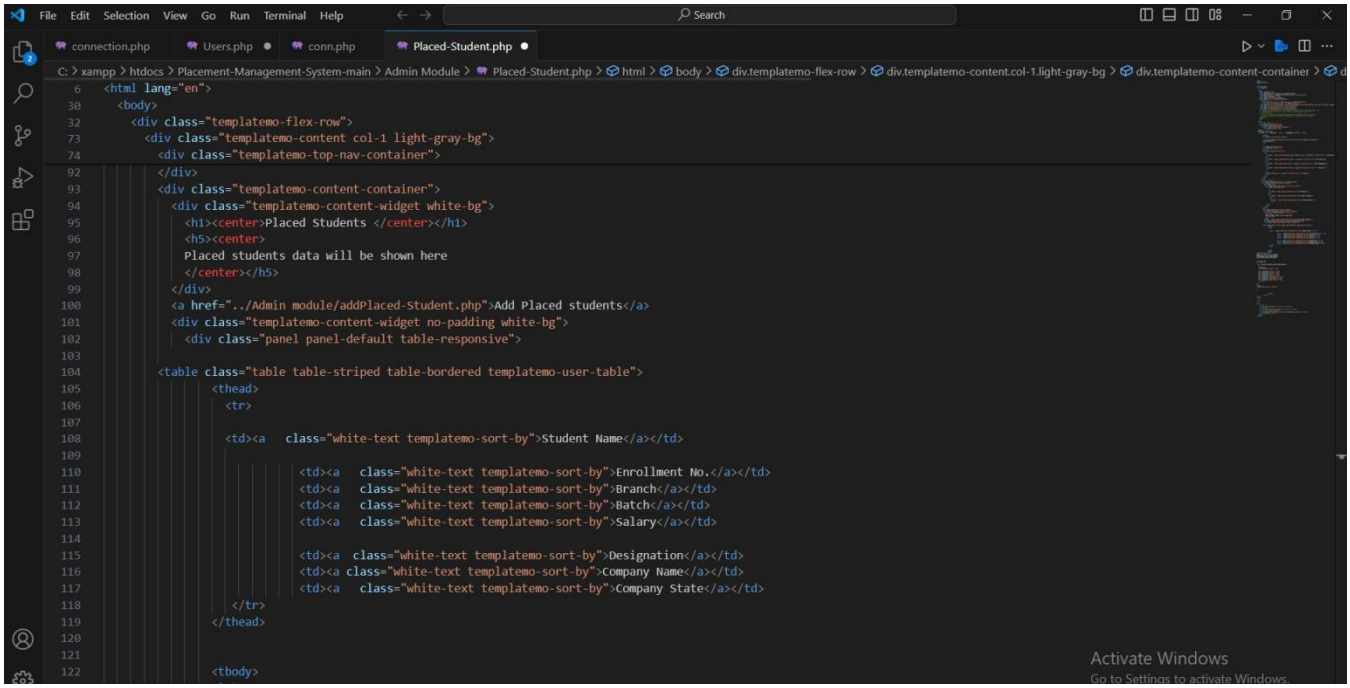
The installation wizard will guide you through the installation process.

Choose your desired installation location and any additional options you want to customize.

Click "Next" or "Install" to begin the installation process.

Complete the Installation:

Once the installation is complete, you can launch VS Code by double-clicking its shortcut on the desktop or searching for it in the Start menu.



3.1.2 XAMPP INSTALLATION:-

Download XAMPP:

Visit the XAMPP website.

Choose the appropriate version of XAMPP for your operating system .

Click on the download button to download the installer.

Run the Installer:

Once the download is complete, locate the downloaded installer file.

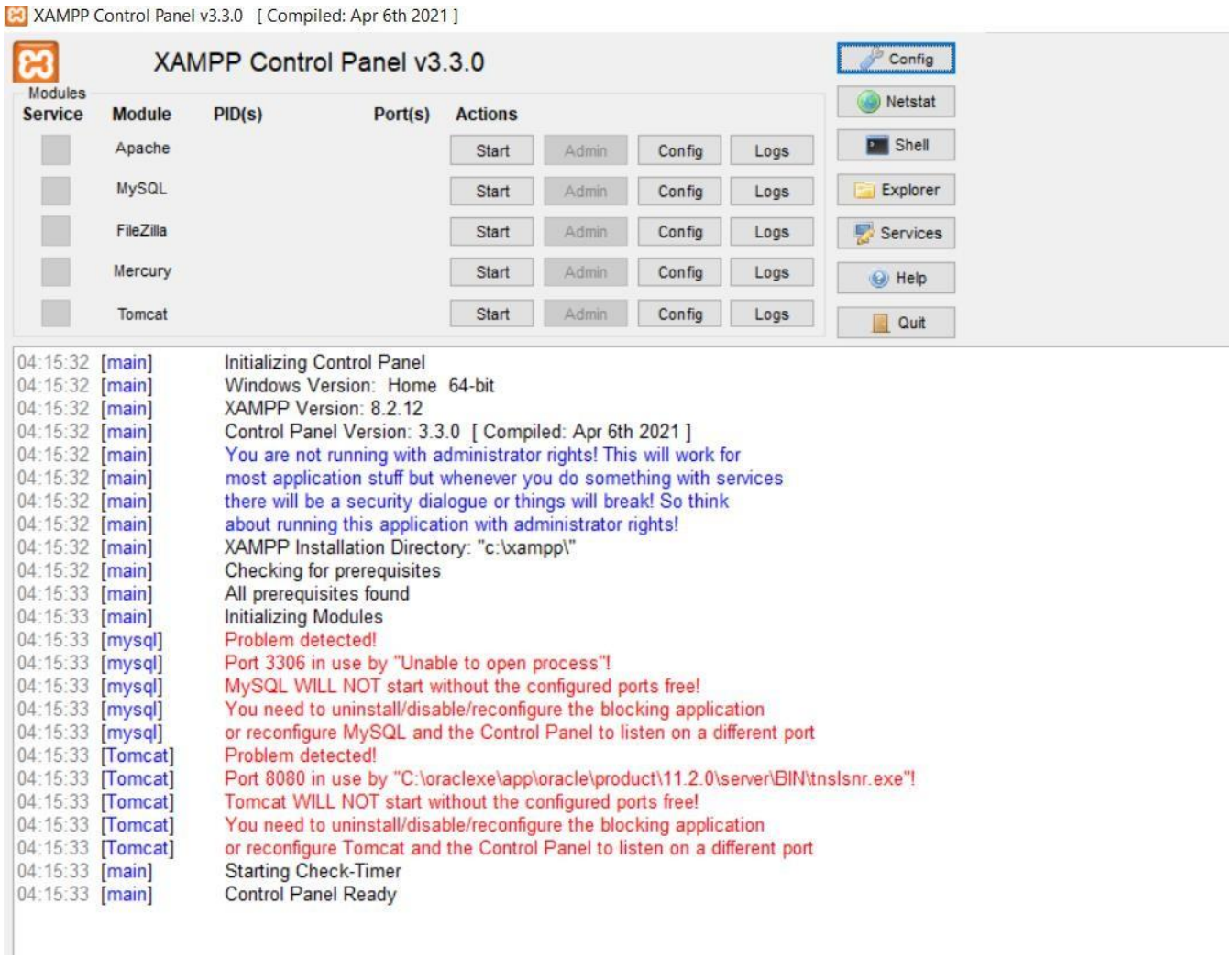
Double-click on the installer file to run it.

Follow the installation wizard instructions.

Choose Components:

During the installation process, you'll be prompted to select the components you want to install. By default, XAMPP installs Apache, MySQL, PHP, and phpMyAdmin..

Complete the Installation:



3.2 MYSQL Server Configuration:-

Installation:-

Download MySQL Server:-

Visit the official MySQL website or repository to download the MySQL Server I installer suitable for your operating system.

Installation Process:-

Run the installer and follow the on-screen instructions.

Choose installation type (typical/custom) and configure settings as needed (port number, installation directory, etc.).

Initial Configuration:

Starting the MySQL Server:-

After installation, start the MySQL server service. This can be done using services (Windows) or commands .

Secure the MySQL Installation:-

Run the security script provided by MySQL to set a root password and improve Security setting.

Command for MySQL 5.7 and later: mysql_secure_installation

PHP MYADMIN:

Start the mysql and click on admin go to the phpadmin page.

Create database for php project.

Key Configuration Parameters:-

Port Configuration: By default, MySQL uses port 3306. You can modify this in the configuration file if needed.

Project Setup:

Start:

Start apache and mysql

Place Your PHP Project Files:

Place your PHP project files in the htdocs directory within your XAMPP installation directory. This directory is the document root for Apache, meaning that PHP files placed here can be accessed via the web browser.

Access Your PHP Project:

Open a web browser and navigate to http://localhost/your_project_folder to access your PHP project.

Replace your_project_folder with the name of the directory containing your PHP project files.

3.3 Database Schema:-

3.3.1 Data Flow Diagram:-

The Data Flow Diagram shows the flow of data or information. It can be partitioned into single processes or functions. Data Flow Diagrams can be grouped together or decomposed into multiple processes. There can be physical DFD's that represent the physical files and transactions, or they can be business DFD's (logical, or conceptual). The DFD is an excellent communication tool for analysts to model processes and functional requirements. One of the primary tools of the structured analysis efforts of the 1970's it was developed and enhanced by the likes of Yourdon, McMenamin, Palmer, Gane and Sarson. It is still considered one of the best modeling techniques for eliciting and representing the processing requirements of a system.

Used effectively, it is a useful and easy to understand modeling tool. It has broad application and usability across most software development projects. It is easily integrated with data modeling, workflow modeling tools, and textual specs. Together with these, it provides analysts and developers with solid models and specs. Alone, however, it has limited usability. It is simple and easy to understand by users and can be easily extended and refined with further specification into a physical version for the design and development teams.

The different versions are Context Diagrams (Level 0), Partitioned Diagrams (single process only -- one level), functionally decomposed, leveled sets of Data Flow Diagrams.

Data Store

It is a repository of information. In the physical model, this represents a file, table, etc. In the logical model, a data store is an object or entity.

