TRAINING AND PLACEMENT WEBSITE

An Industry oriented Mini Project Report Submitted to

Jawaharlal Nehru Technological University, Hyderabad

In partial fulfillment of the requirements
For the award of the degree of

BACHELOR OF TECHNOLOGY
IN
COMPUTER SCIENCE AND ENGINEERING

By

P.SINDHUJA	**(21E11A05O2)**
K.DEEPTHI	**(21E11A05L7)**
J.ALEKHYA	**(21E11A0O8)**
CH.BHAVANI	**(21E11A05L0)**

Under the guidance of

Dr. Rama Prakasha Reddy Ch
Assistant Professor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
BHARAT INSTITUTE OF ENGINEERING AND TECHNOLOGY
(Approved by AICTE, Affiliated to JNTUH Hyderabad, NAAC Accredited, NBA
Accredited for UG Programmes CSE, ECE), Hyderabad – 501510

BHARAT INSTITUTE OF ENGINEERING AND TECHNOLOGY

(Approved by AICTE, Affiliated to JNTUH Hyderabad, NAAC Accredited, NBA Accredited for UG Programmes CSE, ECE), Hyderabad – 501510

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Certificate

*This is to certify that An Industry Oriented Mini project work entitled “*WEB BASED ONLINE STUDY PORTAL*” is the bonafide work done*

By

P.SINDHUJA	**(21E11A05O2)**
K.DEEPTHI	**(21E11A05L7)**
J.ALEKHYA	**(21E11A0O8)**
CH.BHAVANI	**(21E11A05L0)**

In the Department of Computer Science and Engineering, ***BHARAT INSTITUTE OF ENGINEERING AND TECHNOLOGY,*** *Hyderabad is submitted to* ***Jawaharlal Nehru Technological University, Hyderabad*** *in partial fulfillment of the requirements for the award of* ***B.Tech*** *degree in* ***Computer Science and Engineering*** *during* ***2024-2025.***

Guide:

Dr. Rama Prakasha Reddy Ch
Assisstant Professor,
Dept. of CSE,
Bharat Institute of Engineering and Technology,
Hyderabad – 501 510.

Head of the Department:

Dr. Deepak Kachave
Associate Professor & Academic I/C,
Dept. of CSE,
Bharat Institute of Engineering and Technology,
Hyderabad – 501 510.

Viva-Voce held on…………………………………………

Internal Examiner

External Examiner

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of the task would be put incomplete without the mention of the people who made it possible, whose constant guidance and encouragement crown all the efforts with success.

We avail this opportunity to express our deep sense of gratitude and hearty thanks to **Sri CH. Venugopal Reddy,** Secretary & Correspondent of BIET, for providing congenial atmosphere and encouragement.

We would like to thank **Prof. G. Kumaraswamy Rao**, **Director (R&D), Dr. V. Srinivas Rao, Dean CSE,** for having provided all the facilities and support.

We would like to thank **Dr. Deepak Kachave, Associate Professor & Academic I/C, Computer Science and Engineering** for their expert guidance and encouragement at various levels of our Project.

We are thankful to our guide **Dr. Rama Prakasha Reddy Ch** ,**Project coordinator, Computer Science and Engineering** for his sustained inspiring Guidance and cooperation throughout the process of this project. His wise counsel and suggestions were invaluable. We are also thankful to our project coordinator **Dr. Rama Prakasha Reddy Ch** for coordinating project at all levels for completion of the project. (if project is done in industry acknowledge the industry personals).

We express our deep sense of gratitude and thanks to all the **Teaching** and **Non-Teaching Staff** of our college who stood with us during the project and helped us to make it a successful venture.

We place highest regards to our **Parents**, our **Friends** and **Well-wishers** who helped a lot in making the report of this project.

P.SINDHUJA	**(21E11A05O2)**
K.DEEPTHI	**(21E11A05L7)**
J.ALEKHYA	**(21E11A0O8)**
CH.BHAVANI	**(21E11A05L0)**

DECLARATION

We hereby declare that this Industry oriented Mini Project report is titled ***"Web Based Online Study Portal"*** is a genuine project work carried out by us, in **B.Tech (Computer Science and Engineering)** degree course of **Jawaharlal Nehru Technology University Hyderabad, Hyderabad** and has not been submitted to any other course or university for the award of my degree by me.

