SYNOPSIS

1.Synopsis

1.1 Title of the Project:

Career Hunt

1.2. Objective of the Project:

- To develop a user-friendly job portal that bridges the gap between applicants and recruiters.
- To enable applicants to create profiles, upload resumes, and apply for relevant job openings.
- To allow recruiters to post job listings, search for candidates, and manage applications.
- To provide a secure and interactive platform for job hunting and hiring

1.3. Project Category:

Web Application

1.4. Language(s) to be used:

Front End: HTML, CSS, JQuery
Back End: PHP
Database: MySQL

1.5. Structure of the program:

1.5.1Analysis:

The project aims to create an online job portal that connects the job seekers with recruiters. It will provide an efficient platform where applicant can apply for jobs, upload resume, and track applications, while recruiters can post job vacancies, review applications and communicate with potential candidates.

1.5.2. <u>Module Description:</u>

1.5.2.1. Applicant Modules:

1. Student Registration: Allow students to create profiles, upload resumes, and provide relevant information.
2. Job Search: Enable students to search and filter job openings based on criteria like location, industry, and job type.
3. Job Application: Allow students to apply for jobs directly through the platform.
5. Profile Management: Allow students to manage their profiles, update resumes, and track application status.

1.5.2.2. Recruiter Modules:

1. Recruiter Registration: Allow recruiters to create profiles, upload company information, and post job openings.

2. Job Posting: Enable recruiters to post job openings, including job descriptions, requirements and application instructions.
3. Candidate Management: Allow recruiters to view, filter, and manage candidate applications.
5. Reporting and Analytics: Provide recruiters with insights into candidate engagement, application metrics, and other relevant data.

1.5.2.3.Admin Modules:
1. User Management: Allow admins to manage student and recruiter accounts, including registration, profile updates, and account deletion.
2. Job Approval: Enable admins to review, approve, or reject job postings.
3. Reporting and Analytics: Provide admins with insights into platform usage, candidate engagement, and recruiter activity.
4. Settings and Configuration: Allow admins to configure platform settings, such as email notifications, job posting templates, and user permissions.
5. Support and Feedback: Enable admins to provide support to users, collect feedback, and track issues.

<u>1.6. Data structure:</u>

- 1.6.1. Admin:
  >Login
    >Username
    >Password
- 1.6.2. Recruiter:
  >Sign Up:
    >Company name
    >Contact
    >Gmail
    >Password
    >Repeat Password
    >Phone No
    >Company URL
    >Address
    >Logo
  >Login:
    >Gmail
    >Password

- 1.6.3. Applicant:
  >Sign Up :
    >Name
    >Gmail
    >Password
    >Repeat Password
    >Student ID
    >Gender

>Log In:
>Gmail
>Password

1.7. Any other information:

The definition and maintenance of the database is done through the ACCESS Admin interface.

1.8. Future scope of the Project:

- AI-Powered Job Recommendations
- Video Interview Integration
- Automated Candidate Screening
- Resume Parsing
- Integration With LinkedIn and Other Platform.

Software Requirements Specification

2.Software Requirements Specification (SRS)

2.1. Introduction:

The Career Hunt system is a web-based job portal designed to streamline the campus recruitment process by bridging the gap between students and recruiters. This system provides a centralized platform where students can register, upload resumes, search for jobs, and track their applications, while recruiters can post job vacancies, filter candidates, and manage the hiring process. The system also includes an admin panel for managing user accounts, approving job postings, and monitoring platform usage.

With an intuitive interface and secure data handling, Career Hunt aims to enhance job accessibility for students and simplify recruitment for companies. The system is designed to be scalable, secure, and efficient, ensuring a seamless experience for all users. Future enhancements may include AI-powered job recommendations, video interview capabilities, and integration with professional networks like LinkedIn.

2.1.1 Purpose:

The purpose of the Career Hunt platform is to create a user-friendly, secure, and efficient job portal that connects applicants (students) with recruiters (companies) for campus placements. The system allows students to apply for jobs, upload resumes, and track applications, while recruiters can post job vacancies, manage candidates, and conduct hiring processes.

2.1.2 Scope:

The Career Hunt system will provide:

- Students: Registration, profile management, job search & application tracking.
- Recruiters: Job posting, candidate management, and analytics.
- Admin: Platform management, job approval, reporting, and configuration.
- Secure authentication for all users.
- Data analytics for recruiters and administrators.

2.1.3 Definitions, Acronyms, and Abbreviations:

-Not applicable

2.1.4 References:

- IEEE Standard 830-1998 for Software Requirements Specification

2.1.5 Overview:

This document outlines the functional and non-functional requirements of Career Hunt, detailing the system’s features, performance expectations, and constraints.

2.2. Overall Description:

2.2.1 Product Perspective:

Career Hunt is a web-based application that connects students and recruiters, streamlining the recruitment process. It provides a centralized campus hiring platform with a secure database and real-time updates.

2.2.2 Product Functions:

- Student Portal: Register, upload resumes, search & apply for jobs, and track applications.
- Recruiter Portal: Post job vacancies, filter candidates.
- Admin Panel: Manage users, approve job postings, configure settings, and generate reports.

2.2.3 User Characteristics:

- Students: Tech-savvy users familiar with job applications.
- Recruiters: HR professionals with hiring experience.
- Admin: Placement officers with full system control.

2.2.4 General Constraints:

- Secure multi-level authentication for data privacy.
- Compatible with any system that supports modern web browsers.
- 99.5% uptime for uninterrupted access.

2.2.5 Assumptions and Dependencies:

- Users must have valid credentials for access.
- Internet connectivity is required for real-time updates.
- Future integrations (LinkedIn, AI screening, video interviews) may be added.

2.3. Specific Requirements:

The specific requirement section describes all the details that the software developers need to know for designing and developing the system.

2.3.1 External Interface Requirements:

The interface should be simple and easy to understand and use. It should be interactive interface. It should prompt the user and administrator for proper input criteria.

2.3.1.1 User Interfaces:

- Login/Sign-up pages for students, recruiters, and admin.
- Dashboard for job applications and analytics and others.
- Profile Management for Students, Recruiters and Admin.

2.3.1.2 Hardware Interfaces:

- Compatible with all standard browsers.

2.3.1.3 Software Interfaces:

- MySQL database for storing user and job data.
-XAAMP for local Hosting.
- Integration with email services for notifications (future enhancement).

2.3.1.4 Communication Interfaces:

- Email & SMS notifications for job updates, interview invites, and selection results (future enhancement).
- Chat module (future enhancement) for direct recruiter-student communication.

2.3.2 Functional Requirements:

2.3.2.1 Student Module:

- Student Registration: Sign-up with name, email, password, and student ID.
- Job Search: Filter jobs based on industry, location, and eligibility.
- Apply for Jobs: Upload resume, track application status.
- Profile Management: Update resume and view recruiter interactions.

2.3.2.2 Recruiter Module:

- Recruiter Registration: Sign-up with company details and contact information.
- Job Posting: Add job descriptions, salary range, and requirements.
- Candidate Management: Filter applicants, shortlist candidates.
- Reporting & Analytics: View applicant engagement and hiring statistics.

2.3.2.3 Admin Module:

- User Management: Approve or remove student and recruiter accounts.
- Job Approval: Review and validate job postings.
- Platform Settings: Configure notification settings, permissions.
- Reporting & Analytics: View hiring statistics.

2.3.3 Performance Requirements:

- System should handle 100+ concurrent users without performance lag.
- Job search and application processes should complete in <10 seconds.

2.3.4 Design Constraints:

- Security: Data encryption for student, recruiter and admin profiles passwords (Hashing password).
- Scalability: AI-based job recommendations (future versions).

2.3.5 System Attributes:

- Reliability: 99.5% uptime with auto-backup.
- Maintainability: Modular design for future updates.

2.3.6 Other Requirements:

- Data Privacy Compliance: Secure student and recruiter data storage.

System design

3.System Design

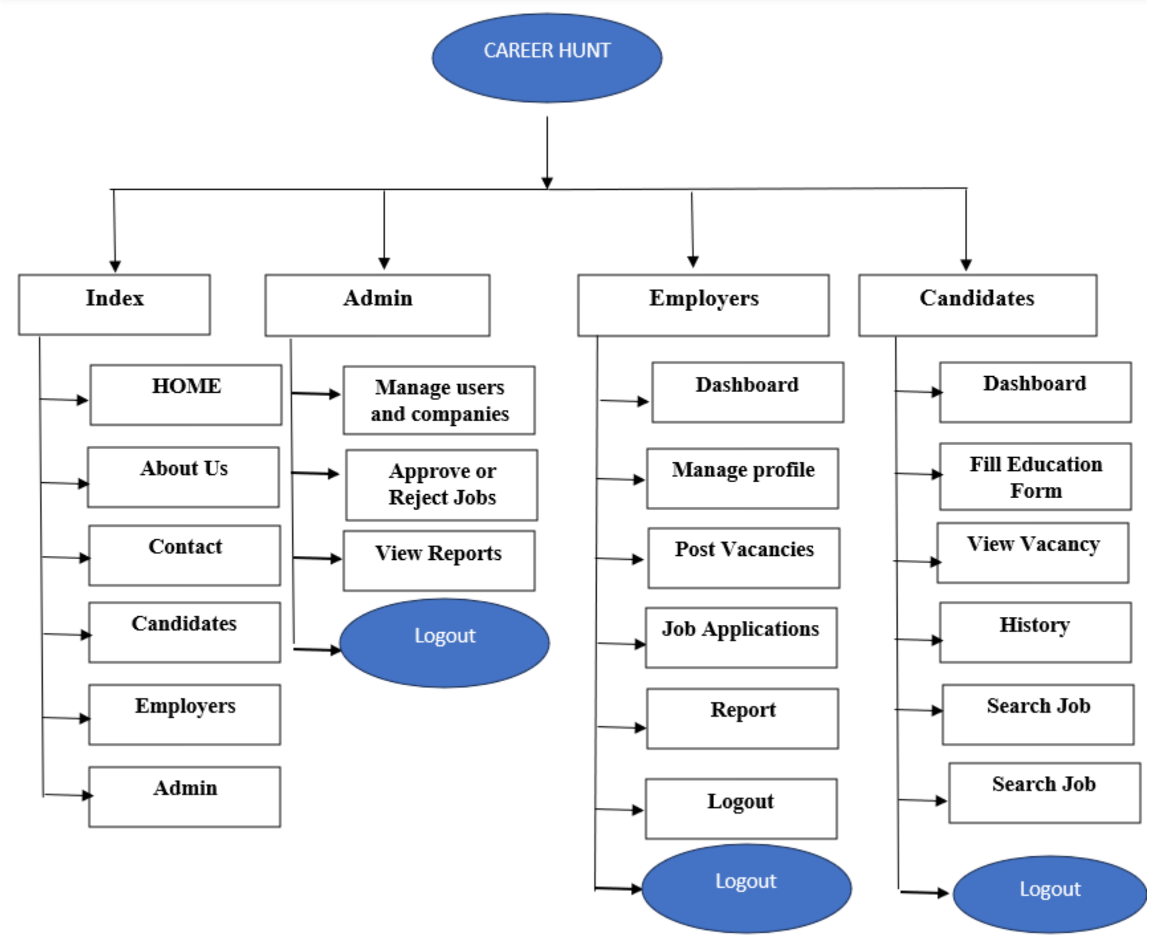
3.1. Introduction:

The System Design phase for the Campus Recruitment System transforms the SRS into a detailed blueprint for development, focusing on the Admin, Company, and Applicant user models. It outlines the system's architecture, including the presentation, business logic, and data access layers, ensuring scalability and maintainability. A relational database will manage data, with robust security measures like RBAC. UML diagrams will illustrate the system's structure and behaviour, facilitating clear communication among stakeholders. This design prioritizes usability, performance, and reliability, aiming for a system that meets all user needs and is delivered efficiently.

3.2. Functional decomposition:

Functional decomposition in the Campus Recruitment System breaks down complex tasks into simpler modules for Admin, Company, and Applicant. For Admin, it includes user management and reporting; for Company, Post Vacancies and Receives Applications; for Applicant, apply for jobs and view the Status. This modular approach simplifies each user's interaction, making the system more manageable and understandable.

3.3. System Software Architecture:



3.4 Functional Components:

3.4.1 Sign In:

This module allows the user (Admin, Company, Applicant) to sign up to the portal using already registered username and password.

3.4.2 Registration:

This module is used to enter the new user (Company, Applicant) to application.

3.4.3 Admin:

This module allows the administrator to manage the Website

- **View Company:**
  This allows admin user to view the registered company and posted vacancies and also can edit them.
- **View Registered Users:**
  This allows to viewing all registered users and the vacancies they have applied.
- **Posted Vacancies/jobs:**
  This is used to Edit and Delete posted vacancies.
- **Total job Applications:**
  This allows Admin to view the Applications that applicant applied.
- **New job Applications:**
  This allows to view list of newly applied job applications.
- **Sort Listed job Applications:**
  Allows to view the sort listed application/applicant.
- **Selected job Applications:**
  Allows to view the selected application/applicant.
- **Rejected job Applications:**
  Allows to view the rejected application/applicant

3.4.4 Company:

This module allows the Company to manage the Application and Vacancies of the college.

- **Posted Vacancies/jobs:**
  This is used to Edit and Delete posted vacancies.
- **Total job Applications:**
  This allows Company to view total job applications received.
- **New job Applications:**
  This allows to view list of newly applied job applications.
- **Sort Listed job Applications:**
- Allows to view the sort listed application/applicant and also take actions.
- **Selected job Applications:**
  Allows to view the selected application/applicant.
- **Rejected job Applications:**
  Allows to view the rejected application/applicant.