Data Flows

DFDs show the flow of data from external entities into the system, showed how the data moved from one process to another, as well as its logical storage. There are only four symbols:

- Squares representing **external entities**, which are sources or destinations of data.
- Rounded rectangles representing **processes**, which take data as input, do something to it, and output it.
- Arrows representing the **data flows**, which can either, be electronic data or physical items.
- Open-ended rectangles representing **data stores**, including electronic stores such as databases or XML files and physical stores such as or filing cabinets or stacks of paper.

There are several common modeling rules for creating DFDs:

- All processes must have at least one data flow in and one data flow out.
- All processes should modify the incoming data, producing new forms of outgoing data.
- Each data store must be involved with at least one data flow.
- Each external entity must be involved with at least one data flow.
- A data flow must be attached to at least one process.

DFDs are nothing more than a network of related system functions and indicate from where information is received and to where it is sent. It is the starting point in the system that decomposes the requirement specifications down to the lowest level detail.

The four symbols in DFD, each of which has its meaning. They are given below:

- External entities are outside to system but they either supply input data in the system or use the system output. These are represented by square of rectangle. External entities that supply data into a system are sometimes called Sources. External entities that use system data are sometimes called sinks.
- Dataflow models that passage of data in the system and are represented by line by joining system components. An arrow indicates the direction of the flow, and the line is labeled by the name of the dataflow.
- Process show that the systems do. Each process has one or more data inputs and one or data outputs. Circles in DFD represent them. Each high-level process may be consisting of more than one lower-level processes. Process will be expanded in sequent level DFD. A circle or a bubble represents a process that transforms incoming data flow into outgoing dataflow.

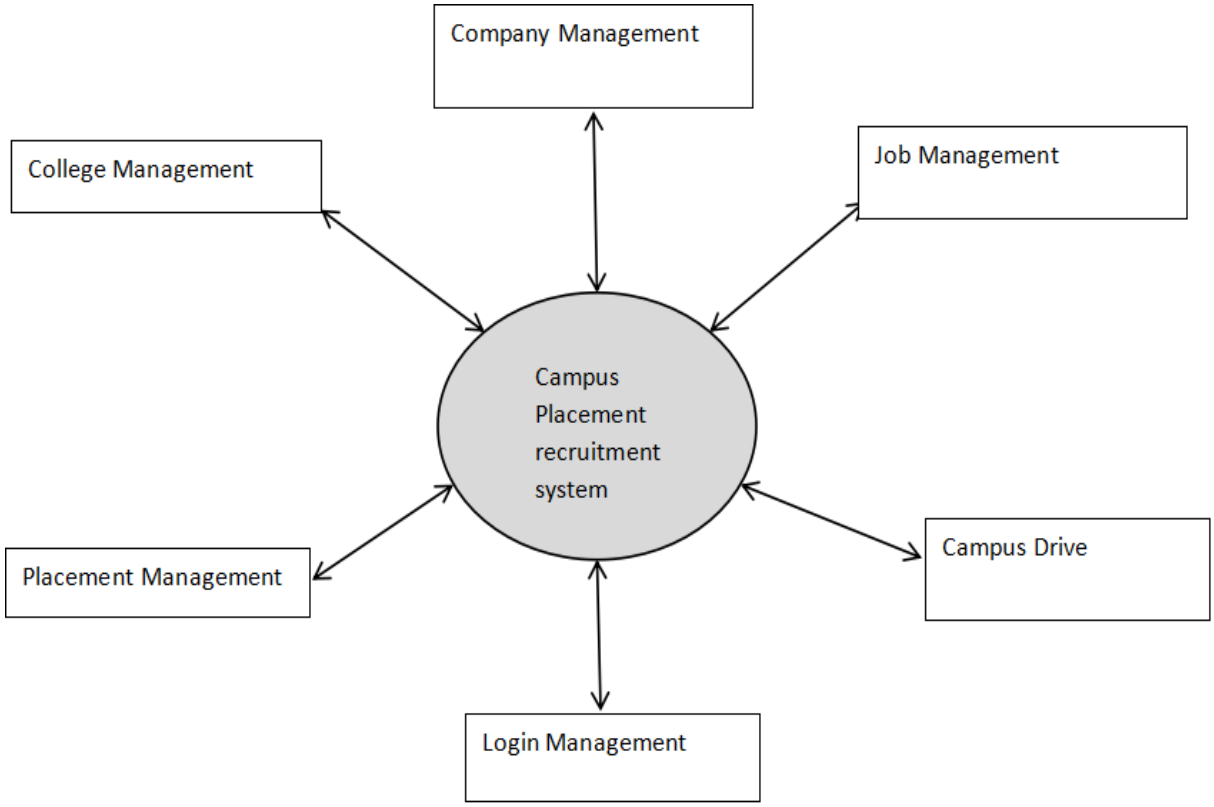
  The high-level processes in a system are:

  - Receivable process.
  - Verifiable process.
  - Disposal process.

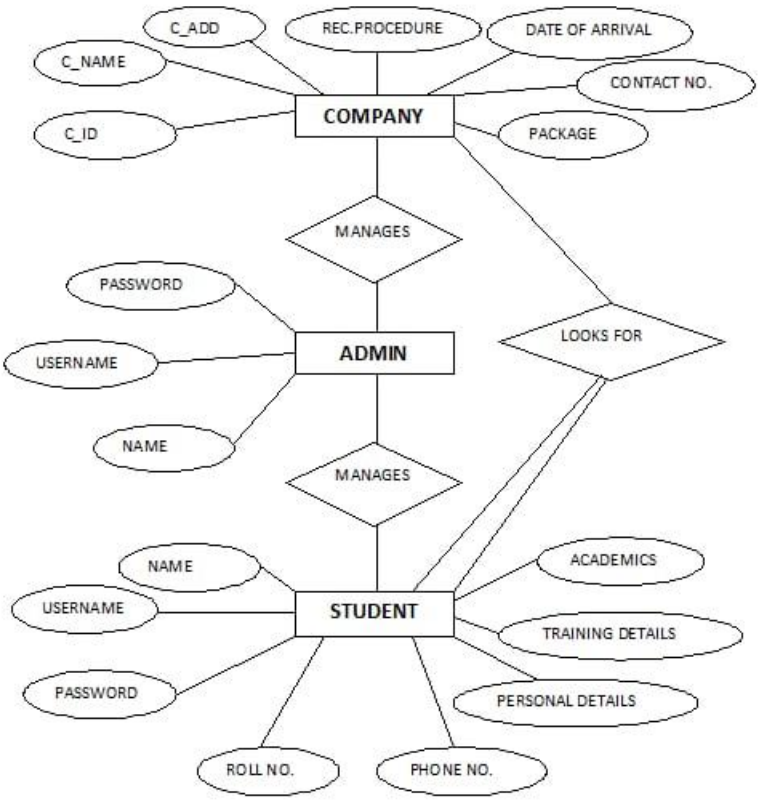
- File or data store is a repository of data. They contain data that is retained in the system. Process can enter data into data store or retrieved data from the data store. An open rectangle is a data store, data at rest.

3.3.2 DFD Diagrams:-

0-Level DFD:-



Zero Level DFD-Campus Placement Recruitment System



Campus Placement and
Recruitment system DFD

3.3.3 Entity Relation Model:-

Simply stated the ER model is a conceptual data model that views the real world as entities and relationships. A basic component of the model is the Entity-Relationship diagram which is used to visually represent data objects. Since Chen wrote his paper the model has been extended and today it is commonly used for database.

Basic Constructs of E-R Modeling

The ER model views the real world as a construct of entities and association between entities.

Entities

Entities are the principal data object about which information is to be collected. Entities are classified as independent or dependent (in some methodologies, the terms used are strong and weak, respectively). An independent entity is one that does not rely on another for identification. A dependent entity is one that relies on another for identification.

Relationships

A Relationship represents an association between two or more entities. Relationships are classified in terms of degree, connectivity, cardinality, and existence.

Attributes

Attributes describe the entity of which they are associated. A particular instance of an attribute is a value. The domain of an attribute is the collection of all possible values an attribute can have. The domain of Name is a character string.

Classifying Relationships

Relationships are classified by their degree, connectivity, cardinality, direction, type, and existence. Not all modeling methodologies use all these classifications.

Degree of a Relationship

The degree of a relationship is the number of entities associated with the relationship. The n-array relationship is the general form for degree n. Special cases are the binary, and ternary, where the degree is 2 and 3 respectively.

Connectivity and Cardinality

The connectivity of a relationship describes the mapping of associated entity instances in the relationship. The values of connectivity are "one" or "many". The cardinality of a relationship is the actual number of related occurrences for each of the two entities. The basic types of connectivity for relations are: one-to-one, one-to-many, and many-to-many.

Direction

The direction of a relationship indicates the originating entity of a binary relationship. The entity from which a relationship originates is the parent entity; the entity where the relationship terminates is the child entity.

The direction of a relationship is determined by its connectivity type. An identifying relationship is one in which one of the child entities is also a dependent entity. A non-identifying relationship is one in which both entities are independent.

Existence

Existence denotes whether the existence of an entity instance is dependent upon the existence of another, related, entity instance. The existence of an entity in a relationship is defined as either mandatory or optional.

Generalization Hierarchies

A generalization hierarchy is a form of abstraction that specifies that two or more entities that share common attributes can be generalized into a higher-level entity type called a supertype or generic entity. The lower level of entities become the subtype, or categories, to the supertype. Subtypes are dependent entities.

ER Notation

The symbols used for the basic ER constructs are:

a) Entities are represented by labeled rectangles. The label is the name of the entity.