Signature of the Student's

P.SINDHUJA
K.DEEPTHI
J.ALEKHYA
CH.BHAVANI

ACRONYMS AND DEFINITIONS

S No	ACRONYM	DESCRIPTION
1	HTML	HYPER TEXT MARKUP LANGUAGE
2	CSS	CASSCADING STYLE SHEETS
3	JS	JAVASCRIPT
4		
5		
6		
7		

ABSTRACT

The **Training and Placement Website** is designed to enhance the efficiency of the placement process for both students and administrative staff. Faculty and staff can effortlessly post job openings, internship opportunities, and hackathon details, ensuring students stay informed about a wide range of career-building activities. The platform's intuitive design makes it easy for students to access on-campus and off-campus job listings, explore internships, participate in training sessions, and track their application progress.

This website acts as a bridge between students and career opportunities, offering a centralized hub for all placement-related activities. By simplifying the management of job postings and enabling students to discover diverse employment options, it ensures a seamless and enriching placement experience. Ultimately, the platform fosters collaboration, saves time, and equips students with the tools they need to build a successful career.

keywords

Training and Placement Website, Job Listings Platform, Internship Opportunities, Career Development, Placement Process, On-Campus Placements, Off-Campus Opportunities, Hackathon Announcements, Career Readiness,User-Friendly Platform, Employment Options, Professional Growth, Centralized Career Hub, Faculty and Administrative Tools,Student-Faculty Collaboration,

CONTENTS

INDEX FOR DIAGRAMS

1. INTRODUCTION

1. INTRODUCTION

1.1 Objective of Project

The objective of the **Training and Placement Website** is to provide a user-friendly platform that connects students with career opportunities. It streamlines the placement process by offering easy access to job listings, internships, and training sessions. The platform supports students, faculty, and administrators, allowing them to manage job postings, applications, and placement activities efficiently. Ultimately, it aims to enhance career readiness and simplify the job application process for students.

1.2 Limitations of project

1.2.1Code Limitations

The website may face performance issues as the number of users (students, faculty, and recruiters) and data (job listings, internships, resumes) increases. The amount of code that can be loaded into form class, or standard module is limited to 1000 – 3000 lines in frontend development and 1000 – 3000 in backend development, total estimate lines of code 8000 – 16000 lines . A single line of code can consist of up to 1023 bytes. Up to 256 blank spaces can precede the actual text on a single line, and no more than twenty-four line-continuation characters (_) can be included in a single logical line.

1.2.2Procedures, Types, and Variables

Procedures are methods or functions used to implement specific features of the project .**(Student/Faculty/Admin):** Handles user registration, assigns roles, and stores user details in the database. Faculty or admin can post job opportunities with details like title, description, eligibility, and deadline. Types refer to the data types and structures used in the code. Primitive Types, Composite Types, Custom Data Types. Variables hold data and are categorized based on their usage.

1.2.3Runtime limitations

The **Training and Placement Website** has runtime limitations related to server capacity and bandwidth, which may cause slower performance during peak usage or with large job listings and file uploads. High traffic or complex data may lead to temporary delays. Proper server scaling and optimization are necessary to maintain smooth performance.

1.2.4Unsupported Web.config elements

The following system.web child elements are not supported:

1. Browser Caps (For a workaround, see Browser capabilities support.)
2. Client Target
3. HTTPRuntime
4. Identity
5. SecurityPolicy
6. ProcessModel
7. SessionState (For a workaround, see Session persistency support.)
8. Trust
9. Cookieless sessions are not supported

2. LITERATURE SURVEY

2. LITERATURE SURVEY

2.1 Introduction

The **Training and Placement (TnP) Website** plays a crucial role in connecting students with career opportunities, making it an essential tool for educational institutions. It is designed to support three distinct user roles: Admin, Faculty, and Students, each with specific functionalities to enhance the platform's usability. Faculty members can post various job listings, including both on-campus and off-campus opportunities, ensuring that students are aware of diverse career options and can explore different paths for their future.