3.4.5 Applicant: This module allows the user(applicant) to view the newly posted vacancies and applied for that Vacancies.

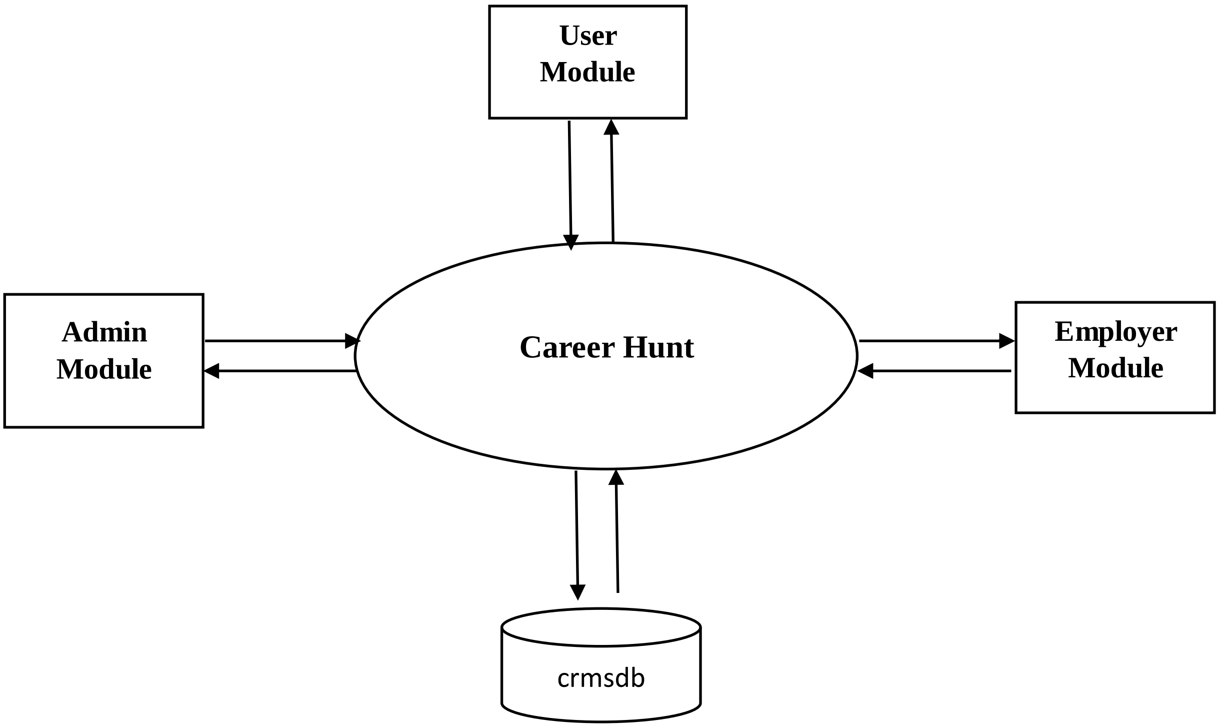
- **Listed Vacancies/jobs:**
  This is used to view the listed vacancies and applied foe that vacancies.
- **Total Applied Jobs:**
  This allows to view all the applied jobs/vacancies.
- **Todays Applied Jobs:**
  This allows to view all the applied jobs/vacancies today.

- **Yesterdays Applied Jobs:**
  This allows to view all the applied jobs/vacancies yesterdays.
- **Applied Jobs in Last 7 days:**
  This allows to view all the applied jobs/vacancies.

3.5. Description of program:

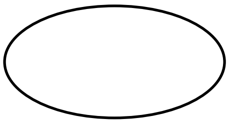
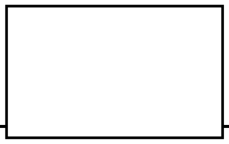
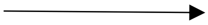
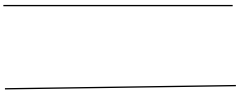
3.5.1. Context Flow Diagram (CFD)

The Campus Recruitment System's CFD visualizes data flow between the system and external entities: Admin, Company, and Applicant.

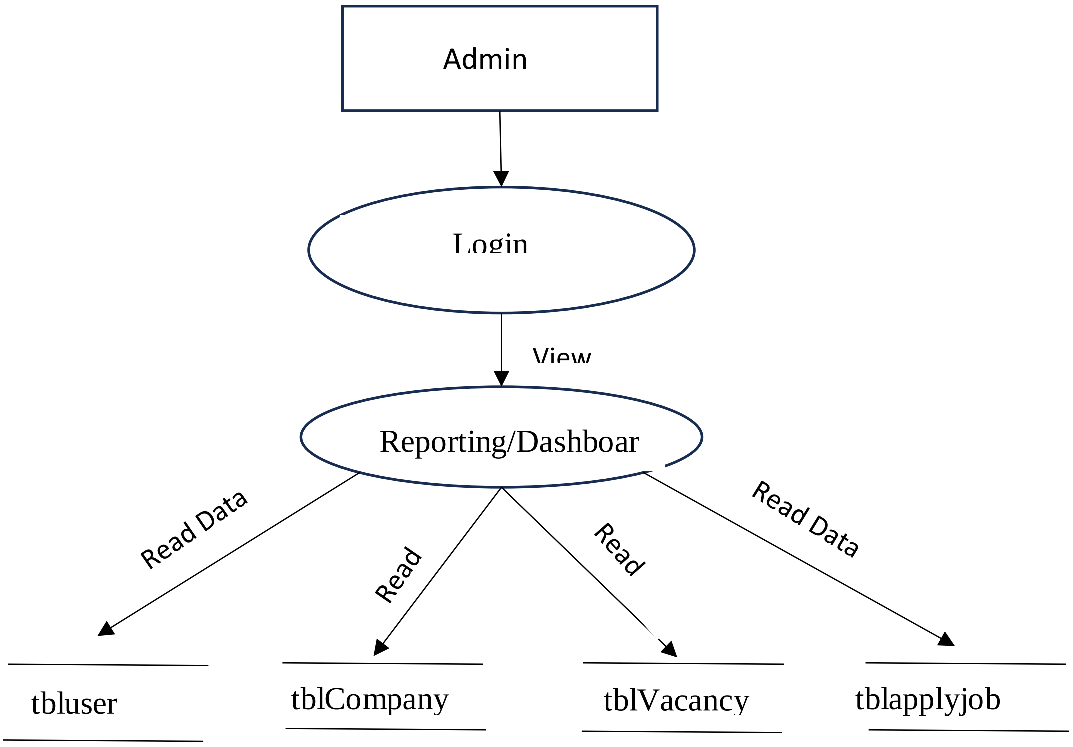


3.5.2 Data Flow Diagram : Data Flow Diagram is a graphical representation of a system or a portion of the system. It contains the data flows, process, sources and sinks and it store all the Description through the use of easily understandable symbols.

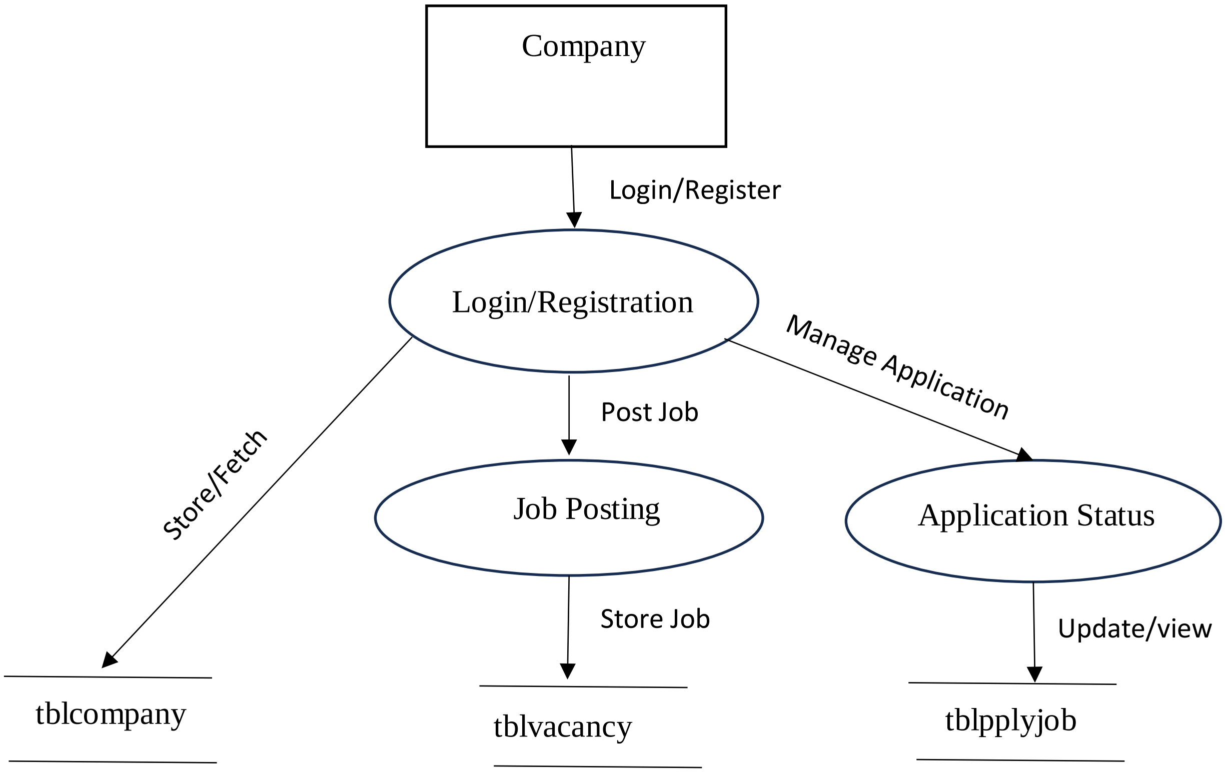
- DFD is one of the most important model tools, it is used to model the system, components that interact with the system, uses the data and DFD.
- DFD shows how the information moves through and hope it is modified by a series of transformation. It is one of the graphical techniques.
- DFD is also known as bubble chart or data flow graph. DFD is used to represent the system at any level abstraction.

Symbols	Name	Description
	**Process**	**It performs transformation of data from one state to another.**
	**Source /Sink**	**It represents the external entity that may be eithersource or Sink.**
	**Flow of data**	**It represents the flow ofdata from source to destination**
	**Data Source/Data storage**	**It is the place where data is stored.**

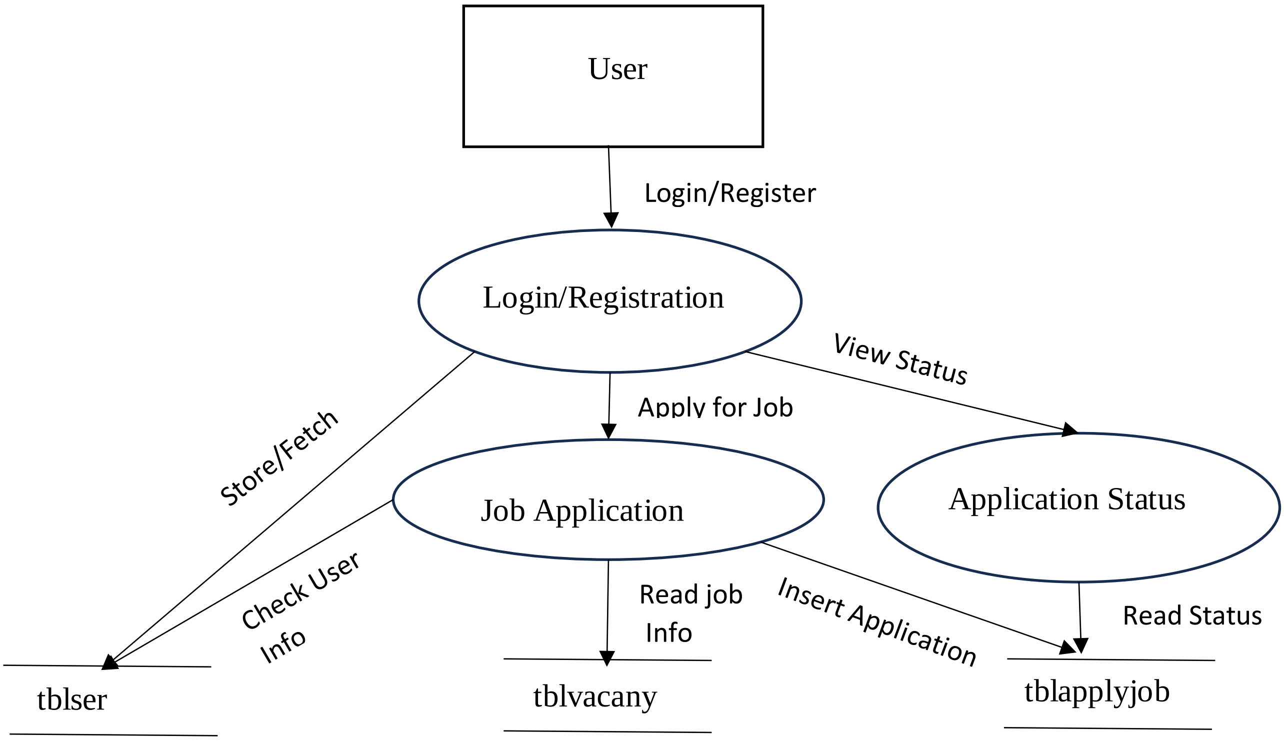
3.5.2.1 DFD for Admin Management



3.5.2.2 DFD for Company Management



3.5.2.3 DFD for User Management



Database Design

4. DATABASE DESIGN

4.1. Introduction

Database is a collection of related data. Relational database stores data in a table or relations. The data stored in a relation are arranged in records. Each record consists of set of attributes. Fields can be referred to characteristics of records. This document describes the table that is used to design software, its attributes, data types, constraints and relationship among those tables.