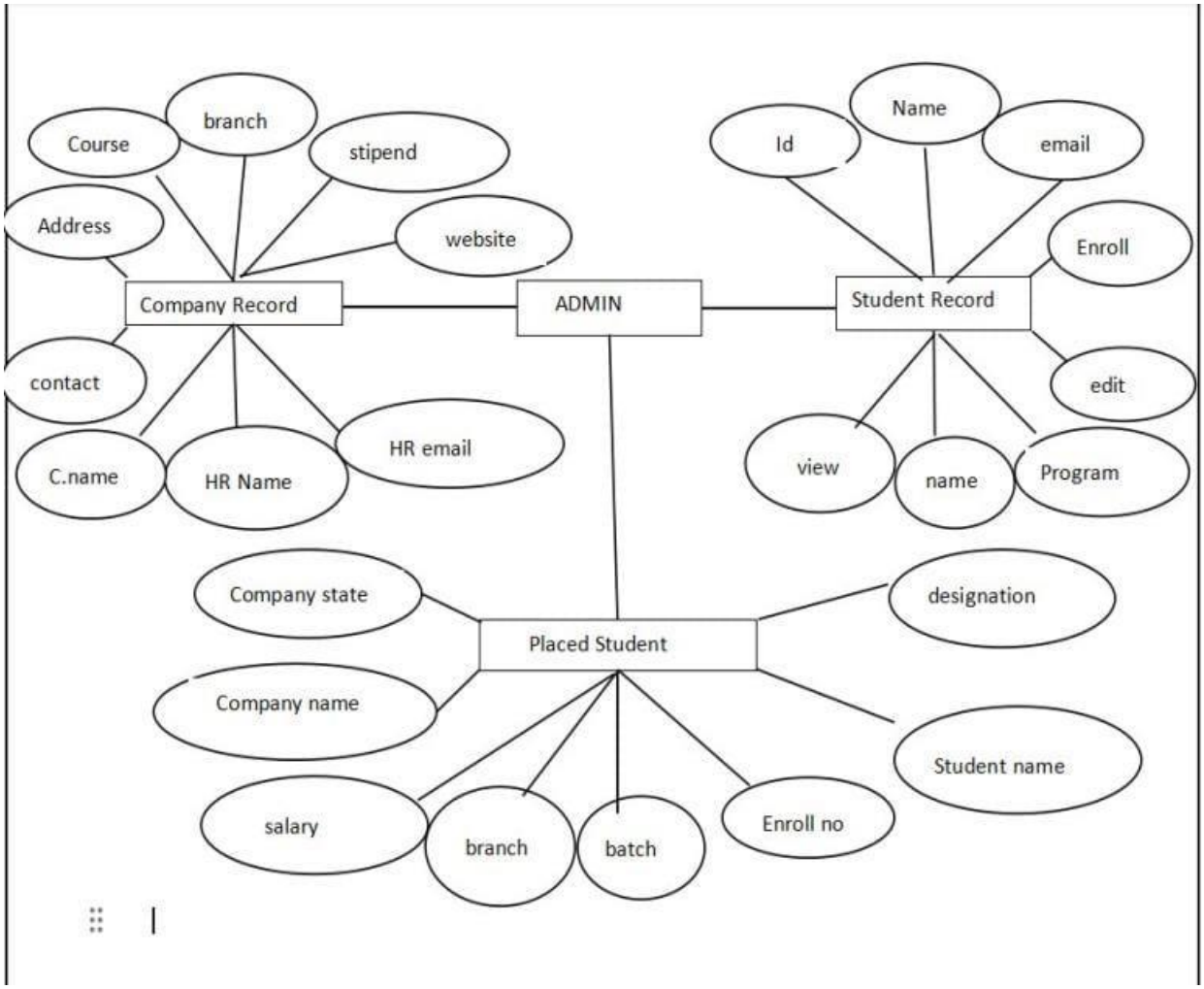
b) Relationships are represented by a solid line connecting two entities. The name of the relationship is written above the line. Relationship names should be verbs.

c) Attributes, when included, are listed inside the entity rectangle. Attributes which are identifiers are underlined. Attribute names should be singular nouns.

d) Cardinality of many is represented by a line ending in a crow's foot. If the crow's foot is omitted, the cardinality is one.

e) Existence is represented by placing a circle or a perpendicular bar on the line. Mandatory existence is shown by the bar (looks like a 1) next to the entity for an instance is required. Optional existence is shown by placing a circle next to the entity that is optional.

f) Existence is represented by placing a circle or a perpendicular bar on the line. Mandatory existence is shown by the bar (looks like a 1) next to the entity for an instance is required. Optional existence is shown by placing a circle next to the entity


Course
branch
stipend
Address
website
Company Record
ADMIN
Student Record
Id
Name
email
Enroll
contact
edit
C.name
HR Name
HR email
view
name
Program
Company state
designation
Placed Student
Company name
Student name
salary
branch
batch
Enroll no

ERD DIAGRAM

3.4 Database Design:-

The general theme behind a database is to handle information as an integrated whole. A database is a collection of inter-related data stored with minimum redundancy to serve single users quickly and efficiently. The general objective is to make information necessary, quick, inexpensive, and flexible for the user.

3.4.1 Database Tables:-

ADMIN :-

Field	Type	Null	Key	length
Username	varchar	No	Primary	24
Password	varchar	No	Primary	24

REGISTRATION:-

Field	Type	Null	Key	length
First name	varchar	No	Primary	200
Middle name	varchar	No		200
Last name	varchar	No		200
Email	varchar	No		200
DOB	int	No		16
Phone No	int	No		10
Gender	varchar	No		200
City	varchar	No		200
Street	varchar	No		200
Username	varchar	No		200
Password	varchar	No		200

TOP'S PAGE:-

Field	Type	Null	Key	length
Company id	int	No	Primary	16
Company name	varchar	No		200
Type	varchar	No		200
Schedule id	int	No		16
Schedule name	varchar	No		200
Criteria	varchar	No		200
Minimum salary	int	No		16
Appearing date	int	No		16
Department	varchar	No		200

COMPANY DETAILS:-

Field	Type	Null	Key	length
Company name	varchar	No	Primary	200
Type	varchar	No		200
HR name	varchar	No		200
Email	varchar	No		200
Phone No	int	No		16
City	varchar	No		200

3.4 Source Code:-

ADMIN:

Myprofile:

```
<?php
session_start();
 ?>
<!DOCTYPE html>
<html lang="en">

  <head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>My Profile</title>
    <meta name="description" content="">
    <meta name="author" content="templatemo">
    <!--favicon-->
        <link rel="shortcut icon" href="favicon.ico" type="image/icon">
        <link rel="icon" href="favicon.ico" type="image/icon">
    <link href='http://fonts.googleapis.com/css?family=Open+Sans:400,300,400italic,700' rel='stylesheet' type='text/css'>
    <link href="css/font-awesome.min.css" rel="stylesheet">
    <link href="css/bootstrap.min.css" rel="stylesheet">
    <link href="css/templatemo-style.css" rel="stylesheet">
    <!-- HTML5 shim and Respond.js for IE8 support of HTML5 elements and media queries -->
    <!-- WARNING: Respond.js doesn't work if you view the page via file:// -->
    <!--[if lt IE 9]>
      <script src="https://oss.maxcdn.com/html5shiv/3.7.2/html5shiv.min.js"></script>
      <script src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script>
    <![endif]-->
  </head>

  <body>
    <!-- Left column -->
    <div class="templatemo-flex-row">
      <div class="templatemo-sidebar">
        <header class="templatemo-site-header">
          <div class="square"></div>
  <?php
  $Welcome = "Welcome";
      echo "<h1>" . $Welcome . "<br>".   $_SESSION['username']. "</h1>";
  ?>
        </header>
        <div class="profile-photo-container">
```

```
    <img src="images/profile-photo.png" alt="Profile Photo" class="img-responsive">
  <h2>Admin</h2>
  </div>


  <div class="mobile-menu-icon">
    <i class="fa fa-bars"></i>
  </div>
  <nav class="templatemo-left-nav">
    <ul>
      <!-- <li>
        <a href="../Admin module/dashboard.php" class="active"><i class="fa fa-home fa-fw"></i>Dashboard</a>
      </li> -->
      <li>
        <a href="../Admin module/myprofile.php"><i class="fa fa-user fa-fw"></i>My Profile</a>
      </li>
      <li>
        <a href="../Admin module/Users.php"><i class="fa fa-users fa-fw"></i>User Management</a>
      </li>
      <li>
        <a href="../Admin module/Records.php"><i class="fa fa-bar-chart fa-fw"></i>Records</a>
      </li>

      <li>
      <a href="logout.php"><i class="fa fa-eject fa-fw"></i>Logout</a>
      </li>
    </ul>
  </nav>
</div>
<!-- Main content -->
<div class="templatemo-content col-1 light-gray-bg">
  <div class="templatemo-top-nav-container">
    <div class="row">
      <nav class="templatemo-top-nav col-lg-12 col-md-12">
        <ul class="text-uppercase">
          <li>
            <a href="">Home</a>
          </li>
          <li>
            <a href="../Admin module/WriteNotification.php">Notifications</a>
```