Admins hold full control over the platform, allowing them to manage the system by hiring and adding administrative staff as needed. Students, on the other hand, can easily navigate through the platform to access job listings, internships, and other career-related resources. This seamless access ensures that students have up-to-date information and can make well-informed decisions regarding their career prospects. This project thus serves as a comprehensive hub for all career-related activities within the educational institution.

2.2 Existing System

.In most educational institutions, the **Training and Placement (TnP)** process is managed manually or through basic systems that are often fragmented and inefficient. Typically, students rely on notice boards or emails for job updates, and faculty or administrative staff may handle job postings and placement-related activities without a unified platform. Communication between students, faculty, and admin staff is often scattered, leading to delays and confusion in managing job listings, internships, and training opportunities. Additionally, there is often no central repository for students to track their applications, internships, or placement progress.

2.3 Disadvantages of Existing System

.- **Manual Handling**: Delays and errors in job postings.
- **Outdated Information**: Missed opportunities due to lack of real-time updates.
- **Scattered Communication**: Confusion between users due to fragmented channels.
- **Restricted Access**: Limited control for students, faculty, and admins.
- **Lack of Tracking**: No centralized system to monitor applications and progress.

2.4 Proposed System

The **Training and Placement Website** aims to address the limitations of the existing system by providing a centralized, user-friendly platform for students, faculty, and admins. The proposed system will offer role-based access, allowing faculty to post job listings, internships, and training opportunities while managing applications and deadlines. Admins will have full control over the platform, including managing staff and overseeing the overall placement process.

Students will have a seamless experience, with easy access to job listings, internships, and application tracking. The platform will provide real-time updates, ensuring that students are always informed about new opportunities

2.5 Advantages of Proposed System

1. **Centralized Platform**: All placement-related activities are managed in one place, improving efficiency.
2. **Real-Time Updates**: Students receive immediate notifications about new job listings, internships, and deadlines.
3. **Role-Based Access**: Different user roles (students, faculty, admins) have specific functionalities, enhancing security and usability.
4. **Streamlined Communication**: Facilitates clear communication between students, faculty, and admin.
5. **Improved Efficiency**: Reduces manual intervention, speeding up the job posting and application process.
6. **Enhanced User Experience**: Easy-to-navigate interface for students, faculty, and admins, improving overall usability.

3. METHODOLOGY

3. METHODOLOGY AND PROBLEM IDENTIFICATION

3.1 User Requirement

To design a website that supports the needs of the company and the users, you must know who your audience is. It is important to determine your users at this early stage of the project. Some of the things you want to discover are:

1. What do they want to do on the site?
2. What will make them return to the site?
3. What is their level of experience with the web?

3.2 Software Requirement

1.	Operating System	: Windows XP.
2.	Front End	: HTML.
3.	Script	: JavaScript.
4.	Storage	: Google Drive.
5.	Browser	: Internet Explorer (Above 12.0 Versions).

3.3 Hardware Requirement

1.	Processor	-	Dual Core.
2.	Speed	-	1.7 GHz.
3.	RAM	-	1 GB (min).
4.	Hard Disk	-	20 GB.
5.	Key Board	-	Standard Windows Keyboard.
6.	Mouse	-	Standard Mouse.
7.	Monitor	-	LCD.

3.4 Content Diagram of project

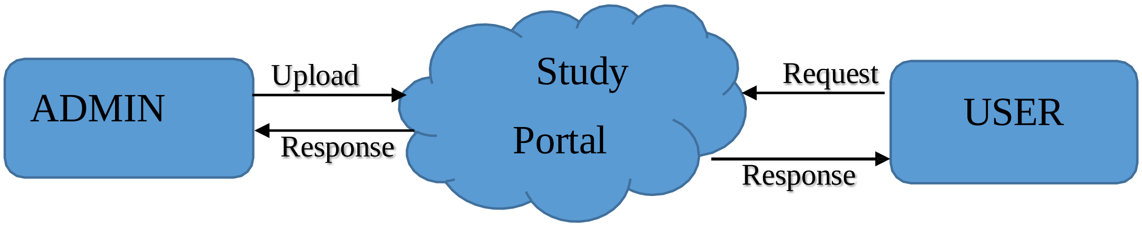


Fig 1: Architecture of System

4. DESIGN AND IMPLEMENTATION

4. DESIGN AND IMPLEMENTATION

4.1 Data Flow Diagram (DFD)

The Data Flow Diagram (DFD) for the Training and Placement website shows how Admins manage the platform, Faculty post jobs, and Students register, explore listings, and apply for opportunities. Key data stores like User Database, Job Listings, and Applications ensure smooth data flow and system functionality.