4.2. Table Definition

4.2.1. Admin(tbladmin)

Field	Type	Constraints	Description
ID	Int	Primary key, auto increment	Unique ID
AdminName	Varchar(120)	Max length :120	Admin Name
UserName	Varchar(120)	Max length :120	User Name
MobileNumber	bigint	unique	Phone Number
Email	Varchar(120)	unique	Email Address
Password	Varchar(120)	Max length :120	Encrypted password
AdminRegdate	timestamp	Default Generated	Registration time

4.2.2. User(tbluser)

Field	Type	Constraints	Description
ID	Int	Not Null, Primary KEY, auto Increment	Unique ID
FullName	Varchar(120)	Max length :120	User Name
Email	Varchar(120)	Max length :120	User Email
MobileNumber	bigint	unique	Mobile Number
StudentId	Varchar(120)	Max length :120	Student ID
Gender	Varchar(120)	Max length :120	Gender Of User
Address	midiumtext	-	Address Of user
Age	int	-	Age Of User
DOB	Varchar(120)	Max length :120	Date Of Birth

Image	Varchar(120)	Max length :120	Profile Photo
Password	Varchar(120)	Max length :120	Password
UserRegdate	timestamp	Default Generated	Registration time

4.2.3 .Company(tblcompany)

Field	Type	Constraints	Description
ID	int	Not Null, Primary KEY, auto Increment	Unique ID
CompanyName	Varchar(120)	Max length :120	Company Name
ContactPerson	Varchar(120)	Max length :120	Company person Contact
CompanyUrl	Varchar(120)	Max length :120	Company URL
CompanyAddress	Midiumtext	unique	Company Address
MobileNumber	Bigint	Unique	Phone Number
CompanyEmail	Varchar(120)	Max length :120	Company Email
ComapnyLogo	Varchar(120)	Max length :120	Company LOGO
Password	Varchar(120)	Max length :120	Password
CompanyRegdate	timestamp	Default Generated	Registration time

4.2.4. Apply job(tblapplyjob)

Field	Type	Constraints	Description
ID	int	Not Null, Primary KEY, auto Increment	Unique ID
UserID	Int	Unique	User ID
JobId	Int	unique	Job ID
Resume	Varchar(120)	Max length :120	Resume
ApplyDate	Timestamp	DEFAULT_GENERATED	Applied Date
Message	Mediumtext	Not null	Message
Remark	Varchar(200)	Not null	Remarks
Status	Varchar(50)	Max length :50	Status
ResponseDate	timestamp	Not null	on update CURRENT_TIMESTAMP

4.2.5. message (tblmessage)

Field	Type	Constraints	Description
ID	int	Not Null, Primary KEY, auto Increment	Unique ID

AppID	int	Unique	APP ID
Message	Midiumtext	Not null	Messages
Status	Varchar(120)	-	Status
ResponseDate	Timestamp	DEFAULT_GENERATED	CURRENT_TIMESTAMP
IsRead	Char(1)	-	-

4.2.6. Education(tbleducation)

Field	Type	Constraints	Description
ID	Int	Not Null, Primary KEY, auto Increment	Unique ID
UserID	Int	Unique	User ID
SecondaryBoard	varchar(100)	Max length :100	X -Board Name
SecondaryBoardyop	varchar(120)	Max length :120	Year
SecondaryBoardper	varchar(120)	Max length :120	Percentage
SecondaryBoardcgpa	varchar(120)	Max length :120	CGPA
SSecondaryBoard	varchar(120)	Max length :120	12th -Board Name
SSecondaryBoardyop	varchar(120)	Max length :120	Year
SSecondaryBoardper	varchar(120)	Max length :120	Percentage
SSecondaryBoardcgpa	varchar(120)	Max length :120	CGPA
GraUni	varchar(120)	Max length :120	Graduation
GraUniyop	varchar(120)	Max length :120	Year

GraUnidper	varchar(120)	Max length :120	percentage
GraUnicgpa	varchar(120)	Max length :120	CGPA
PGUni	varchar(120)	Max length :120	Post Graduation
PGUniyop	varchar(120)	Max length :120	Year
PGUniper	varchar(120)	Max length :120	Percentage
PGUnicgpa	varchar(120)	Max length :120	CGPA
ExtraCurriculars	varchar(120)	Max length :120	Extra Activities
OtherAchivement	varchar(120)	Max length :120	Achivements

4.2.7. Vacancy(tblvacancy)

Field	Type	Constraints	Description
ID	int	Not Null, Primary KEY, auto Increment	Unique ID
CompanyID	Int	Unique	Company ID
JobTitle	varchar(200)	Max length :200	Job Title
MonthlySalary	varchar(120)	Max length :120	Monthly Salary
JobDescriptions	Midiumtaxt	Not null	Job Descriptions
NoofOpenings	varchar(50)	Max length :50	No Of Openings
JobLocation	varchar(120)	Max length :120	Job Location
ApplyDate	varchar(120)	Max length :120	Applied date
LastDate	varchar(120)	Max length :120	Last Date
JobpostingDate	Datetime	DEFAULT_GENERATED	CURRENT_TIMESTAMP

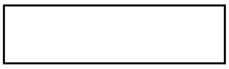
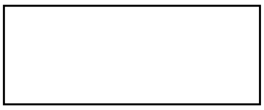
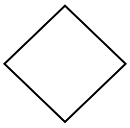
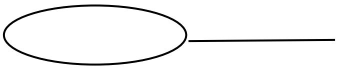
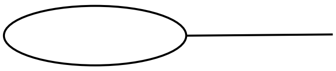
4.2.8. page(tblpage)

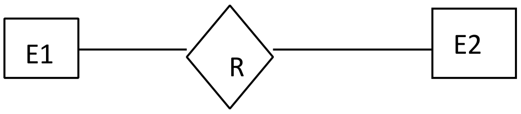
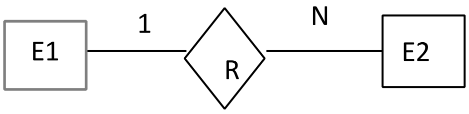
Field	Type	Constraints	Description
ID	Int	Not Null, Primary KEY, auto Increment	Unique ID
PageType	varchar(120)	Max length :120	Type of page
PageTitle	varchar(200)	Max length :200	Page Title
PageDescription	Midiumtext	mediumtext	Page Description
Email	varchar(120)	Max length :120	Email Address
MobileNumber	Bigint	Unique	Mobile Number
Updation Date	Timestamp	DEFAULT_GENERATED	CURRENT_TIMESTAMP

4.3 Entity Relation Model:

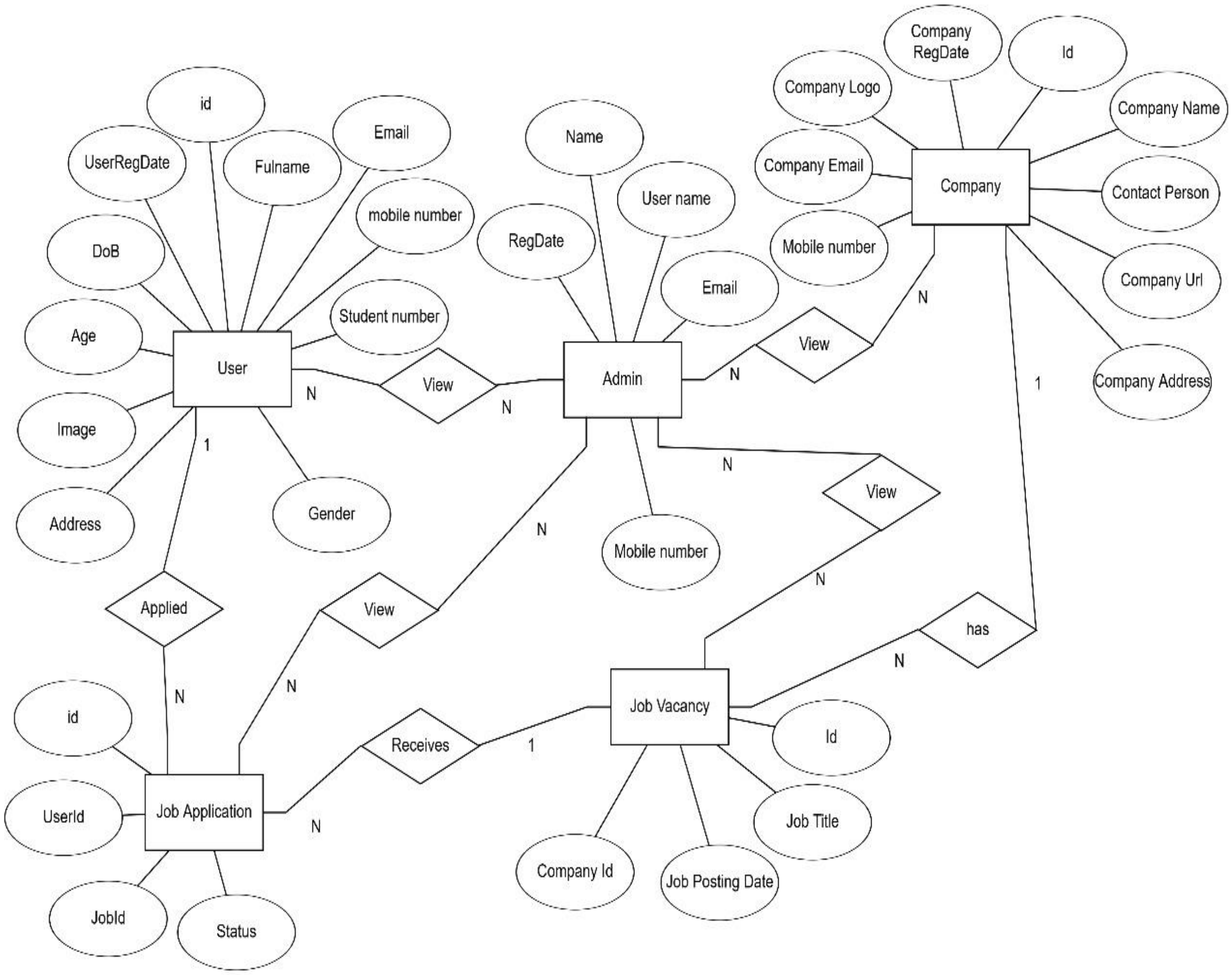
A diagrammatical representation of relationship between the entity and its attribute is refer as an ER relationship model he’s a popular conceptual data model. The relationship among the table is defined via ER diagram. An entity may be an object with physical or conceptual existence. The properties that are used to describe the entity that are called attributes. The entity that does not have key attribute of their own are called week entity. The relationship type that relates a week entity to is owner is call identifying relationship of week entity type. A weak entity always has total participation constraints with respect to its identifying relation. The entities are represented as rectangle, attributes are shown as oval that are connected to entity ER diagram also shown various key attributes in each entity. The rhombus shows the relationship between different entities. The cardinality in ratio specifies the relationship instance the in participated in some convention used while designing the ER diagram the ER diagram are shown below.