Placed student:

```php
<?php
session_start();
 ?>

<!DOCTYPE html>
<html lang="en">

  <head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Placed Students</title>
    <meta name="description" content="">
    <meta name="author" content="templatemo">
    <!--favicon-->
        <link rel="shortcut icon" href="favicon.ico" type="image/icon">
        <link rel="icon" href="favicon.ico" type="image/icon">
    <link href='http://fonts.googleapis.com/css?family=Open+Sans:400,300,400italic,700' rel='stylesheet' type='text/css'>
    <link href="css/font-awesome.min.css" rel="stylesheet">
    <link href="css/bootstrap.min.css" rel="stylesheet">
    <link href="css/templatemo-style.css" rel="stylesheet">
    <!-- HTML5 shim and Respond.js for IE8 support of HTML5 elements and media queries -->
    <!-- WARNING: Respond.js doesn't work if you view the page via file:// -->
    <!--[if lt IE 9]>
      <script src="https://oss.maxcdn.com/html5shiv/3.7.2/html5shiv.min.js"></script>
      <script src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script>
    <![endif]-->
  </head>

  <body>
    <!-- Left column -->
    <div class="templatemo-flex-row">
      <div class="templatemo-sidebar">
        <header class="templatemo-site-header">
          <div class="square"></div>
  <?php
  $Welcome = "Welcome";
      echo "<h1>" . $Welcome . "<br>".   $_SESSION['username']. "</h1>";
  ?>
        </header>
        <div class="profile-photo-container">
```

```
    <img src="images/profile-photo.png" alt="Profile Photo" class="img-responsive">
  <h2>Firstname</h2>
  </div>


  <div class="mobile-menu-icon">
    <i class="fa fa-bars"></i>
  </div>
  <nav class="templatemo-left-nav">
    <ul>
      <li>
        <a href="../Admin module/dashboard.php" class="active"><i class="fa fa-home fa-fw"></i>Dashboard</a>
      </li>
      <li>
        <a href="../Admin module/myprofile.php"><i class="fa fa-user fa-fw"></i>My Profile</a>
      </li>
      <li>
        <a href="../Admin module/Users.php"><i class="fa fa-users fa-fw"></i>User Management</a>
      </li>
      <li>
        <a href="../Admin module/Records.php"><i class="fa fa-bar-chart fa-fw"></i>Records</a>
      </li>

      <li>
      <a href="logout.php"><i class="fa fa-eject fa-fw"></i>Logout</a>
      </li>
    </ul>
  </nav>
</div>
<!-- Main content -->
<div class="templatemo-content col-1 light-gray-bg">
  <div class="templatemo-top-nav-container">
    <div class="row">
      <nav class="templatemo-top-nav col-lg-12 col-md-12">
        <ul class="text-uppercase">

          <li>
            <a href="../Admin module/Student-Record.php">Students</a>
          </li>
          <li>
            <a href="../Admin module/Placed-Student.php">Placed students</a>
```

```
            <li>
              <a href="../Admin module/Company-Record.php">Company Records </a>
              </li>

          </ul>
        </nav>
      </div>
    </div>
    <div class="templatemo-content-container">
      <div class="templatemo-content-widget white-bg">
        <h1><center>Placed Students </center></h1>
        <h5><center>
        Placed students data will be shown here
        </center></h5>
      </div>
      <a href="../Admin module/addPlaced-Student.php">Add Placed students</a>
      <div class="templatemo-content-widget no-padding white-bg">
        <div class="panel panel-default table-responsive">

    <table class="table table-striped table-bordered templatemo-user-table">
            <thead>
              <tr>

              <td><a   class="white-text templatemo-sort-by">Student Name</a></td>

                          <td><a   class="white-text templatemo-sort-by">Enrollment No.</a></td>
                          <td><a   class="white-text templatemo-sort-by">Branch</a></td>
                          <td><a   class="white-text templatemo-sort-by">Batch</a></td>
                          <td><a   class="white-text templatemo-sort-by">Salary</a></td>

                          <td><a  class="white-text templatemo-sort-by">Designation</a></td>
                          <td><a class="white-text templatemo-sort-by">Company Name</a></td>
                          <td><a   class="white-text templatemo-sort-by">Company State</a></td>
               </tr>
            </thead>


            <tbody>
            <?php
```

```
include('conn.php');
$query = "SELECT * FROM record";
$data=mysqli_query($db,$query);
$total=mysqli_num_rows($data);


if($total!=0)
{
  while($results=mysqli_fetch_assoc($data))
  {
    echo"<tr>
  <td>".$results['name']."</td>

  <td>".$results['enroll']."</td>
  <td>".$results['branch']."</td>
  <td>".$results['batch']."</td>
  <td>".$results['salary']."</td>
  <td>".$results['designation']."</td>
  <td>".$results['comp_name']."</td>
  <td>".$results['comp_state']."</td>
    </tr>";
  }
}
else {
  echo "Table has no records";
}

 ?>

                </tbody>
          </table>
</div>
</div>

</div>

</div>
    <!-- JS -->
    <script src="js/jquery-1.11.2.min.js"></script>
    <!-- jQuery -->
    <script src="js/jquery-migrate-1.2.1.min.js"></script>
    <!-- jQuery Migrate Plugin -->
```

Student Record:

```php
<?php
session_start();
 ?>

<!DOCTYPE html>
<html lang="en">

  <head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Users</title>
    <meta name="description" content="">
    <meta name="author" content="templatemo">
    <!--favicon-->
        <link rel="shortcut icon" href="favicon.ico" type="image/icon">
        <link rel="icon" href="favicon.ico" type="image/icon">
    <link href='http://fonts.googleapis.com/css?family=Open+Sans:400,300,400italic,700' rel='stylesheet' type='text/css'>
    <link href="css/font-awesome.min.css" rel="stylesheet">
    <link href="css/bootstrap.min.css" rel="stylesheet">
    <link href="css/templatemo-style.css" rel="stylesheet">
    <!-- HTML5 shim and Respond.js for IE8 support of HTML5 elements and media queries -->
    <!-- WARNING: Respond.js doesn't work if you view the page via file:// -->
    <!--[if lt IE 9]>
      <script src="https://oss.maxcdn.com/html5shiv/3.7.2/html5shiv.min.js"></script>
      <script src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script>
    <![endif]-->
  </head>

  <body>
    <!-- Left column -->
    <div class="templatemo-flex-row">
      <div class="templatemo-sidebar">
        <header class="templatemo-site-header">
          <div class="square"></div>
  <?php
  $Welcome = "Welcome";
      echo "<h1>" . $Welcome . "<br>".   $_SESSION['username']. "</h1>";
  ?>
        </header>
        <div class="profile-photo-container">
```

```
    <img src="images/profile-photo.png" alt="Profile Photo" class="img-responsive">
  <h2>Firstname</h2>
  </div>


  <div class="mobile-menu-icon">
    <i class="fa fa-bars"></i>
  </div>
  <nav class="templatemo-left-nav">
    <ul>
      <!-- <li>
        <a href="../Admin module/dashboard.php" class="active"><i class="fa fa-home fa-fw"></i>Dashboard</a>
      </li> -->
      <li>
        <a href="../Admin module/myprofile.php"><i class="fa fa-user fa-fw"></i>My Profile</a>
      </li>
      <li>
        <a href="../Admin module/Users.php"><i class="fa fa-users fa-fw"></i>User Management</a>
      </li>
      <li>
        <a href="../Admin module/Records.php"><i class="fa fa-bar-chart fa-fw"></i>Records</a>
      </li>

      <li>
      <a href="logout.php"><i class="fa fa-eject fa-fw"></i>Logout</a>
      </li>
    </ul>
  </nav>
</div>
<!-- Main content -->
<div class="templatemo-content col-1 light-gray-bg">
  <div class="templatemo-top-nav-container">
    <div class="row">
      <nav class="templatemo-top-nav col-lg-12 col-md-12">
        <ul class="text-uppercase">

          <li>
            <a href="../Admin module/Student-Record.php">Students</a>
          </li>
          <li>
            <a href="../Admin module/Company-Record.php">Company</a>
```

```
            <li>
              <a href="../Admin module/Company-Record.php">Company</a>
              </li>
              <li>
                <a href="../Admin module/Placed-Student.php">PLaced Students</a>
                </li>
          </ul>
        </nav>
      </div>
    </div>
    <div class="templatemo-content-container">
      <div class="templatemo-content-widget white-bg">
        <h1><center>Students Records </center></h1>
      </div>
      <div class="templatemo-content-widget no-padding white-bg">
        <div class="panel panel-default table-responsive">


    <table class="table table-striped table-bordered templatemo-user-table">
            <thead>
              <tr>
                <td><a class="white-text templatemo-sort-by">Student name</a></td>
                <td><a   class="white-text templatemo-sort-by">Enrollment No</a></td>
                <td><a   class="white-text templatemo-sort-by">Unique Id </a></td>

                <td><a class="white-text templatemo-sort-by">Email </a></td>
                <td><a  class="white-text templatemo-sort-by">Program</a></td>


                <td><a  class="white-text templatemo-sort-by">View </a></td>
                <td><a class="white-text templatemo-sort-by">Edit </a></td>
                <!-- <td><a class="white-text templatemo-sort-by">Delete</a></td> -->
               </tr>
            </thead>


            <tbody>
              <?php
              include '../connection.php';
              $q1 = "SELECT * FROM `student_record` inner join users on student_record.usn=users.username where users.status='Approved'";
              $datas=mysqli_query($db,$q1);
              $totals=mysqli_num_rows($datas);



               if($totals!=0)
               {
                 while($result=mysqli_fetch_assoc($datas))
                 {
?>
                  <tr>
                  <td><?php  echo $result['fname']." ".$result['lname']?></td>
                  <td><?php  echo $result['enroll_no']?></td>
                  <td><?php  echo $result['usn']?></td>

                  <td><?php  echo $result['email'] ?></td>
                  <td><?php  echo $result['program']?></td>

                  <td> <a href="viewsr.php?id=<?php echo $result['usn'];?>" class="btn btn-primary">View</a></td>
                  <td> <a href="updatesr.php?id=<?php echo $result['usn'];?>" class="btn btn-success">Edit</a></td>
                  <!-- <td><a href="delelerecords.php?id=<?php echo $result['usn'];?>" onclick="return checkdelete()" class="btn btn-danger">Delete</a></td> -->
                  </tr>
                  <?php

                 }
               }
               else {
                 echo "Table has no records";
               }

                ?>

              </tbody>
        </table>

</div>
</div>

</div>

</div>
    <!-- JS -->
    <script src="js/jquery-1.11.2.min.js"></script>
    <!-- jQuery -->
    <script src="js/jquery-migrate-1.2.1.min.js"></script>
    <!-- jQuery Migrate Plugin -->
```