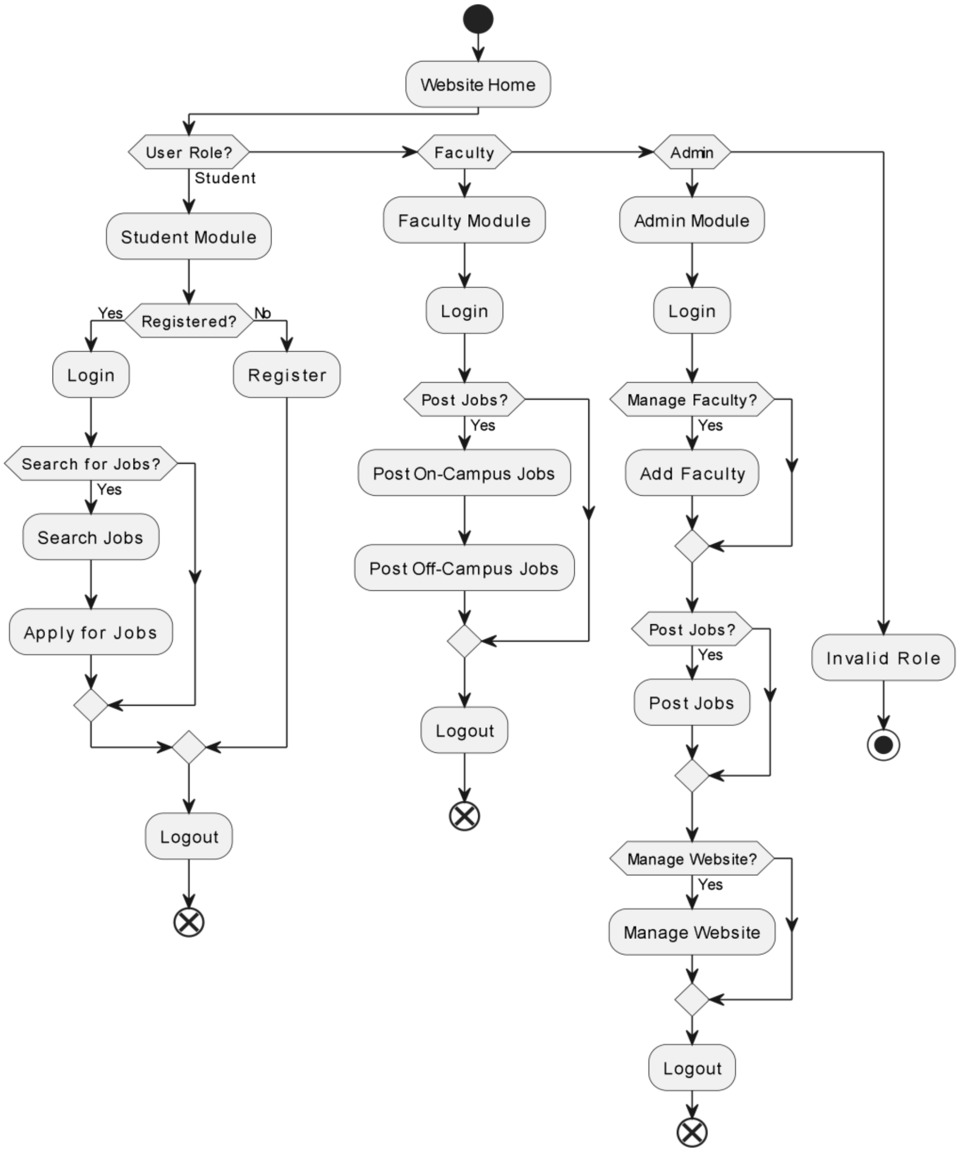


Fig : 2 activity diagram of system

4.1.2 Use Case Diagram:

The Use Case Diagram for this project represents the interactions between the primary actors—Admin, Faculty, and Students—and the system. Admins manage the platform by adding faculty, overseeing job listings, and monitoring student activities. Faculty members interact with the system to post job opportunities, internships, and hackathon details. Students use the system to register, explore job listings, apply for jobs, and access career-related resources. The diagram highlights key use cases such as user registration, job posting, and application tracking, showcasing how the system streamlines the placement process for all stakeholders.