4.3.1. Symbolic notation and their description:

Symbols	**Meaning/Convention**
	Entity
	Weak Entity
	Relation
	Attribute
S 	Key Attribute

	Cardinality ratio 1:1 for E1:E2 in R
	Cardinality ratio 1:N for E1:E2 in R

4.3.2 ER Diagram:



Coding

5.Coding

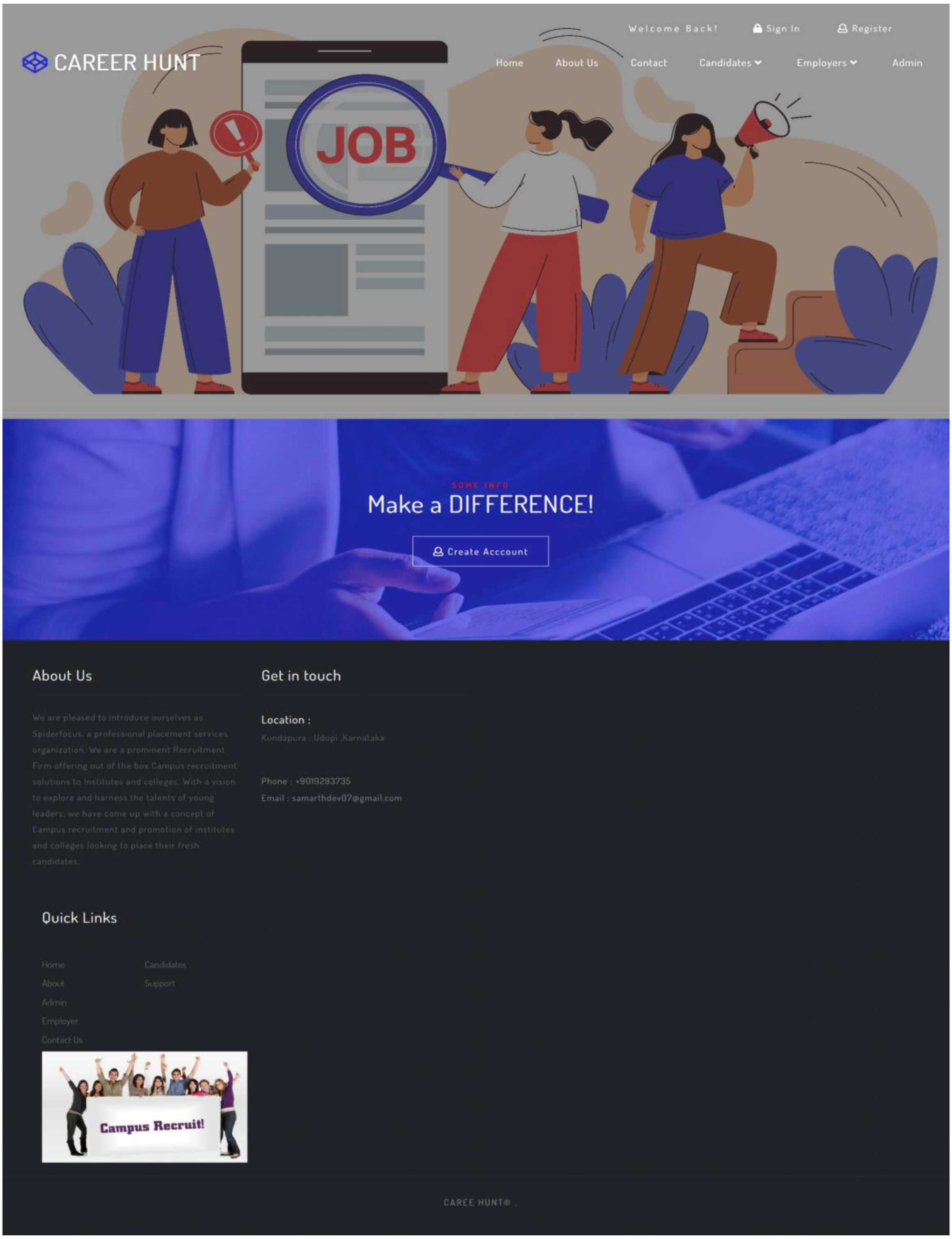
```
<?php
session_start();
error_reporting(0);
include('includes/dbconnection.php');
?>
<!DOCTYPE html>
<html lang="zxx">

<head>
    <title>CAREER HUNT|| Home Page</title>

    <script>
        addEventListener("load", function() {
            setTimeout(hideURLbar, 0);
        }, false);

        function hideURLbar() {
            window.scrollTo(0, 1);
        }
    </script>
    <link href="css/bootstrap.css" rel='stylesheet' type='text/css' />
    <link href="css/zoomslider.css" rel='stylesheet' type='text/css' />
    <link href="css/style6.css" rel='stylesheet' type='text/css' />
    <link href="css/style.css" rel='stylesheet' type='text/css' />
    <link href="css/fontawesome-all.css" rel="stylesheet">
    <link
href="//fonts.googleapis.com/css?family=Dosis:200,300,400,500,600,700"
rel="stylesheet">
    <link href="//fonts.googleapis.com/css?family=Quicksand:300,400,500,700"
rel="stylesheet">
</head>


<body>
    <!-- banner-inner -->
    <div id="demo-1" data-zs-src='["image1.jpg"]'>
        <div class="demo-inner-content">
            <div class="header-top">
               <?php include_once('includes/header.php');?>
            </div>
            </div>
    </div>
    <!-- banner-text -->




    <!--job -->
    <section class="banner-bottom-wthree mid py-lg-5 py-3">
        <div class="container">
            <div class="inner-sec-w3ls py-lg-5  py-3">
                <div class="mid-info text-center pt-3">
                    <h3 class="tittle text-center cen mb-lg-5 mb-3">
                        <span>Some Info</span>Make a DIFFERENCE!</h3>
                    <p></p>
                    <div class="resume">
                        <a href="user/user-signup.php">
                            <i class="far fa-user"></i> Create Acccount</a>
                    </div>
                </div>

            </div>
        </div>
    </section>
    <!--//job -->

    <!--footer -->
    <?php include_once('includes/footer.php');?>
    <!-- //footer -->

    <!--/slider-->
    <script src="js/jquery-1.11.1.min.js"></script>
    <script src="js/modernizr-2.6.2.min.js"></script>
    <script src="js/jquery.zoomslider.min.js"></script>
    <!--//slider-->
    <!--search jQuery-->
    <script src="js/classie-search.js"></script>
    <script src="js/demo1-search.js"></script>
    <!--//search jQuery-->

    <script>
        $(document).ready(function() {
            $(".dropdown").hover(
              function() {
                $('.dropdown-menu', this).stop(true, true).slideDown("fast");
                $(this).toggleClass('open');
              },
              function() {
                $('.dropdown-menu', this).stop(true, true).slideUp("fast");
                $(this).toggleClass('open');
              }
            );
          });
        </script>
        <!-- //dropdown nav -->

        <!-- password-script -->
        <script>
          window.onload = function() {
            document.getElementById("password1").onchange = validatePassword;
            document.getElementById("password2").onchange = validatePassword;
          }

          function validatePassword() {
            var pass2 = document.getElementById("password2").value;
            var pass1 = document.getElementById("password1").value;
            if (pass1 != pass2)
              document.getElementById("password2").setCustomValidity("Passwo
    rds Don't Match");
            else
          document.getElementById("password2").setCustomValidity(");
        //empty string means no validation error
      }
    </script>
    <!-- //password-script -->

    <!-- stats -->
    <script src="js/jquery.waypoints.min.js"></script>
    <script src="js/jquery.countup.js"></script>
    <script>
      $('.counter').countUp();
    </script>
    <!-- //stats -->

    <!-- //js -->
    <script src="js/bootstrap.js"></script>
    <!--/ start-smoth-scrolling -->
    <script src="js/move-top.js"></script>
    <script src="js/easing.js"></script>
    <script>
      jQuery(document).ready(function($) {
        $(".scroll").click(function(event) {
          event.preventDefault();
          $('html,body').animate({
            scrollTop: $(this.hash).offset().top
          }, 900);
        });
```

```
        });
    </script>
    <script>
        $(document).ready(function() {
            /*
                                var defaults = {
                                    containerID: 'toTop', // fading element id
                                  containerHoverID: 'toTopHover', // fading element 
hover id
                                  scrollSpeed: 1200,
                                  easingType: 'linear' 
                                 };
                                */


            $().UItoTop({
                easingType: 'easeOutQuart'
            });


        });
    </script>
    <!--// end-smoth-scrolling -->
</body>

</html>
```

User Interface

6.User Interface

6.1. Index Page



About Us

We are pleased to introduce ourselves as Spiderfocus, a professional placement services organization. We are a prominent Recruitment Firm offering out of the box Campus recruitment solutions to Institutes and colleges. With a vision to explore and harness the talents of young leaders, we have come up with a concept of Campus recruitment and promotion of institutes and colleges looking to place their fresh candidates.

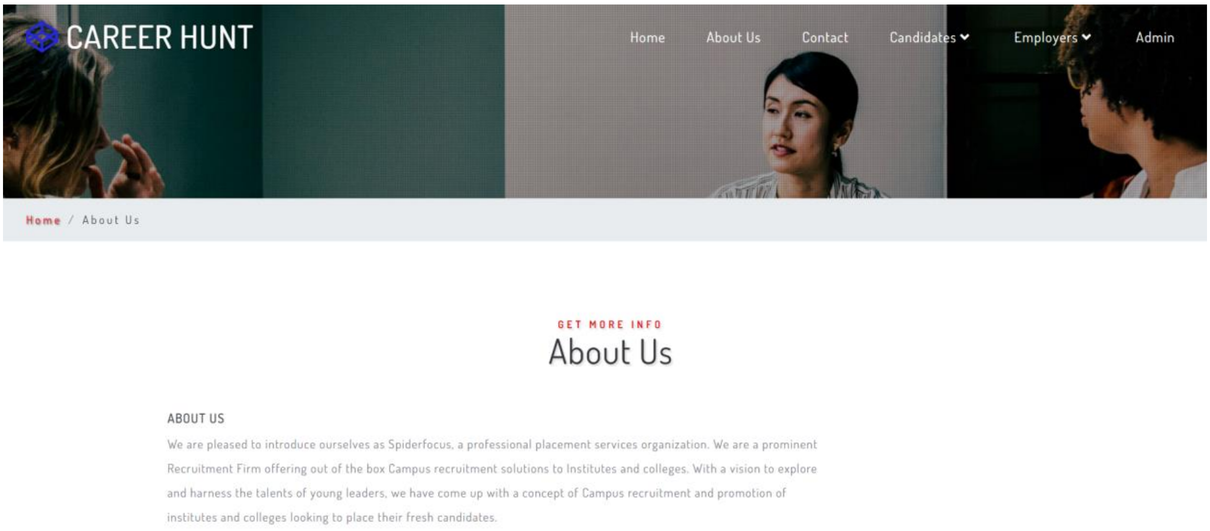
Get in touch

Location :
Kundapura , Udupi ,Karnataka

Phone : +9019293735
Email : samarthdev07@gmail.com

Quick Links

6.2. About Us



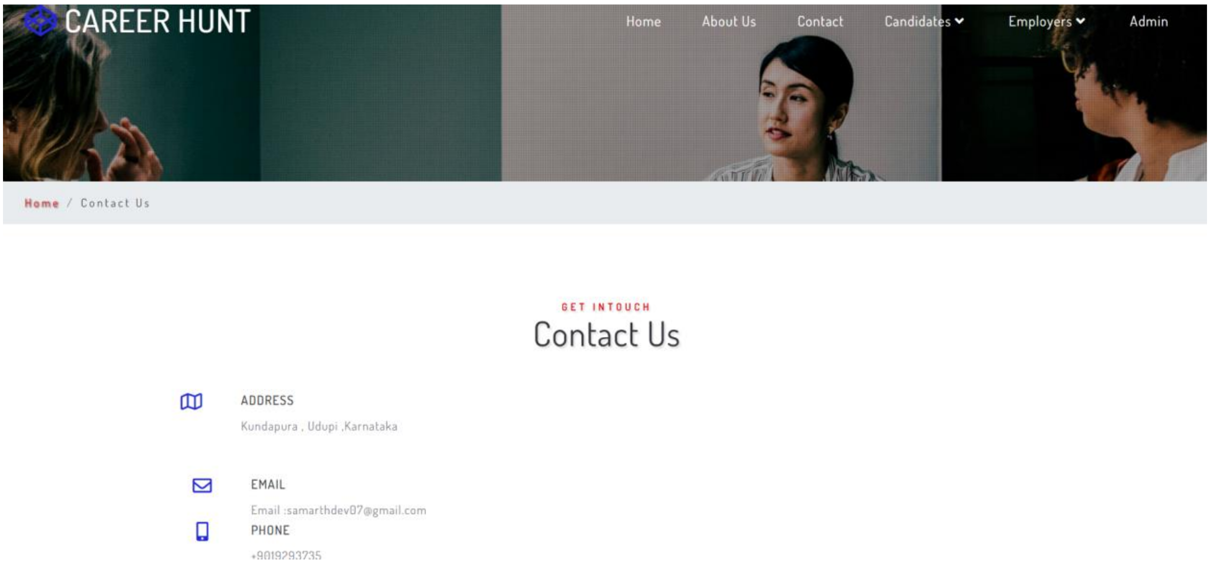
GET MORE INFO

About Us

ABOUT US

We are pleased to introduce ourselves as Spiderfocus, a professional placement services organization. We are a prominent Recruitment Firm offering out of the box Campus recruitment solutions to Institutes and colleges. With a vision to explore and harness the talents of young leaders, we have come up with a concept of Campus recruitment and promotion of institutes and colleges looking to place their fresh candidates.