View Student Record:

```php
<?php
include '../connection.php';
$usn=$_GET['id'];
$sql="Select * from student_record where usn='$usn'";
$data=mysqli_query($db,$sql);
$res=mysqli_fetch_assoc($data);
/*$res['fname']
$res['lname']
$res['enroll_no']
$res['usn']
$res['phn']
*/
 ?>



<!DOCTYPE html>
<html lang="en">
<head>

  <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Update</title>
      <meta name="description" content="">
      <meta name="author" content="templatemo">
  <!--favicon-->
      <link rel="shortcut icon" href="favicon.ico" type="image/icon">
      <link rel="icon" href="favicon.ico" type="image/icon">
    <link href='http://fonts.googleapis.com/css?family=Open+Sans:400,300,400italic,700' rel='stylesheet' type='text/css'>
    <link href="css/font-awesome.min.css" rel="stylesheet">
    <link href="css/bootstrap.min.css" rel="stylesheet">
    <link href="css/templatemo-style.css" rel="stylesheet">
      <!-- Footer -->

    <!-- HTML5 shim and Respond.js for IE8 support of HTML5 elements and media queries -->
    <!-- WARNING: Respond.js doesn't work if you view the page via file:// -->
    <!--[if lt IE 9]>
      <script src="https://oss.maxcdn.com/html5shiv/3.7.2/html5shiv.min.js"></script>
      <script src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script>
    <![endif]-->
</head>
<
```

```php
<body class="light-gray-bg">
  <div class="templatemo-content-widget templatemo-login-widget white-bg">



       <form action="myprofile(php).php" class="templatemo-login-form" method="post" enctype="multipart/form-data">
         <div class="row form-group" >

           <div class="row">
               <div class="col-md-12">
                 <img  class="mb-1" src="<?php echo "../".$res['profile_pic']?>" width="200px"; height="200px"; style="margin-left:80px; margin-bottom: 30px ;border-radius:50%" >


           </div>
           </div>
                                <div class="row">
                                    <div class="col-md-6">
                                        <label>Full name</label>
                                    </div>
                                    <div class="col-md-6">
                                        <p><?php echo $res['fname']." ".$res['lname']?></p>
                                    </div>
                                </div>
                                <div class="row">
                                    <div class="col-md-6">
                                        <label>Enrollment No</label>
                                    </div>
                                    <div class="col-md-6">
                                        <p><?php echo $res['enroll_no']?></p>
                                    </div>
                                </div>
                                <div class="row">
                                    <div class="col-md-6">
                                        <label>Unique Id.</label>
                                    </div>
                                    <div class="col-md-6">
                                        <p><?php echo $res['usn']?></p>
```

```
<div class="row">
    <div class="col-md-6">
        <label>Unique Id.</label>
    </div>
    <div class="col-md-6">
        <p><?php echo $res['usn']?></p>
    </div>
</div>
<div class="row">
    <div class="col-md-6">
        <label>Phone</label>
    </div>
    <div class="col-md-6">
        <p><?php echo $res['phn_no']?></p>
    </div>
</div>
<div class="row">
    <div class="col-md-6">
        <label>Email</label>
    </div>
    <div class="col-md-6">
        <p><?php echo $res['email']?></p>
    </div>
</div>
<div class="row">
    <div class="col-md-6">
        <label>Date of birth</label>
    </div>
    <div class="col-md-6">
        <p><?php echo $res['dob']?></p>
    </div>
</div>
<hr>
<div class="row">
    <div class="col-md-6">
        <label>Permanent Address</label>
    </div>
    <div class="col-md-6">
        <p><?php echo $res['permt_add']?></p>
    </div>
</div>
<div class="row">
```

```
                <label>City</label>
            </div>
            <div class="col-md-6">
                <p><?php echo $res['city']  ?></p>
            </div>
        </div>
        <div class="row">
            <div class="col-md-6">
                <label>Pincode</label>
            </div>
            <div class="col-md-6">
                <p><?php echo $res['pincode'] ?></p>
            </div>
        </div>
        <div class="row">
            <div class="col-md-6">
                <label>State</label>
            </div>
            <div class="col-md-6">
                <p><?php echo $res['state'] ?></p>
            </div>
        </div>
          <hr >
        <div class="row">
            <div class="col-md-6">
                <label>Program</label>
            </div>
            <div class="col-md-6">
                <p><?php echo $res['program'] ?></p>
            </div>
        </div>
        <div class="row">
            <div class="col-md-6">
                <label>Current Semester</label>
            </div>
            <div class="col-md-6">
                <p><?php echo $res['curr_sem'] ?></p>
            </div>
        </div>

        <div class="row">
            <div class="col-md-6">
```

Chapter 4:-System Testing

Software testing forms an activity of software development. Software testing identifies errors at an early stage. A planned testing identifies the difference between the expected results and the actual results. The main objective of software testing is to find errors. A successful testing is one that uncovers, as many as yet undiscovered errors, which helps to make the software more rugged and reliable.Testing is applied at different levels in the software development life cycle, but the testing done is different in nature and has different objective at each level.The focus of all testing is to find errors, but different type of error are looked for each level. Software testing is done at different levels. They are unit testing and system testing which comprises of integration testing and acceptance testing.

7.1 Unit Testing:-

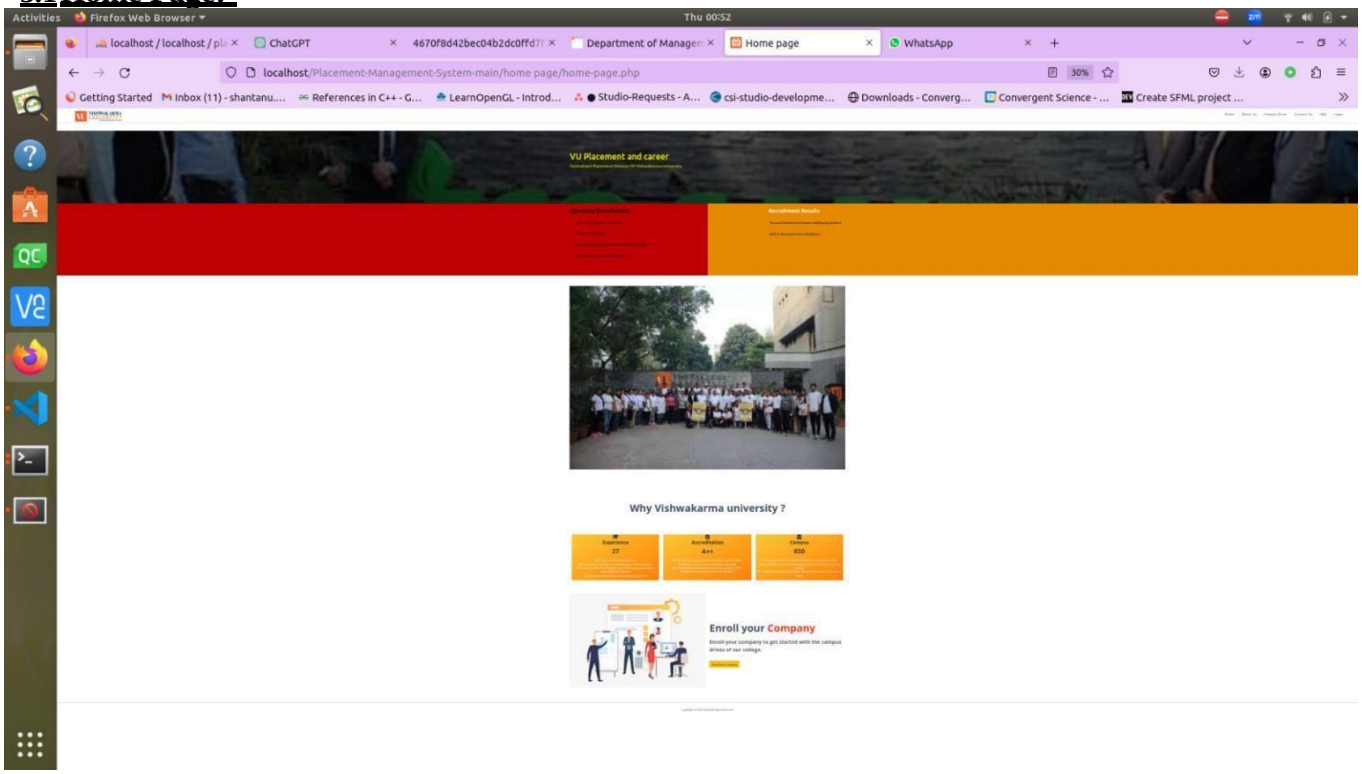
At the level, the function of the basic unit of software is tested in isolation. This is where the most detailed investigation of the internal working of individual units is carried out. The programmer who wrote the code of then performs unit testing. The purpose of unit testing is to find errors in the individual units, which could be logic-related errors. The test case can be derived from their program specification or design document. Units which cannot be tested in isolation may require the creation of small test programs known as harness. Here in our Project we have tested all module individually like login, signup, encryption, storage, decryption.

7.2 Integration Testing:-

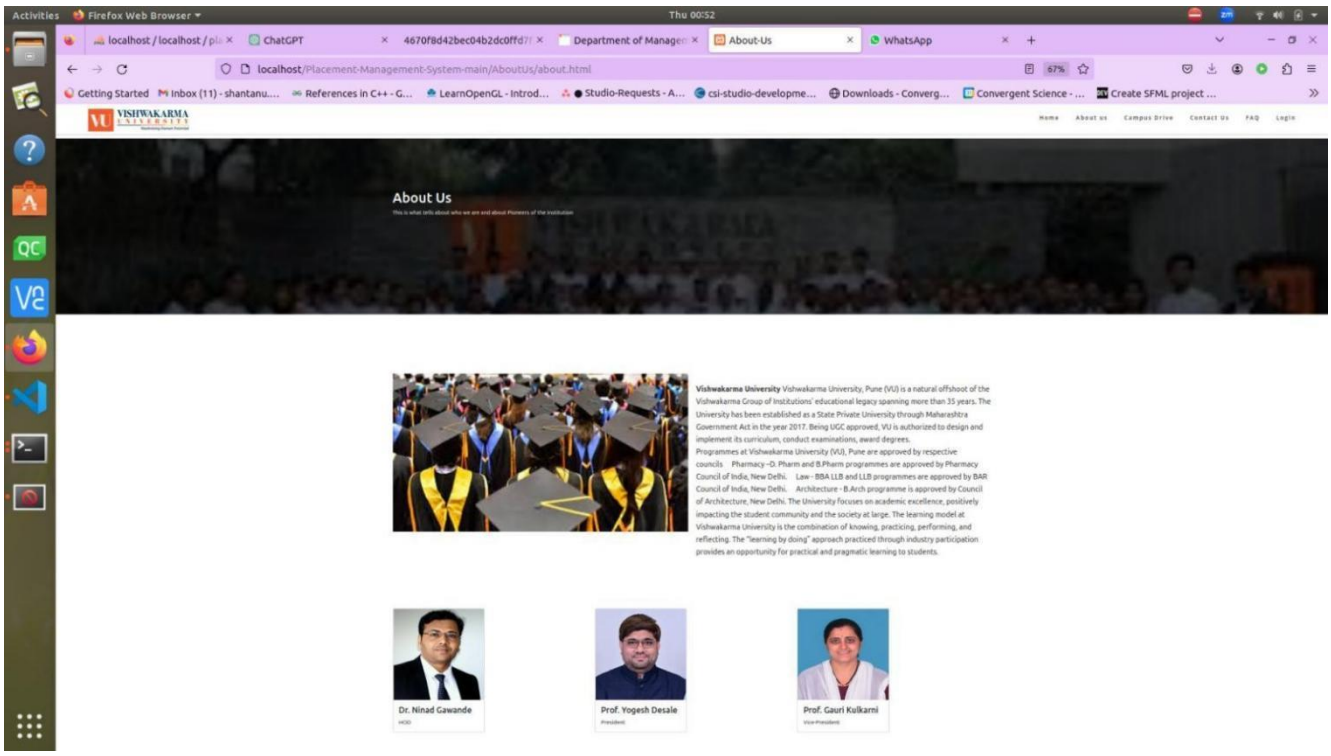
At the level of develop mentation each and every module will be tested individually. When coming to the integration testing, we have to integrate all modules and test the flow and working of modules together. Here in this project also done the same thing and passed the Integration test successfully.