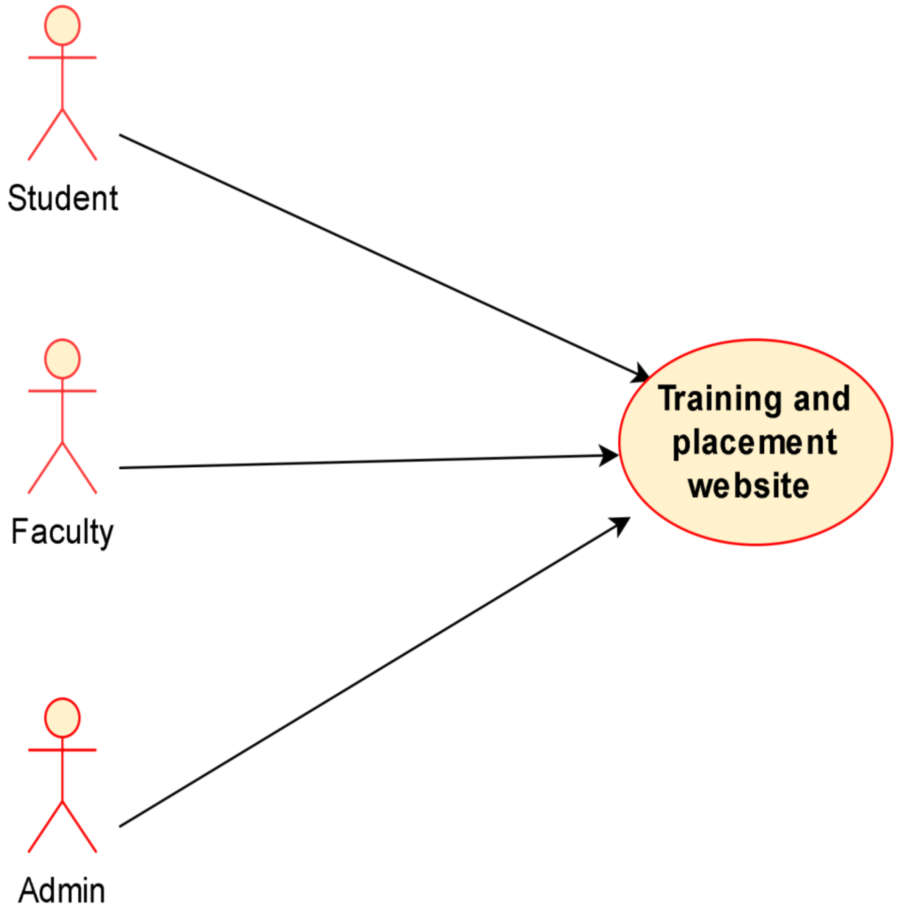


Fig 3 use case diagram of system

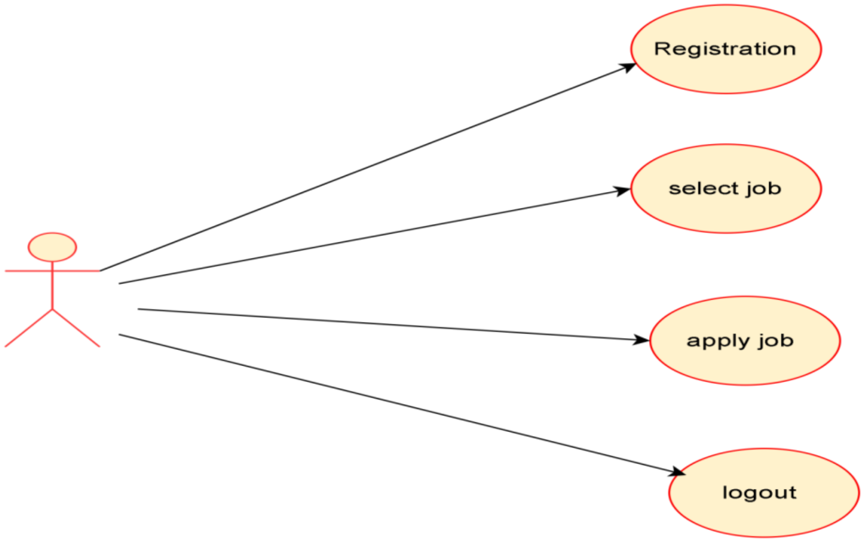


Fig 4 use case diagram of student

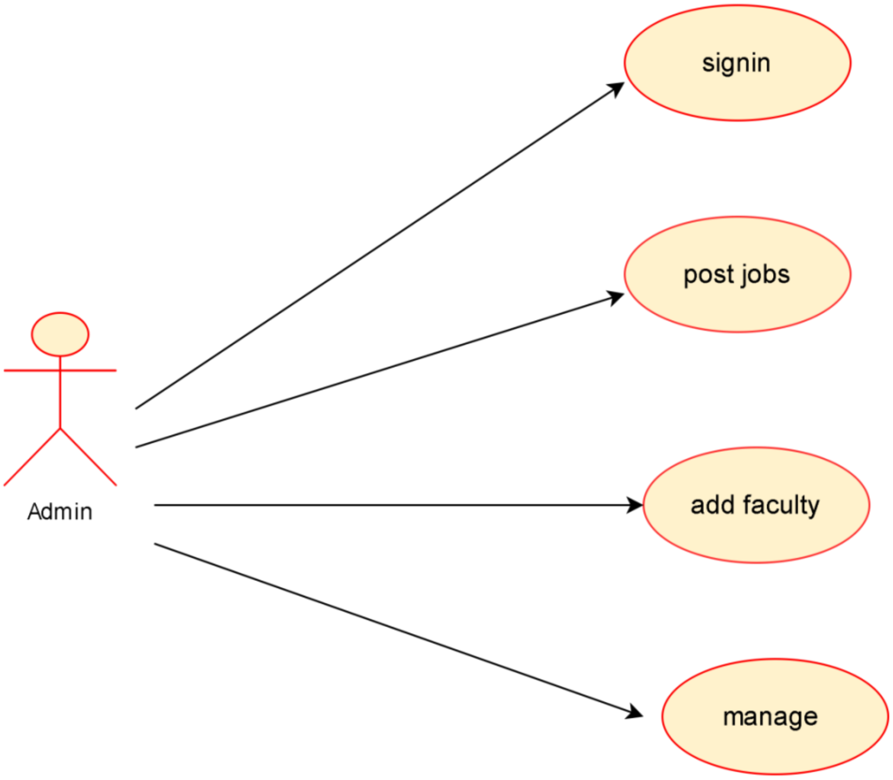


Fig 5 use case diagram of admin

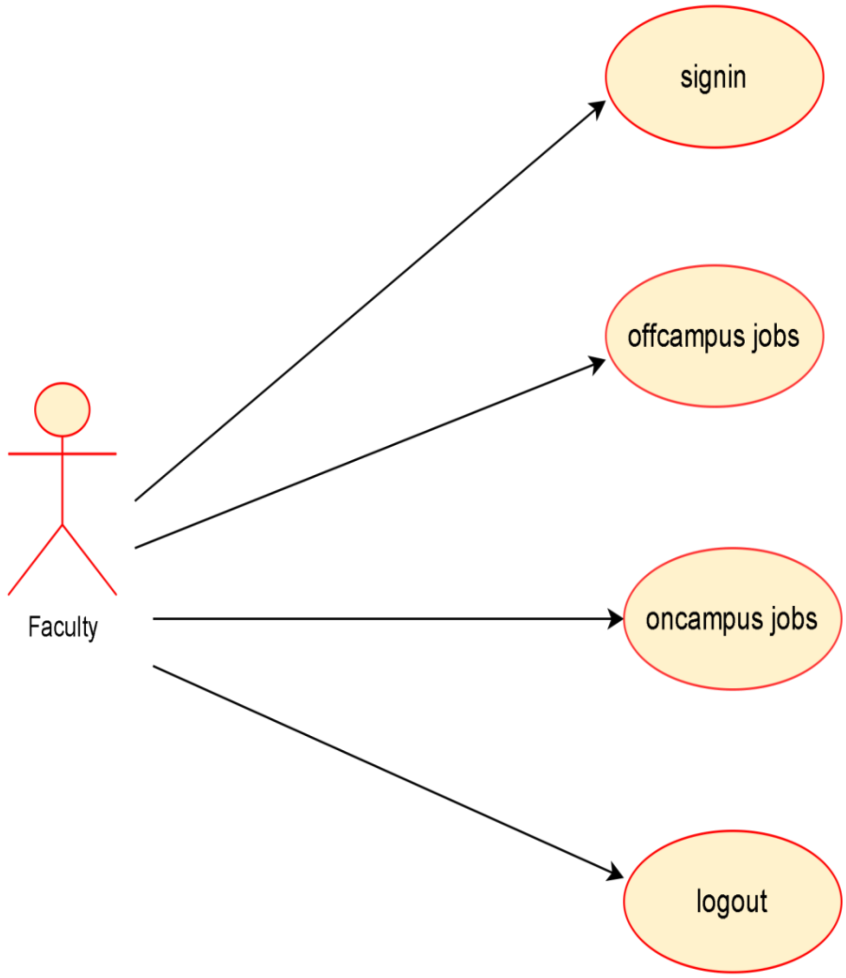