6.3. Contact us



GET INTOUCH

Contact Us

ADDRESS

Kundapura , Udupi ,Karnataka

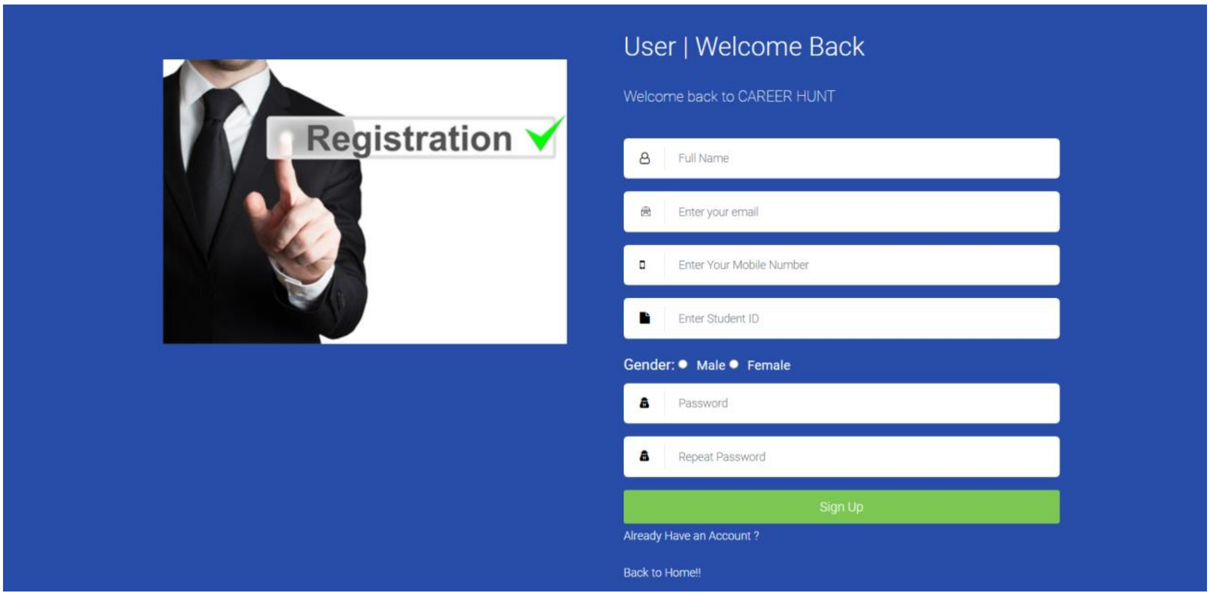
EMAIL

Email :samarthdev07@gmail.com

PHONE

+9019293735

6.4. Candidate sign up



User | Welcome Back

Welcome back to CAREER HUNT

Full Name

Enter your email

Enter Your Mobile Number

Enter Student ID

Gender: ● Male ● Female

Password

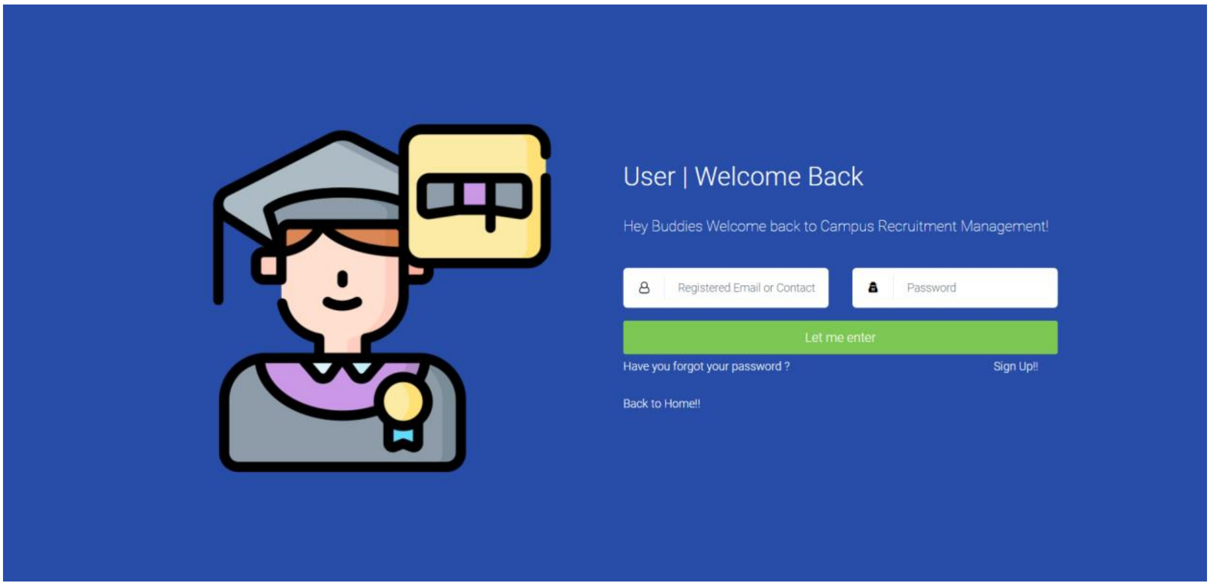
Repeat Password

Sign Up

Already Have an Account ?

Back to Home!!

6.5. Candidate Login



User | Welcome Back

Hey Buddies Welcome back to Campus Recruitment Management!

Registered Email or Contact

Password

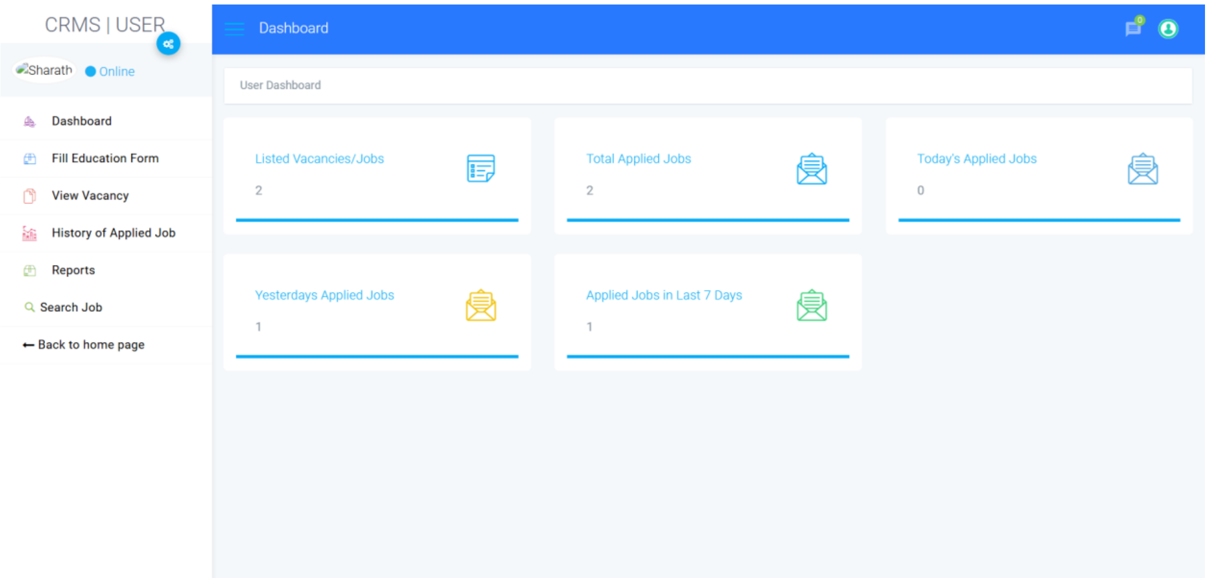
Let me enter

Have you forgot your password ?

Sign Up!!

Back to Home!!

6.6. Candidate Dashboard



CRMS | USER

Sharath Online

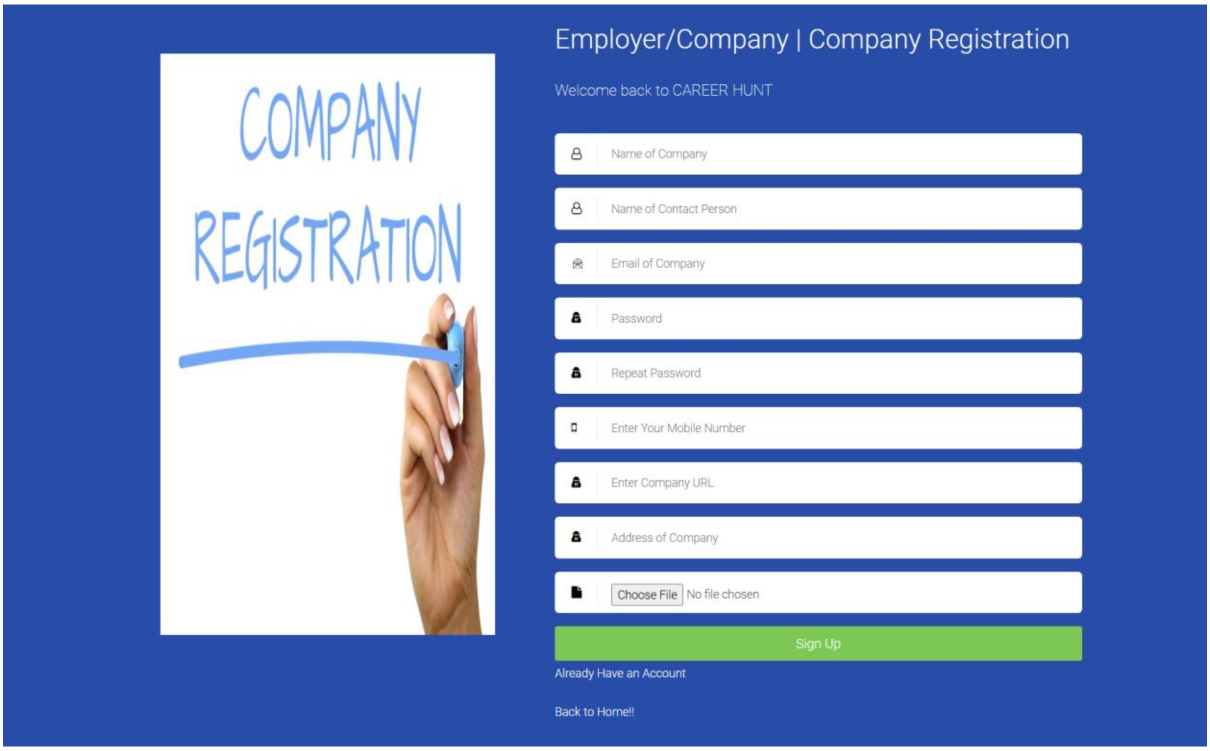
- Dashboard
- Fill Education Form
- View Vacancy
- History of Applied Job
- Reports
- Search Job
- ← Back to home page

Dashboard

User Dashboard

Listed Vacancies/Jobs	2
Total Applied Jobs	2
Today's Applied Jobs	0
Yesterdays Applied Jobs	1
Applied Jobs in Last 7 Days	1

6.7. Company signup



Employer/Company | Company Registration

Welcome back to CAREER HUNT

Name of Company

Name of Contact Person

Email of Company

Password

Repeat Password

Enter Your Mobile Number

Enter Company URL

Address of Company

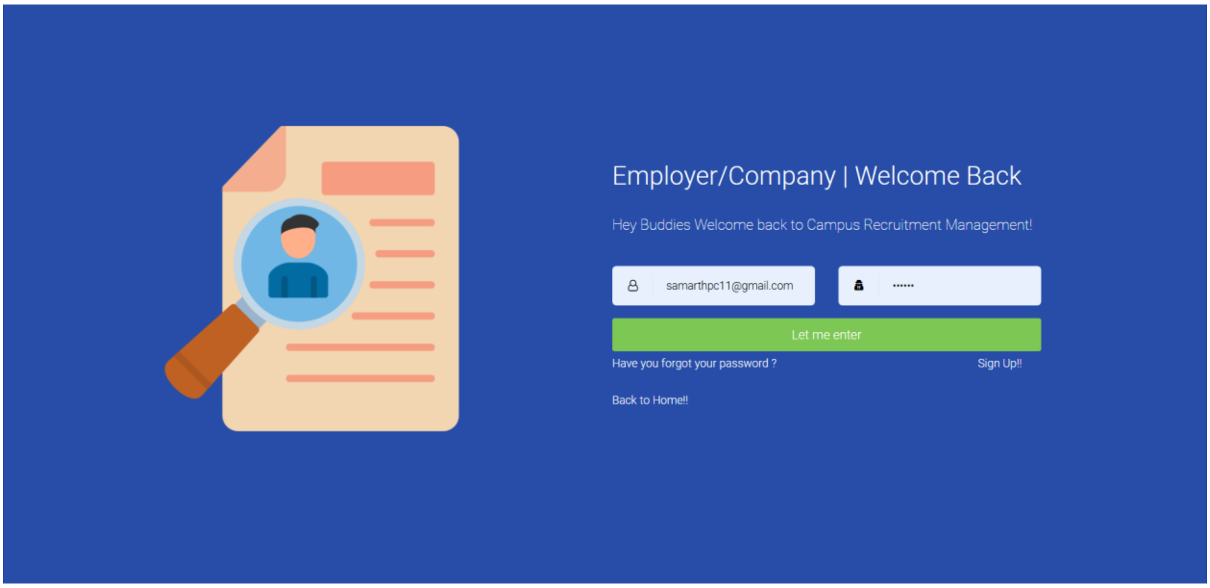
Choose File No file chosen

Sign Up

Already Have an Account

Back to Home!!