Chapter 5:-Sample Screen

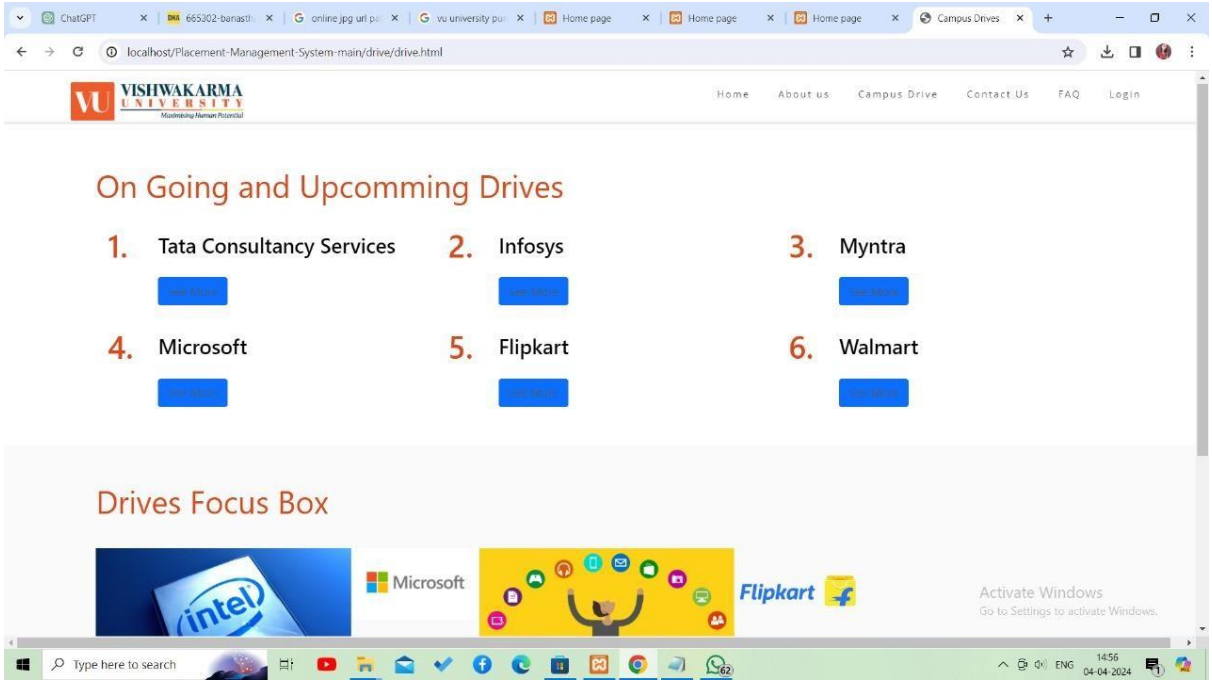
5.1 Home Page:-



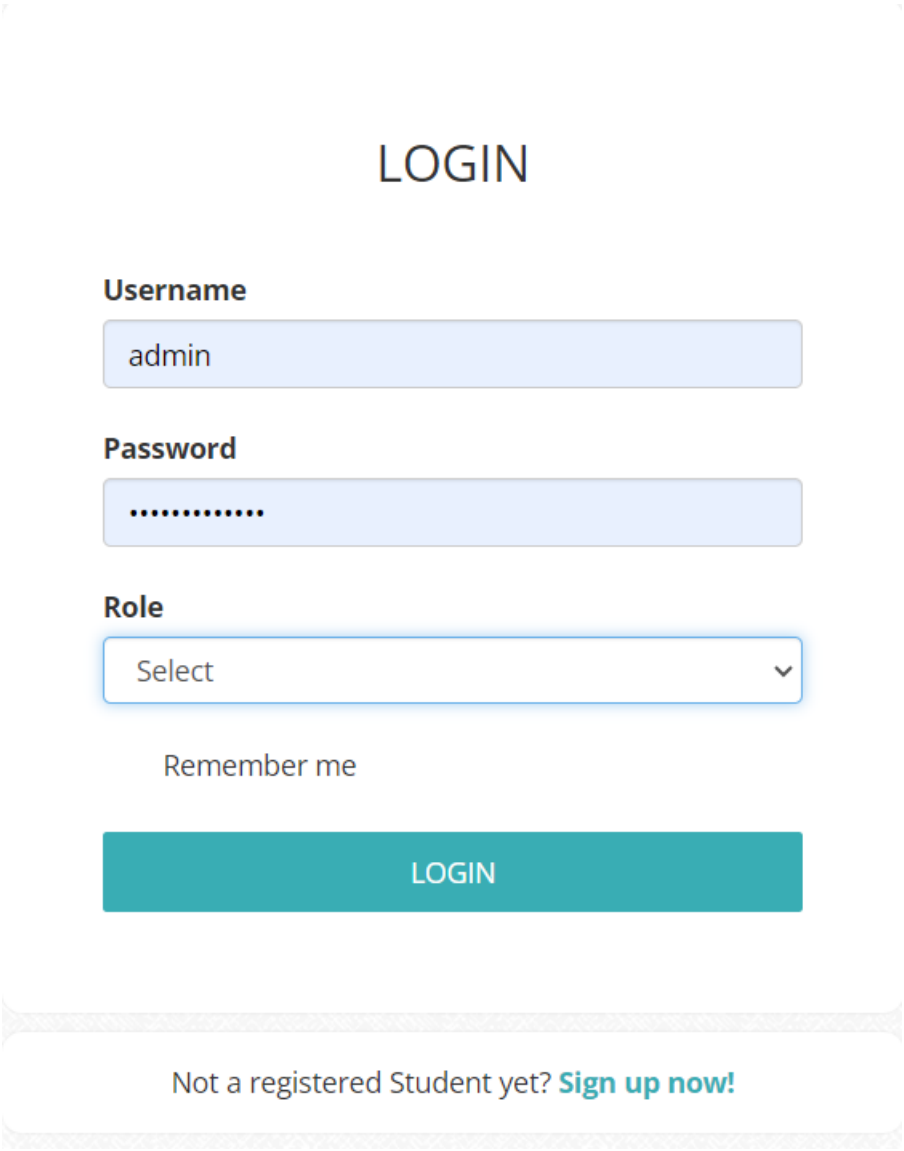
5.2 About us:-


Activities
Firefox Web Browser
Thu 00:52
localhost / localhost / pla
ChatGPT
4670f8d42bec04b2dc0ffd7f
Department of Manage
About-Us
WhatsApp
localhost/Placement-Management-System-main/AboutUs/about.html
67%
Getting Started
Inbox (11) - shantanu....
References in C++ - G...
LearnOpenGL - Introd...
Studio-Requests - A...
csi-studio-developme...
Downloads - Converg...
Convergent Science - ...
Create SFML project ...
VISHWAKARMA UNIVERSITY
Home
About us
Campus Drive
Contact Us
FAQ
Login
About Us
Vishwakarma University Vishwakarma University, Pune (VU) is a natural offshoot of the Vishwakarma Group of Institutions' educational legacy spanning more than 35 years. The University has been established as a State Private University through Maharashtra Government Act in the year 2017. Being UGC approved, VU is authorized to design and implement its curriculum, conduct examinations, award degrees.
Programmes at Vishwakarma University (VU), Pune are approved by respective councils Pharmacy –D. Pharm and B.Pharm programmes are approved by Pharmacy Council of India, New Delhi. Law - BBA LLB and LLB programmes are approved by BAR Council of India, New Delhi. Architecture - B.Arch programme is approved by Council of Architecture, New Delhi. The University focuses on academic excellence, positively impacting the student community and the society at large. The learning model at Vishwakarma University is the combination of knowing, practicing, performing, and reflecting. The "learning by doing" approach practiced through industry participation provides an opportunity for practical and pragmatic learning to students.
Dr. Ninad Gawande
Prof. Yogesh Desale
Prof. Gauri Kulkarni