Fig 6 use case diagram of faculty

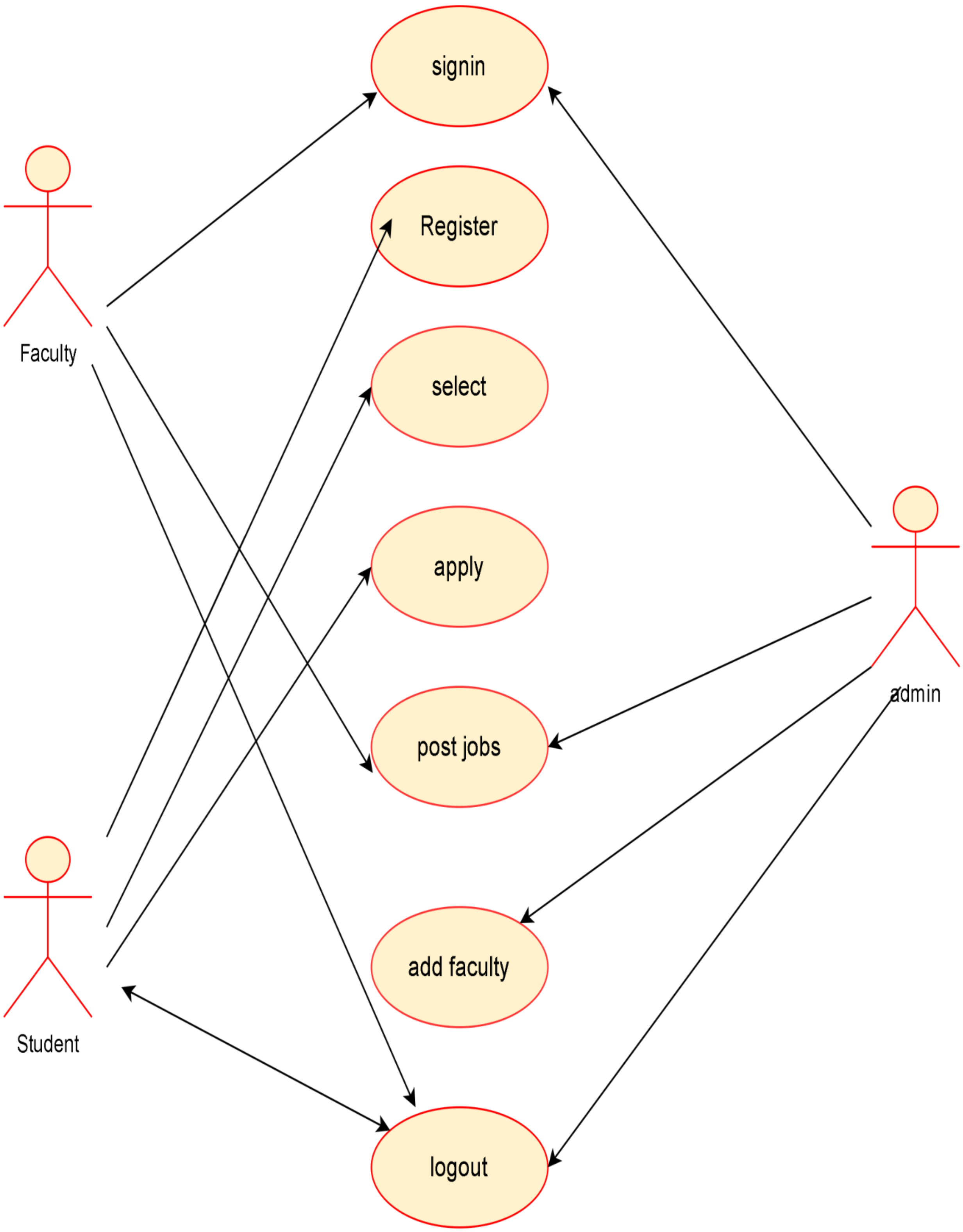


Fig 7 use case diagram of tnp

4.1.3 Class Diagram:

The Class Diagram for the Training and Placement website outlines key classes like Admin, Faculty, and Student, each with specific attributes and methods. It also includes