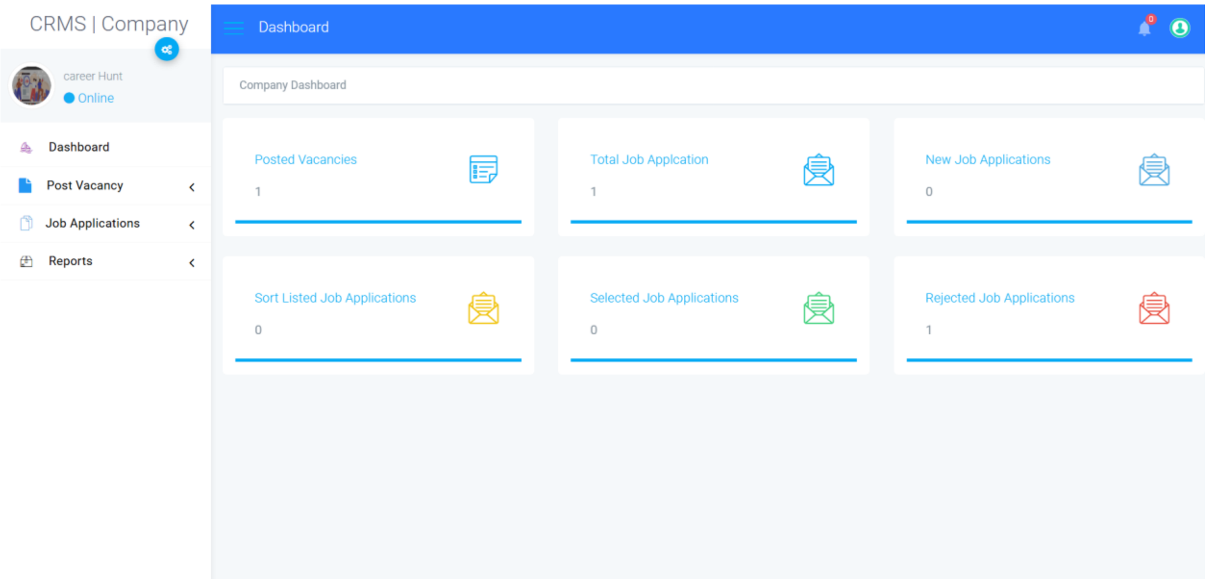
6.8. Company login



Employer/Company | Welcome Back

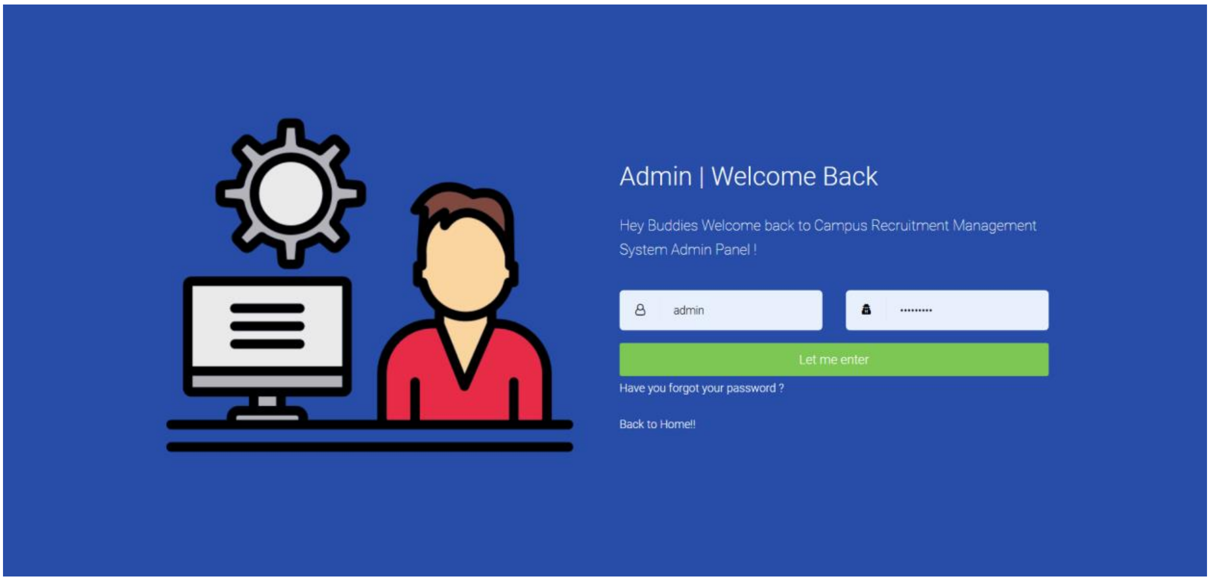
Hey Buddies Welcome back to Campus Recruitment Management!

6.9. Company Dashboard



Posted Vacancies	1
Total Job Applcation	1
New Job Applications	0
Sort Listed Job Applications	0
Selected Job Applications	0
Rejected Job Applications	1

6.10. Admin login



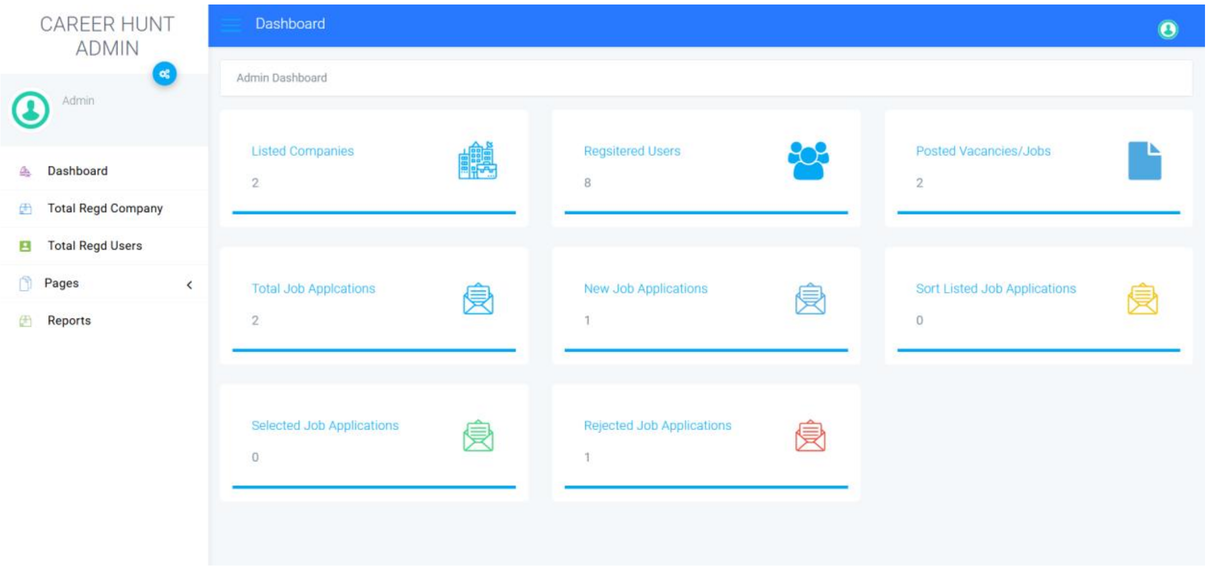
Admin | Welcome Back

Hey Buddies Welcome back to Campus Recruitment Management System Admin Panel !

Have you forgot your password ?

Back to Home!!

6.11. Admin Dashboard



CAREER HUNT ADMIN

Admin

Listed Companies 2

Regsitered Users 8

Posted Vacancies/Jobs 2

Total Job Applcations 2

New Job Applications 1

Sort Listed Job Applications 0

Selected Job Applications 0

Rejected Job Applications 1

Tools and Technologies

7. Tools and Technologies

7. 1. Frontend Technologies

7.1.1 Core Technologies

-7.1.2.HTML

- Semantic markup
- Form elements
- Responsive design structure

7.1.3.CSS3

- Styling and layout
- Responsive design
- Animations and transitions
- Bootstrap framework integration

7.1.4. Java Script

- Form validation
- Dynamic content loading
- User interaction handling

7.1.5. Frameworks and Libraries

7.1.6 Bootstrap

- Responsive grid system
- UI components
- Pre-built CSS classes
- Mobile-first approach

7.1.7. jQuery

- DOM manipulation
- Event handling
- AJAX requests
- Animation effects

7.2. Backend Technologies

7.2.1 Server-Side Language

7.2.2 PHP

- Version: 7.4 or higher
- Server-side scripting
- Database operations
- Session management
- File handling

7.3 Database

-7.3.1 MySQL

- Relational database management
- Data storage and retrieval
- User authentication
- Database management

7.4 Web Server

-Apache

- HTTP server
- URL rewriting

- SSL/TLS support
- Virtual hosting

7.6 Development Tools

7.6.1 Visual Studio Code

- Code editing
- Debugging
- Git integration
- Extensions support

7.6.2 XAMPP

- Local development environment
- Apache server
- MySQL database
- PHP environment

Testing

8. TESTING

8.1 Introduction

Testing is the major quality control measures and during the software development it is used to detect errors that could have occurred during any of the phase like requirement analysis, design, coding. The goal of the testing is to uncover errors in the program.

8.2 Levels of Testing

Testing is done in different levels which includes the following.

- Unit Testing

In Unit testing each module gets tested during the coding phase itself. The purpose is to exercise the different parts of the module code to detect the coding errors.

- Integration Testing

After new testing the modules are gradually integrated into sub systems. It is performed to detect design errors by focusing on testing the interconnection between modules.

- System Testing

System is tested against the system requirement if all the requirements are met and if the system performs as specified by the requirement

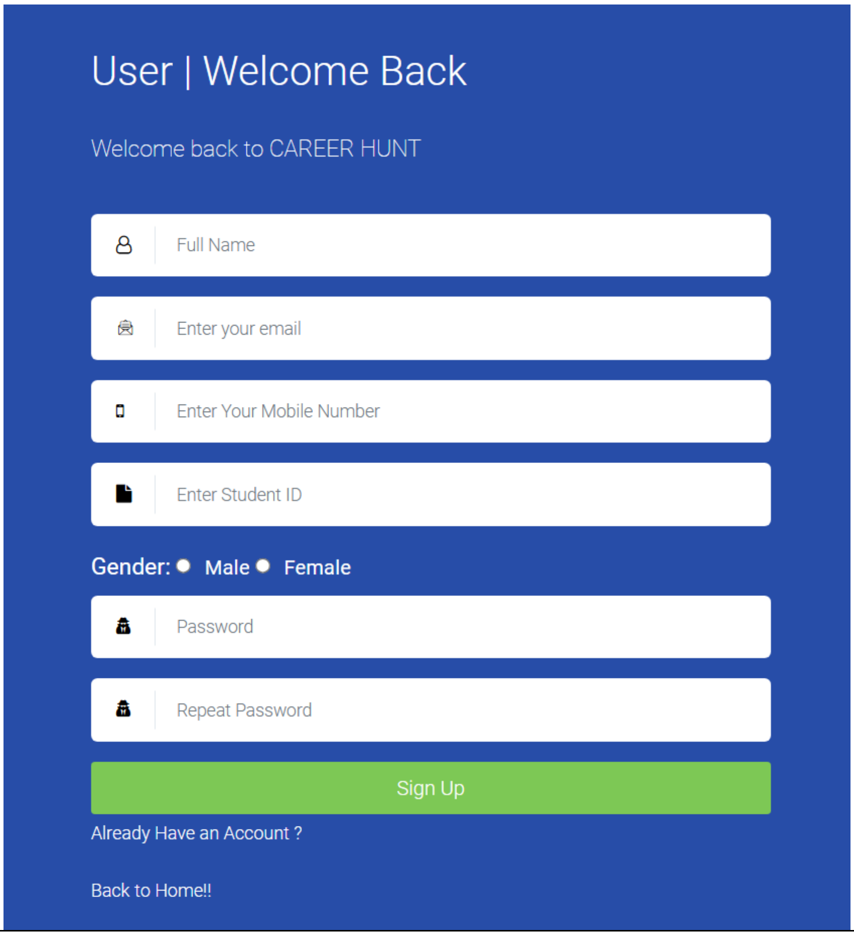
- Acceptance Testing

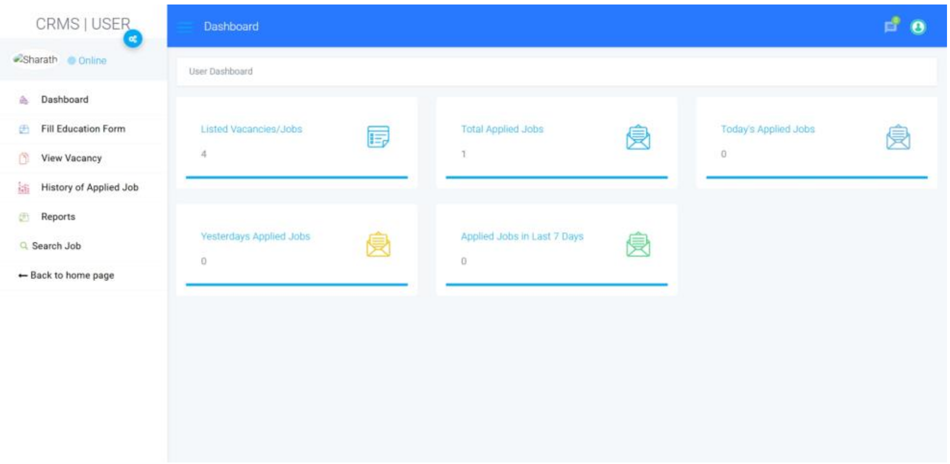
It is performed to demonstrate to the client on real life data of the client, the operation of the system.

8.3 Test Case

It is the input that tests the genuineness of the program, and successful execution of the test case revels that there are no errors in the program that is under testing. It is a set of conditions or variables under which tester will determine whether an application or software is working correctly.