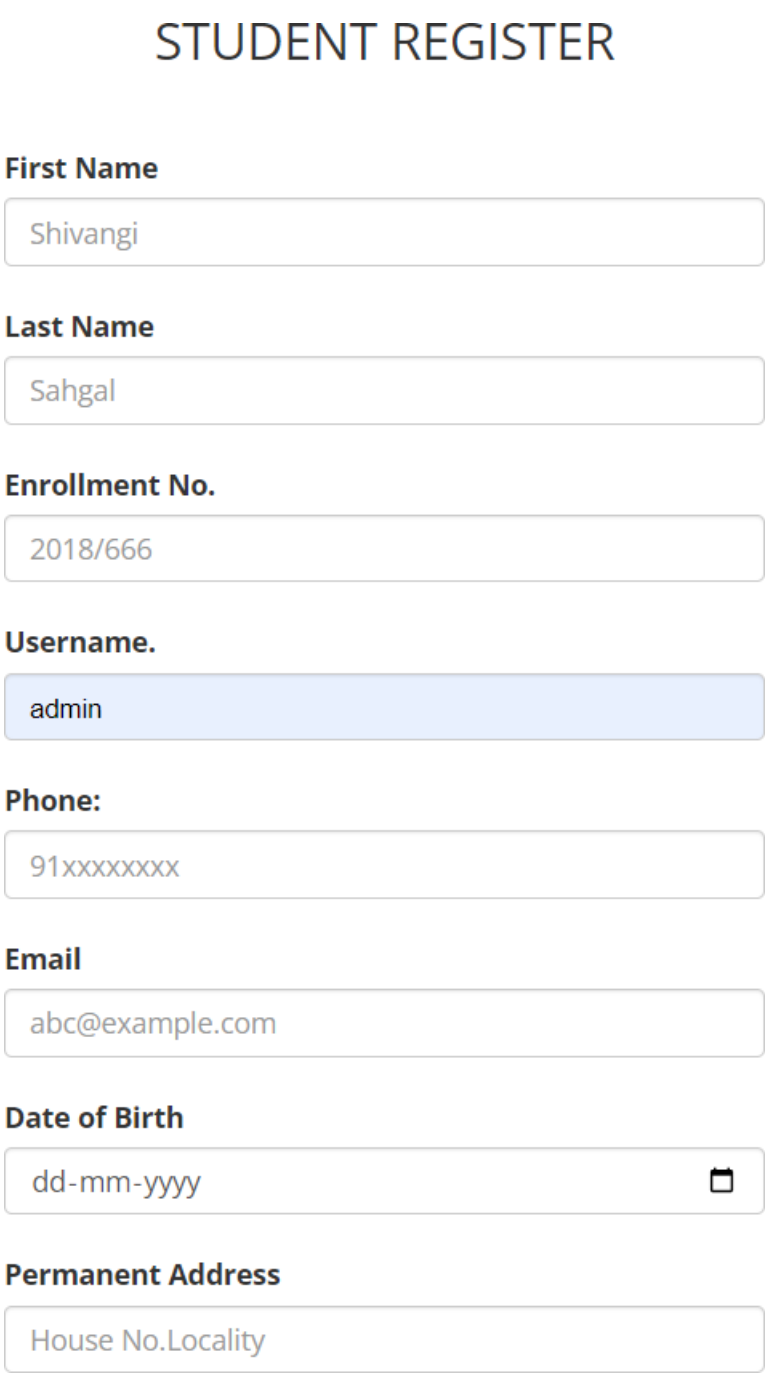
5.3 Campus Drive:-



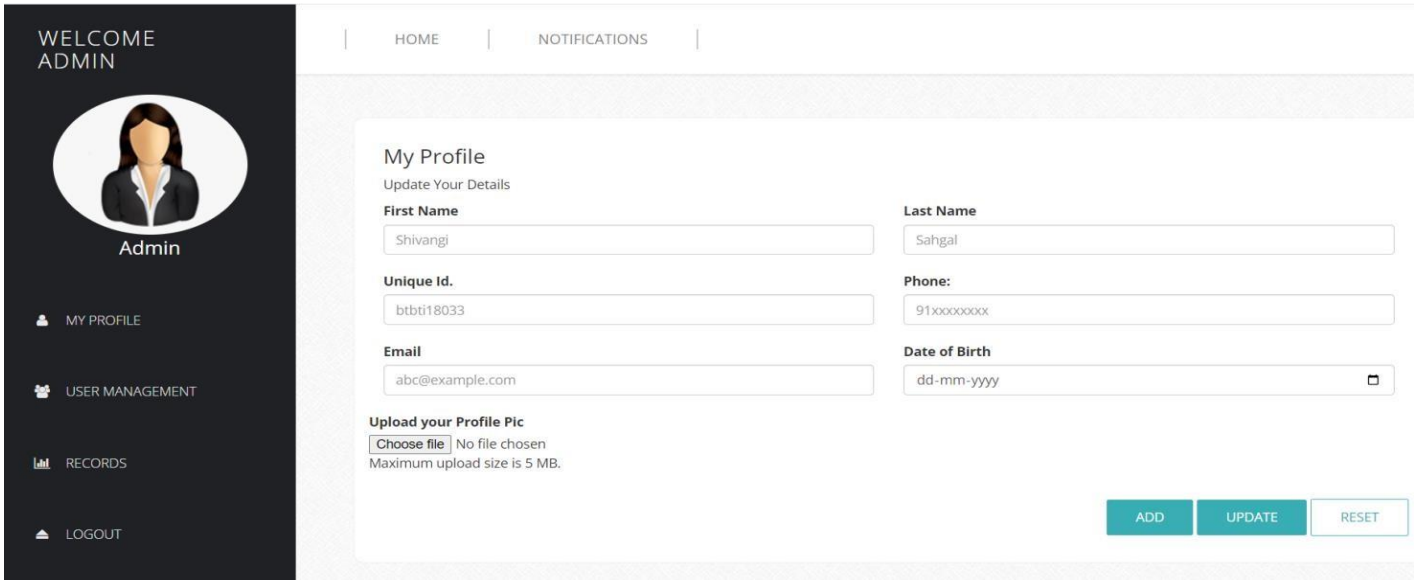
5.4 Login Page:-


LOGIN
Username
admin
Password
Role
Select
Remember me
LOGIN
Not a registered Student yet? Sign up now!