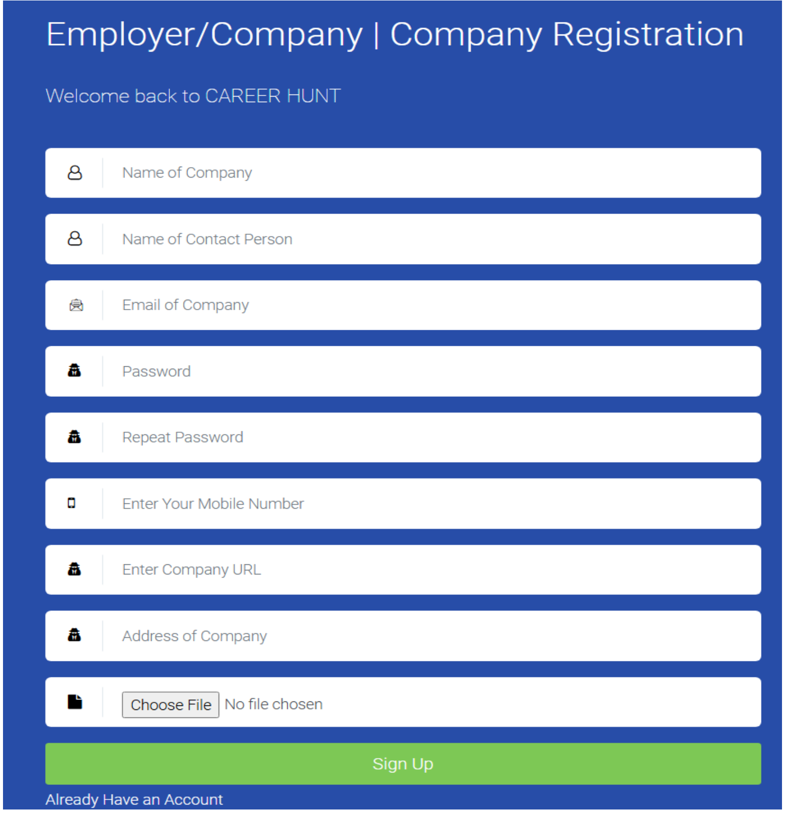
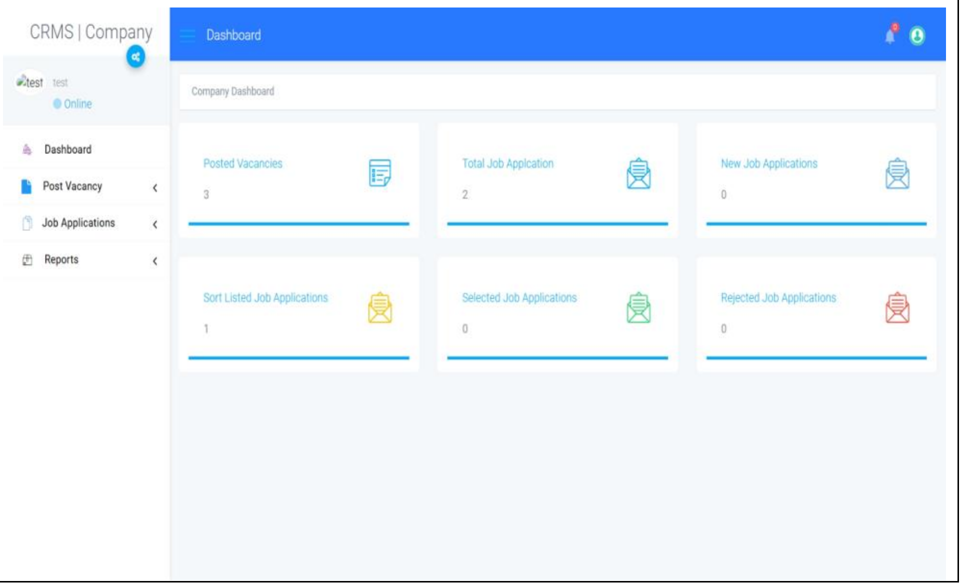
8.3.1. Authorization and Authentication(user)

Test Case ID	01
Test Case Title	To verify that a new user can successfully register through the registration form
Test Data	Full Name, Email, Mobile Number, Student ID, Gender, Password.
Preconditions	User is on the registration page
Steps	1.Navigate to User Registration page. 2.Enter Full Name 3.Enter Email 4.Enter Mobile Number 5.Enter Student ID 6.Choose Gender 7.Enter Password 8.Enter Repeat Password 9.Click on sign up
Expected Result	Valid Input:  User | Welcome Back Welcome back to CAREER HUNT Full Name Enter your email Enter Your Mobile Number Enter Student ID Gender: ● Male ● Female Password Repeat Password Sign Up Already Have an Account ? Back to Home!!

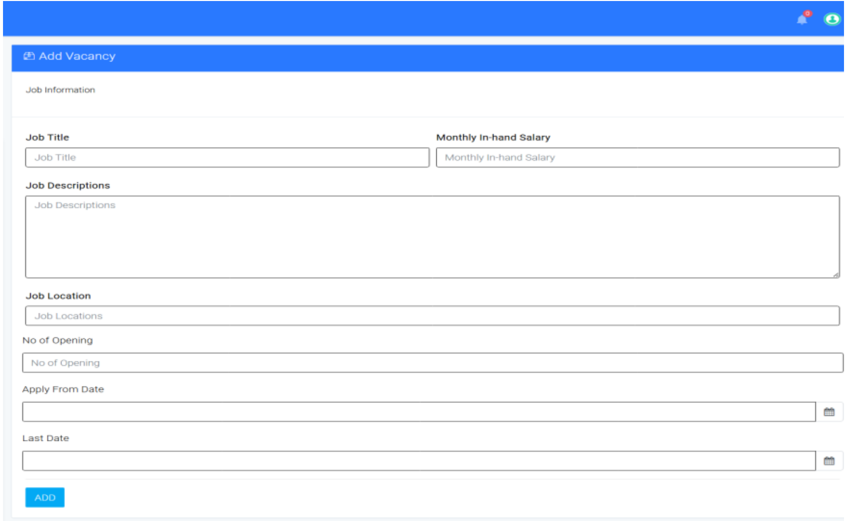
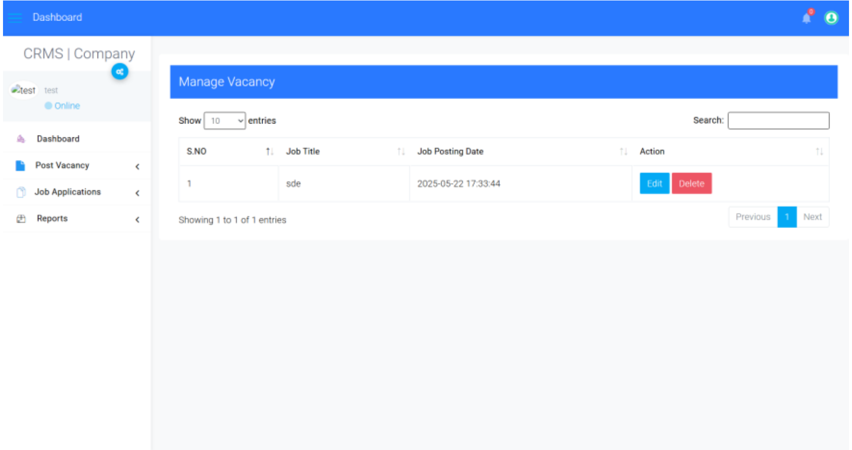
	Valid Output  CRMS | USER Sharath Online Dashboard Fill Education Form View Vacancy History of Applied Job Reports Search Job ← Back to home page Dashboard User Dashboard Listed Vacancies/Jobs 4 Total Applied Jobs 1 Today's Applied Jobs 0 Yesterdays Applied Jobs 0 Applied Jobs in Last 7 Days 0

8.3.2. Authorization and Authentication(company)

Test Case ID	02
Test Case Title	To verify that a new user can successfully register through the registration form
Test Data	Company Name, Contact Person, Email, Password, Mobile Number, Company URL, Logo.
Preconditions	User is on the registration page
Steps	1.Navigate to User Registration page. 2.Enter Company Name 3.Enter Contact Person 4.Enter Email of company 5.Enter Password 6.Enter Repeat Password 7.Enter Mobile Number 8.Enter Company URL 9.Enter Address of company 10.Enter the Logo 11.Click on sign up

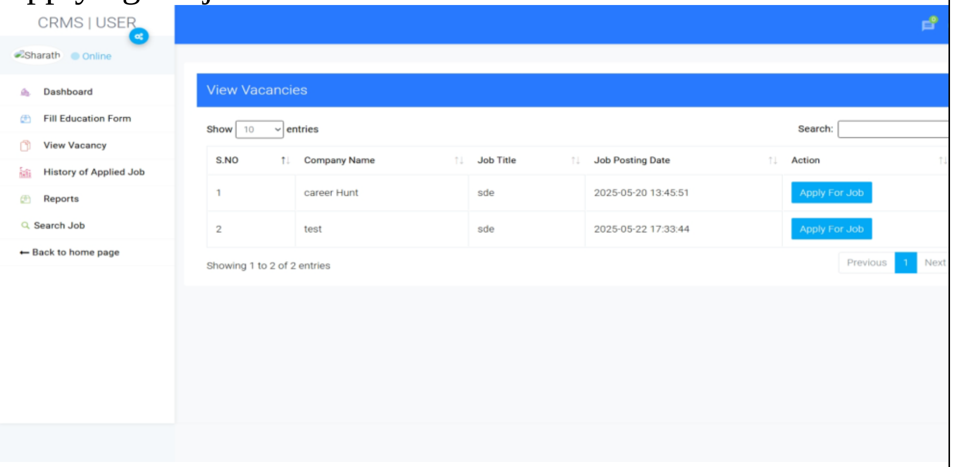
Expected Result	Valid Input:  Employer/Company | Company Registration Welcome back to CAREER HUNT Name of Company Name of Contact Person Email of Company Password Repeat Password Enter Your Mobile Number Enter Company URL Address of Company Choose File No file chosen Sign Up Already Have an Account
	Valid Output  CRMS | Company test test Online Dashboard Post Vacancy Job Applications Reports Dashboard Company Dashboard Posted Vacancies 3 Total Job Application 2 New Job Applications 0 Sort Listed Job Applications 1 Selected Job Applications 0 Rejected Job Applications 0

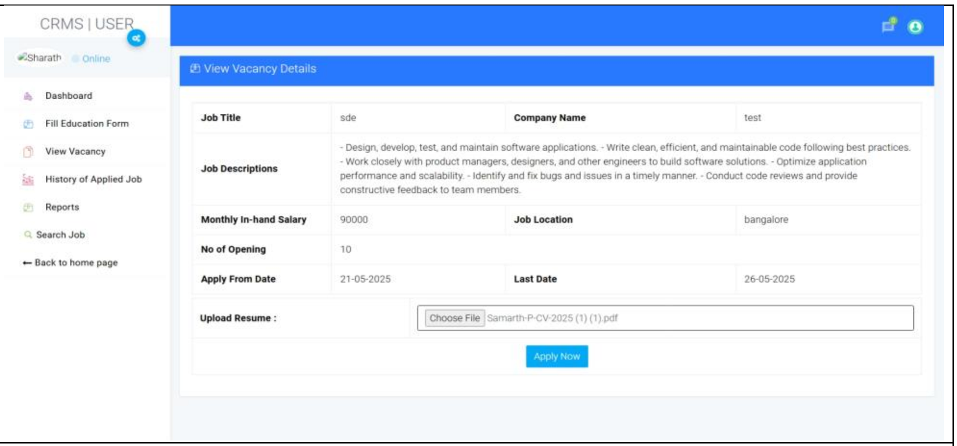
8.3.2. Job Posting

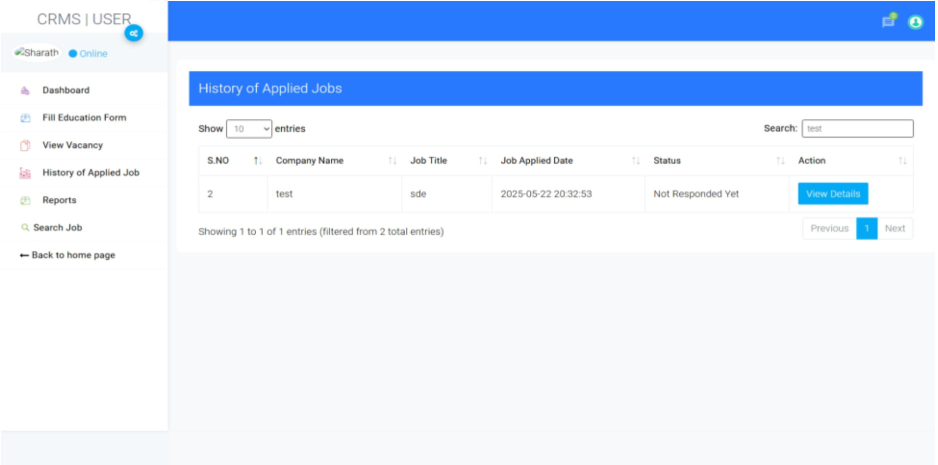
Test Case Id	03
Test Case Title	Job Posting
Purpose of Testing	To check the Job Posting OR not
Preconditions	1. Navigate to Post Vacancy 2. Add Job Title 3. Enter monthly salary 4. Enter Job Description 5. Enter Job Location 6. Enter the number of opening. 7. Enter Apply Form Date 8. Enter Last Date. 9. Enter Add button
Expected Result	Valid Input  Add Vacancy Job Information Job Title: Job Title Monthly In-hand Salary: Monthly In-hand Salary Job Descriptions: Job Descriptions Job Location: Job Locations No of Opening: No of Opening Apply From Date: Last Date: ADD
	Valid Output  Dashboard CRMS | Company test Online Dashboard Post Vacancy Job Applications Reports Manage Vacancy Show 10 entries Search: S.NO | Job Title | Job Posting Date | Action 1 | sde | 2025-05-22 17:33:44 | Edit Delete Showing 1 to 1 of 1 entries Previous 1 Next
Negative Test Cases	is apply date is greater than last date → show validation error

	If required fields are empty → show appropriate error message If date is not selected → show validation error If word count exceeds max capacity → show warning
Postconditions	If job posting is sucessful it is visible to recruiter in manage vacancy and for users in the show all vacancy section.

8.3.3. Applying for Job

Test Case Id	04
Test Case Title	Applying for job
Purpose of Testing	To check the it is applying OR not
Preconditions	1. Navigate view Vacancy 2. Add Resume 3. Enter apply for job button
Expected Result	Applying for job  CRMS | USER Sharath Online Dashboard Fill Education Form View Vacancy History of Applied Job Reports Search Job ← Back to home page View Vacancies Show 10 entries Search: S.NO | Company Name | Job Title | Job Posting Date | Action 1 | career Hunt | sde | 2025-05-20 13:45:51 | Apply For Job 2 | test | sde | 2025-05-22 17:33:44 | Apply For Job Showing 1 to 2 of 2 entries Previous 1 Next Valid Input

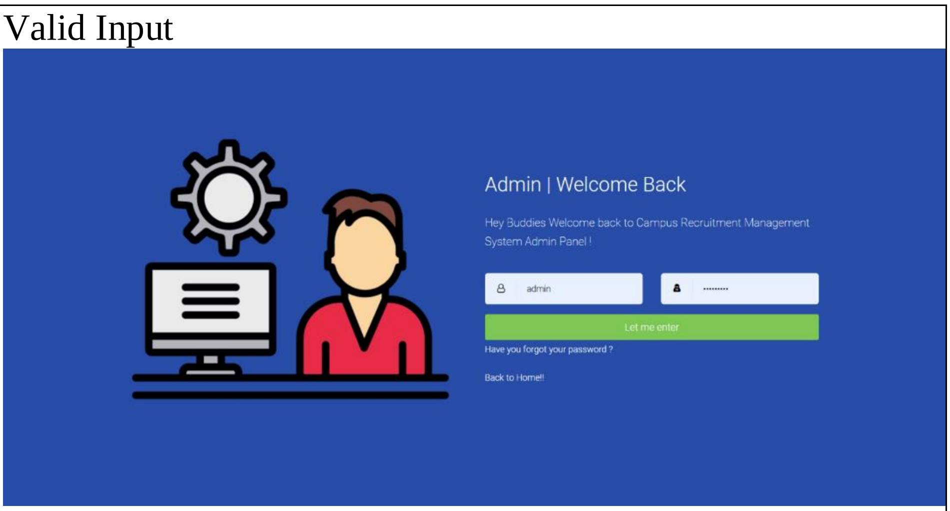
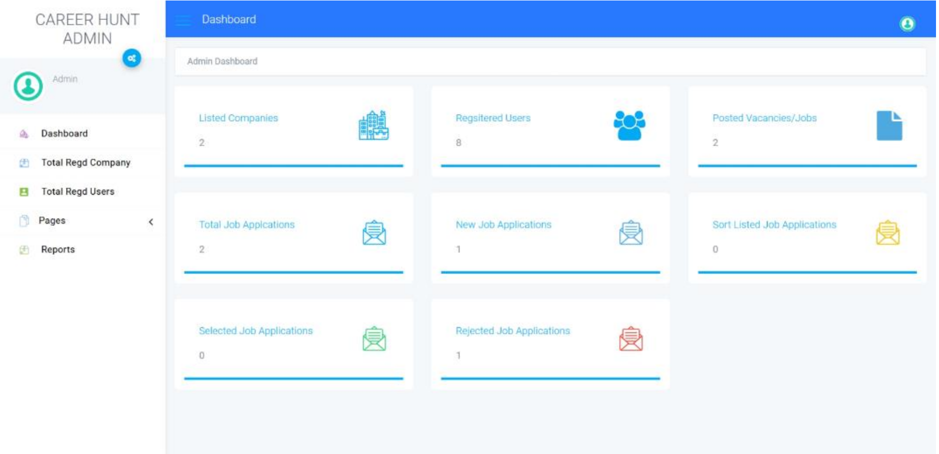
<table>
<tr>
<td></td>
<td>
CRMS | USER
Sharath Online
Dashboard
Fill Education Form
View Vacancy
History of Applied Job
Reports
Search Job
← Back to home page
View Vacancy Details
<table>
<tr><td>Job Title</td><td>sde</td><td>Company Name</td><td>test</td></tr>
<tr><td>Job Descriptions</td><td colspan="3">- Design, develop, test, and maintain software applications. - Write clean, efficient, and maintainable code following best practices. - Work closely with product managers, designers, and other engineers to build software solutions. - Optimize application performance and scalability. - Identify and fix bugs and issues in a timely manner. - Conduct code reviews and provide constructive feedback to team members.</td></tr>
<tr><td>Monthly In-hand Salary</td><td>90000</td><td>Job Location</td><td>bangalore</td></tr>
<tr><td>No of Opening</td><td colspan="3">10</td></tr>
<tr><td>Apply From Date</td><td>21-05-2025</td><td>Last Date</td><td>26-05-2025</td></tr>
<tr><td>Upload Resume :</td><td colspan="3">Choose File Samarth-P-CV-2025 (1) (1).pdf</td></tr>
</table>

</td>
</tr>
<tr>
<td></td>
<td>Valid Output

CRMS | USER
Sharath Online
Dashboard
Fill Education Form
View Vacancy
History of Applied Job
Reports
Search Job
← Back to home page
History of Applied Jobs
Show 10 entries Search: test
<table>
<tr><th>S.NO</th><th>Company Name</th><th>Job Title</th><th>Job Applied Date</th><th>Status</th><th>Action</th></tr>
<tr><td>2</td><td>test</td><td>sde</td><td>2025-05-22 20:32:53</td><td>Not Responded Yet</td><td>View Details</td></tr>
</table>
Showing 1 to 1 of 1 entries (filtered from 2 total entries)

</td>
</tr>
<tr>
<td>Negative Test Cases</td>
<td>is apply date is greater than last date → show validation error
If required fields are empty → show appropriate error message</td>
</tr>
<tr>
<td>Postconditions</td>
<td>If apply was successful it Goes to history of applied job</td>
</tr>
</table>

8.3.3. Admin Login

Test Case Id	05
Test Case Title	Admin Login
Purpose of Testing	To check the admin is able to login or not
Preconditions	1. Navigate Admin 2. Add User Name and Password 3. Enter Login

Expected Result	Valid Input  Admin | Welcome Back Hey Buddies Welcome back to Campus Recruitment Management System Admin Panel ! admin ········· Let me enter Have you forgot your password ? Back to Home!!
	Valid Output  CAREER HUNT ADMIN Admin Dashboard Total Regd Company Total Regd Users Pages Reports Dashboard Admin Dashboard Listed Companies 2 Regsitered Users 8 Posted Vacancies/Jobs 2 Total Job Applications 2 New Job Applications 1 Sort Listed Job Applications 0 Selected Job Applications 0 Rejected Job Applications 1
Negative Test Cases	is admin name and password is wrong → show error If required fields are empty → show appropriate error message
Postconditions	If apply was successful it Goes to Admin Dash Board

Future Scope

9.Future scope

The Enhanced User Experience system, with its core functionalities of mobile application development and interactive elements, has a solid foundation for future expansion. Future enhancements could include:

- Mobile Application Development:
  Creating dedicated mobile applications for iOS and Android would provide greater accessibility and convenience. Key features could include push notifications, offline functionality, and mobile-specific enhancements.
- Interactive User Elements:
  Improving the user experience through drag-and-drop functionality, real-time form validation, animated transitions, and responsive design improvements would create a seamless and engaging interface.

The Artificial Intelligence Integration system, with its capabilities in smart job matching and automated screening, has strong potential for advancement. Future enhancements could include:

- AI-Powered Job Matching:
  Enhancing resume parsing capabilities, skill-based matching, job recommendations, and candidate ranking would refine the efficiency and accuracy of the hiring process.
- Automated Screening & AI Interview System:
  Implementing AI-driven video interviews, automated assessment mechanisms, skill evaluations, and personality analysis would streamline the screening process, ensuring a more precise candidate selection.
- Chatbot Integration:
  Expanding 24/7 candidate support through AI-driven automated responses, FAQ handling, and application assistance would enhance user engagement and responsiveness.

The Advanced Analytics system, focusing on business intelligence and advanced reporting, could be enhanced through:

- Custom Reporting & Data Visualization:
  Improved reporting mechanisms with predictive modelling, trend analysis, and refined visualization techniques could offer deeper insights into performance metrics and business strategy.

The Enhanced Security Features system, with advanced authentication and data protection mechanisms, is crucial for safeguarding user data. Future improvements could include:

- Multi-Factor Authentication:
  Strengthening security through biometric authentication, OTP verification, SSO integration, and role-based access would enhance data protection.
- Advanced Encryption & Data Protection:
  Enhancing encryption protocols with end-to-end encryption, secure file transfers, data masking, and privacy protection would ensure data confidentiality.

The Advanced Features system, centered on video interview platforms and assessment tools, could be further refined through:

- Virtual Interview Systems:
  Supporting live video interviews, recorded interviews, interview scheduling, and feedback mechanisms would create a robust digital hiring platform.
- Online Assessment Platform:
  Incorporating skill testing, psychometric evaluations, and coding challenges would enhance candidate assessment accuracy.

Limitations

10.Limitations

The **System Performance** module, while functional, faces notable scalability and database constraints that could limit efficiency in high-demand scenarios. Future improvements could include:

- **Optimized Scalability & Performance:**
  Enhancing concurrent user handling, reducing response times, and optimizing resource utilization would ensure stability during peak usage.
- **Database Enhancements:**
  Expanding storage capacity, improving query performance for large datasets, and refining backup and recovery processes would strengthen reliability.

The **Technical Infrastructure**, with limitations in server deployment and security constraints, has room for enhancement. Key areas for future development include:

- **Server Performance Optimization:**
  Implementing cloud-based solutions, load balancing mechanisms, and enhanced bandwidth availability would improve processing capabilities.
- **Security Advancements:**
  Strengthening encryption protocols, refining authentication systems, and upgrading firewall protections would bolster data security.

The **Functional Limitations** in user interface design, user experience, and data exchange could be addressed through advanced feature integration:

- **UI & UX Enhancements:**
  Implementing a more dynamic and responsive design, introducing interactive elements, and expanding customization options would improve usability.
- **Improved Data Exchange Capabilities:**
  Strengthening import/export functionalities, enhancing data migration tools, and supporting bulk operations would streamline efficiency.

The **Security Limitations** module currently offers basic authentication and data protection, but future refinements could include:

- **Enhanced Authentication Protocols:**
  Introducing advanced multi-factor authentication, refining session management, and expanding role-based access controls would strengthen security.
- **Data Encryption & Storage Improvements:**
  Implementing robust encryption methods, secure backups, and improved privacy controls would ensure data integrity.

The **Mobile Limitations**, with restricted functionality and limited technology updates, could be expanded through:

- **Mobile Optimization & Features:**
  Enhancing responsive design, incorporating offline access capabilities, and improving mobile notifications would enhance the mobile experience.
- **Advanced Mobile Security & Updates:**
  Refining security measures, supporting location services, and ensuring regular framework updates would maintain system compatibility.

Conclusion

11.Conclusion

The Campus Recruitment Management System (CAREER HUNT) was successfully implemented using PHP and MySQL, along with modern web technologies such as HTML5, CSS3, JavaScript, and Bootstrap. This system efficiently handles the recruitment process through key features such as user registration, job posting, application management, and email notifications. Throughout the development, several challenges arose, including secure user authentication, real-time application status updates, email system integration, database optimization, and mobile responsiveness. These were effectively addressed through secure password hashing, input validation, efficient database queries, reliable email delivery mechanisms, and responsive design principles, ensuring smooth system functionality.

This project has provided invaluable insights into multiple aspects of web development, including user authentication systems, database design and management, email notification mechanisms, form validation techniques, and security best practices. Additionally, the development process fostered proficiency in PHP programming, frontend development using Bootstrap, MySQL database management, project documentation, testing methodologies, UI/UX principles, version control with Git, and API integration. By implementing these skills, the system successfully meets its primary objectives: providing a secure platform for user registration and authentication, enabling efficient job posting and application management, facilitating seamless communication between recruiters and candidates, ensuring high data security standards, and delivering a user-friendly experience for all stakeholders.

To expand its capabilities, future enhancements could include AI-powered job matching to improve recruitment efficiency, the development of dedicated mobile applications for better accessibility, the implementation of advanced analytics to gain deeper insights into hiring trends, enhanced security features to protect user data further, and integration with third-party job portals to streamline the recruitment process. These upgrades would ensure the system remains adaptive and scalable, catering to the evolving needs of both recruiters and candidates.

In conclusion, the project has not only met but exceeded initial expectations by delivering a comprehensive and scalable Campus Recruitment Management System. With its solid foundation in modern web technologies and industry best practices, the system remains relevant in the dynamic recruitment landscape. Its well-structured design and modular framework allow for future enhancements while maintaining security, usability, and performance standards. By addressing key challenges and incorporating best practices, the system ensures efficiency, making it a valuable tool for educational institutions and recruiters seeking an effective recruitment management solution.

Lawrence Lobo: U05BB22S0093

Ganesh R: U05BB22S0038

Samarth P: U05BB22S0051

Bibliography

9.BIBLIOGRAPHY

1.Boostrap

>https://getbootstrap.com/docs/5.0/getting-started/introduction/

2.MySQL

>https://www.w3schools.com/sql/

3.JQuery

>https://api.jquery.com/

4.W3schools

>https://www.w3schools.com/sql/

5.BlackBox AI

>https://www.blackbox.ai/

5.Software testing

>https://www.geeksforgeeks.org/software-testing-basics/

Book Reference:

>IEEE Standard 830-1998 for Software Requirements Specification