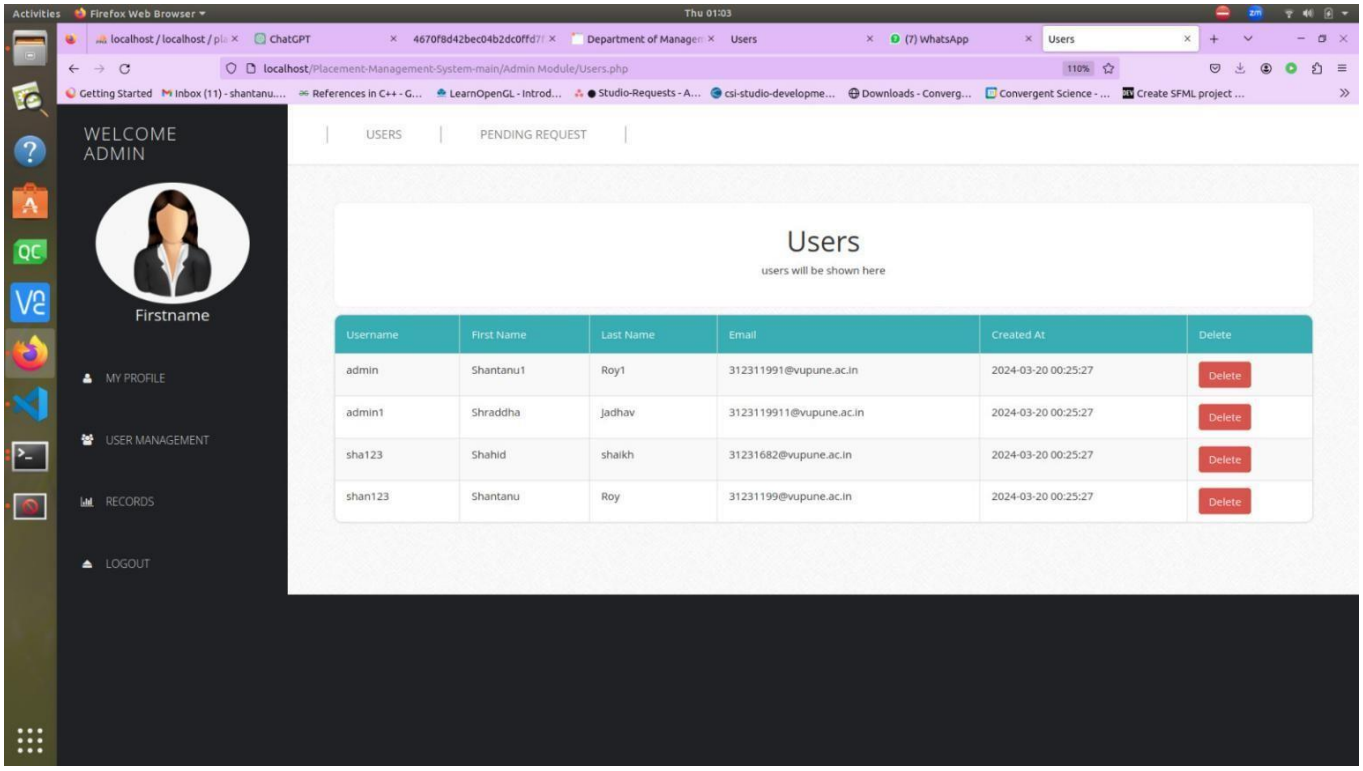
Registration Page:-



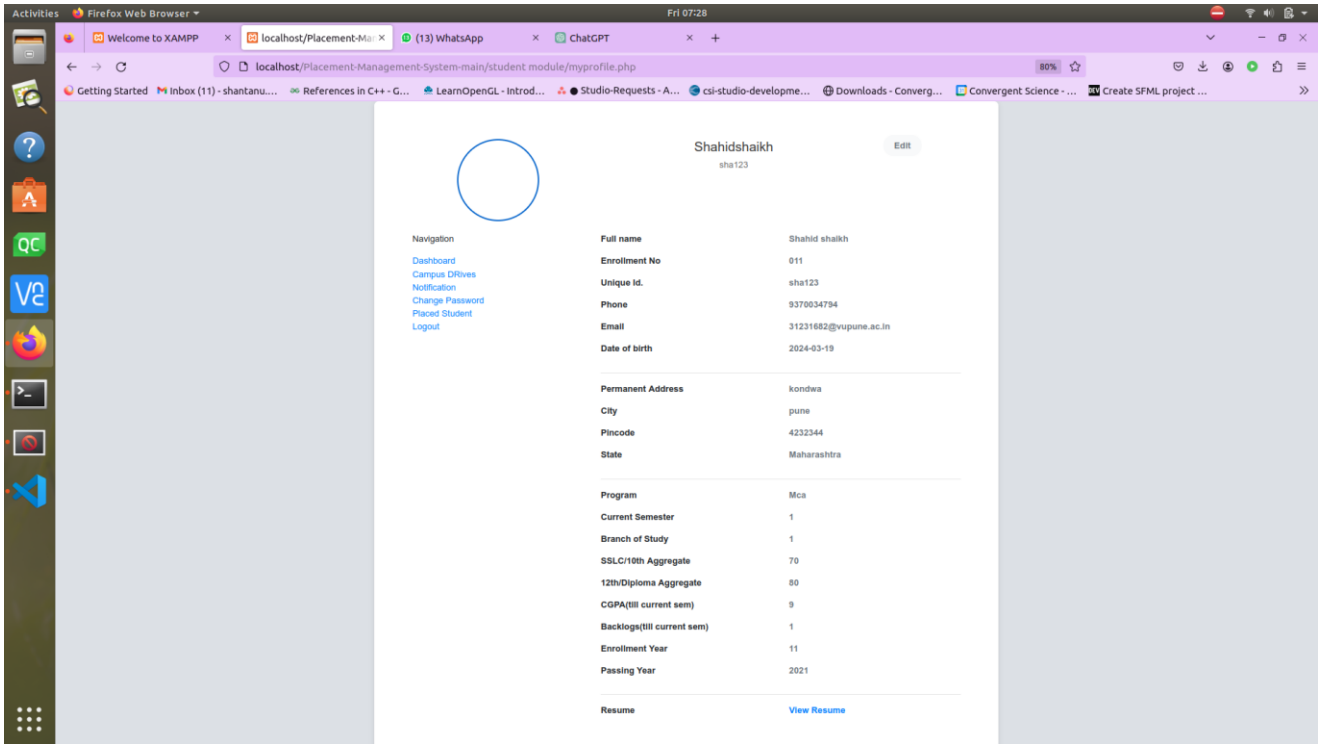
5.6 Admin Page:-



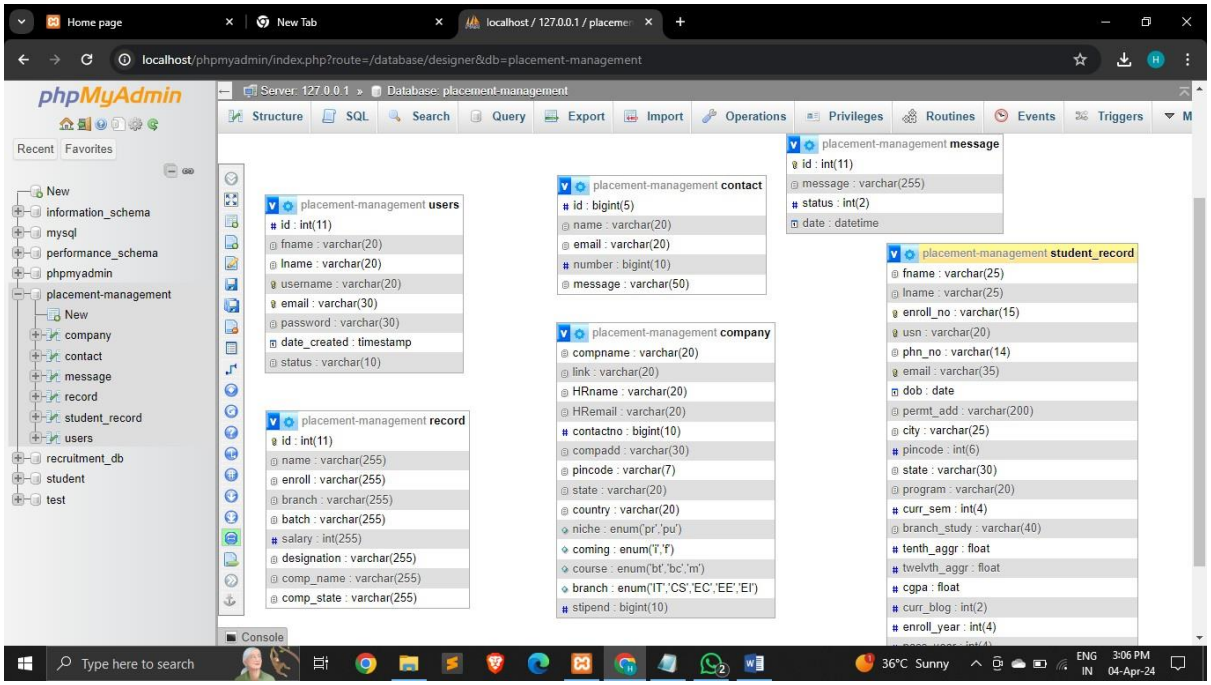
USER PAGE:



USER PAGE:



Database Design:


Home page
New Tab
localhost / 127.0.0.1 / placemer
localhost/phpmyadmin/index.php?route=/database/designer&db=placement-management
phpMyAdmin
Recent
Favorites
New
information_schema
mysql
performance_schema
phpmyadmin
placement-management
New
company
contact
message
record
student_record
users
recruitment_db
student
test
Server: 127.0.0.1 » Database: placement-management
Structure
SQL
Search
Query
Export
Import
Operations
Privileges
Routines
Events
Triggers
placement-management users
id : int(11)
fname : varchar(20)
lname : varchar(20)
username : varchar(20)
email : varchar(30)
password : varchar(30)
date_created : timestamp
status : varchar(10)
placement-management record
id : int(11)
name : varchar(255)
enroll : varchar(255)
branch : varchar(255)
batch : varchar(255)
salary : int(255)
designation : varchar(255)
comp_name : varchar(255)
comp_state : varchar(255)
placement-management contact
id : bigint(5)
name : varchar(20)
email : varchar(20)
number : bigint(10)
message : varchar(50)
placement-management company
compname : varchar(20)
link : varchar(20)
HRname : varchar(20)
HRemail : varchar(20)
contactno : bigint(10)
compadd : varchar(30)
pincode : varchar(7)
state : varchar(20)
country : varchar(20)
niche : enum('pr','pu')
coming : enum('i','f')
course : enum('bt','bc','m')
branch : enum('IT','CS','EC','EE','EI')
stipend : bigint(10)
placement-management message
id : int(11)
message : varchar(255)
status : int(2)
date : datetime
placement-management student_record
fname : varchar(25)
lname : varchar(25)
enroll_no : varchar(15)
usn : varchar(20)
phn_no : varchar(14)
email : varchar(35)
dob : date
permt_add : varchar(200)
city : varchar(25)
pincode : int(6)
state : varchar(30)
program : varchar(20)
curr_sem : int(4)
branch_study : varchar(40)
tenth_aggr : float
twelvth_aggr : float
cgpa : float
curr_blog : int(2)
enroll_year : int(4)
Console
Type here to search
36°C Sunny
ENG
IN
3:06 PM
04-Apr-24

Chapter 6:-conclusion and Future Enhancement

6.1 Conclusion:-

1. In conclusion, the "Campus Placement Recruitment System" revolutionizes the way campus placements are conducted and navigated. By providing a user-centric platform, it offers a range of benefits for students, educational institutions, and potential employers.

2. With features such as easy search and application for placement opportunities, convenient access to placement details, and ongoing career support this system ensures that students are empowered in their pursuit of opportunities.

3. The "Campus Placement Recruitment System" is a game-changer in the realm of campus placements, creating a positive impact on the future careers of students and the success of educational institutions.

6.2 Future enhancements:-

The future enhancements of a Campus Placement and Recruitment System could involve integrating advanced technologies and addressing emerging trends to improve efficiency, effectiveness, and user experience. Here are some potential areas for enhancement:

AI-Powered Matching: Implement artificial intelligence (AI) algorithms to enhance candidate-employer matching. AI can analyze candidate profiles, job requirements, and historical data to suggest the best-fit candidates for job openings, improving the quality of matches and reducing recruitment time.

Predictive Analytics: Utilize predictive analytics to forecast hiring trends, identify high-demand skills, and anticipate future job market needs. By analyzing historical placement data and industry trends, the system can provide insights to students and institutions to better prepare for upcoming opportunities.

Virtual Recruitment Events: Enhance the system with virtual recruitment event capabilities, allowing employers to conduct online job fairs, interviews, and networking sessions. Virtual platforms can facilitate global participation, increase accessibility, and reduce logistical constraints associated with physical events specific intervals or events.

- **Predictive Analytics:-**

  Incorporate predictive analytics to forecast placement trends, identify potential job opportunities, or recommend training programs based on historical data.

User Experience and Interface:-

- **Responsive Web Interface:-**

  Develop a responsive web-based interface using modern frontend frameworks like React, Angular, or Vue.js for better accessibility across devices.

- **Personalized Dashboards:-**

  Implement customizable dashboards allowing users to arrange and display relevant reports and data widgets based on their preferences.

- **Enhanced Search and Filter Options:-**

  Improve search functionalities with advanced filtering, sorting, and data grouping options to quickly find and analyze specific information.

- **Intuitive Visualization of Data:-**

  Provide intuitive visual representations of complex data relationships through interactive UI elements and data hierarchies.

Functional Enhancements:-

- **Integration with External APIs:-**

  Integrate with job boards, career websites, or industry-specific APIs to fetch real-time job listings, industry trends, or company information for comprehensive reports.

- **Machine Learning for Recommendation Systems:-**

  Use machine learning algorithms to suggest suitable job opportunities or training programs based on individual student profiles and industry requirements.

- **Automated Email Alerts:-**

  Implement automated email notifications/alerts for upcoming interviews, application deadlines, or important placement-related events.

- **Feedback and Survey Modules:-**

  Develop modules to gather feedback from students, recruiters, and trainers to continually improve the training and placement processes.

Security and Performance:-

- **Data Encryption and Security Measures:-**

  Strengthen security by implementing data encryption, role-based access control, and secure authentication mechanisms to protect sensitive information.

- **Performance Optimization:-**

  Continuously optimize database queries, indexing strategies, and server configurations for better system performance and responsiveness.

By focusing on these areas for enhancement, the Training and Placement Management System can evolve to meet the changing needs of educational institutions, students, and recruiters, providing more comprehensive and efficient services for training and placement activities.

Bibliography

Smith, John. "PHP for Beginners."

This book provided a comprehensive introduction to PHP programming, helping in understanding the fundamentals of PHP language.

Welling, Luke, and Thomson, Laura. "PHP and MySQL Web Development." Addison-Wesley, Year.

This book offered insights into building dynamic web applications using PHP and MySQL, which was crucial for developing the backend of the campus placement system.

Freeman, Elisabeth. "Head First PHP & MySQL." O'Reilly Media, Year.

This resource provided a beginner-friendly approach to learning PHP and MySQL, offering practical examples and exercises that were helpful in implementing various features of the project.

Online PHP Documentation: php.net

The official PHP documentation was consulted regularly for reference on PHP functions, syntax, and best practices.

Online MySQL Documentation: dev.mysql.com

The official MySQL documentation served as a guide for database design, SQL queries, and optimization techniques.

Stack Overflow: stackoverflow.com

Various questions and answers on Stack Overflow were referred to for troubleshooting specific issues and gaining insights into best practices in PHP development.

W3Schools PHP Tutorial: w3schools.com

This online tutorial provided additional explanations and examples for PHP programming concepts, helping in reinforcing understanding and solving coding challenges.

Tutorialspoint PHP Tutorial: tutorialspoint.com

Another online tutorial that offered in-depth coverage of PHP programming concepts, aiding in the development of complex functionalities for the campus placement system.

GitHub: github.com