classes for Job Listings, Applications, and User Database, showing how these interact to manage user accounts, job postings, and applications efficiently.

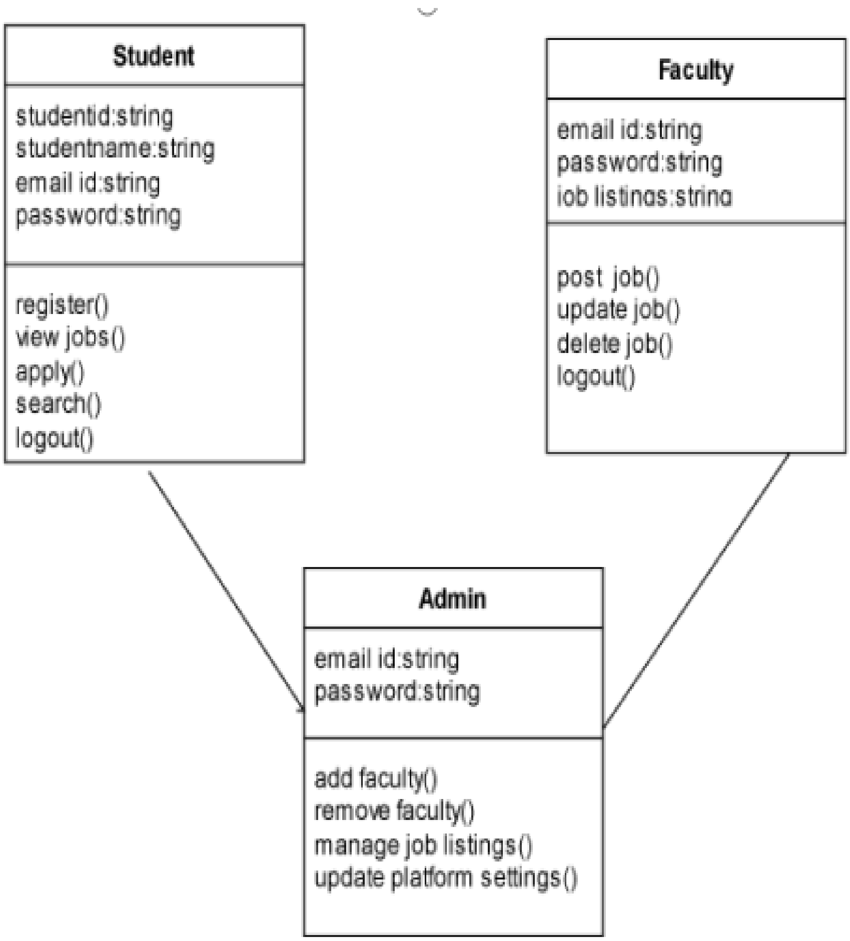


Fig 8 class diagram

4.1.4 Sequence Diagram:

The Sequence Diagram for this project shows interactions between Admin, Faculty, Student, and the system. Faculty post jobs, students apply for jobs, and Admin manages user roles and job listings. The system validates inputs and updates relevant databases for job listings and applications, ensuring smooth task execution.

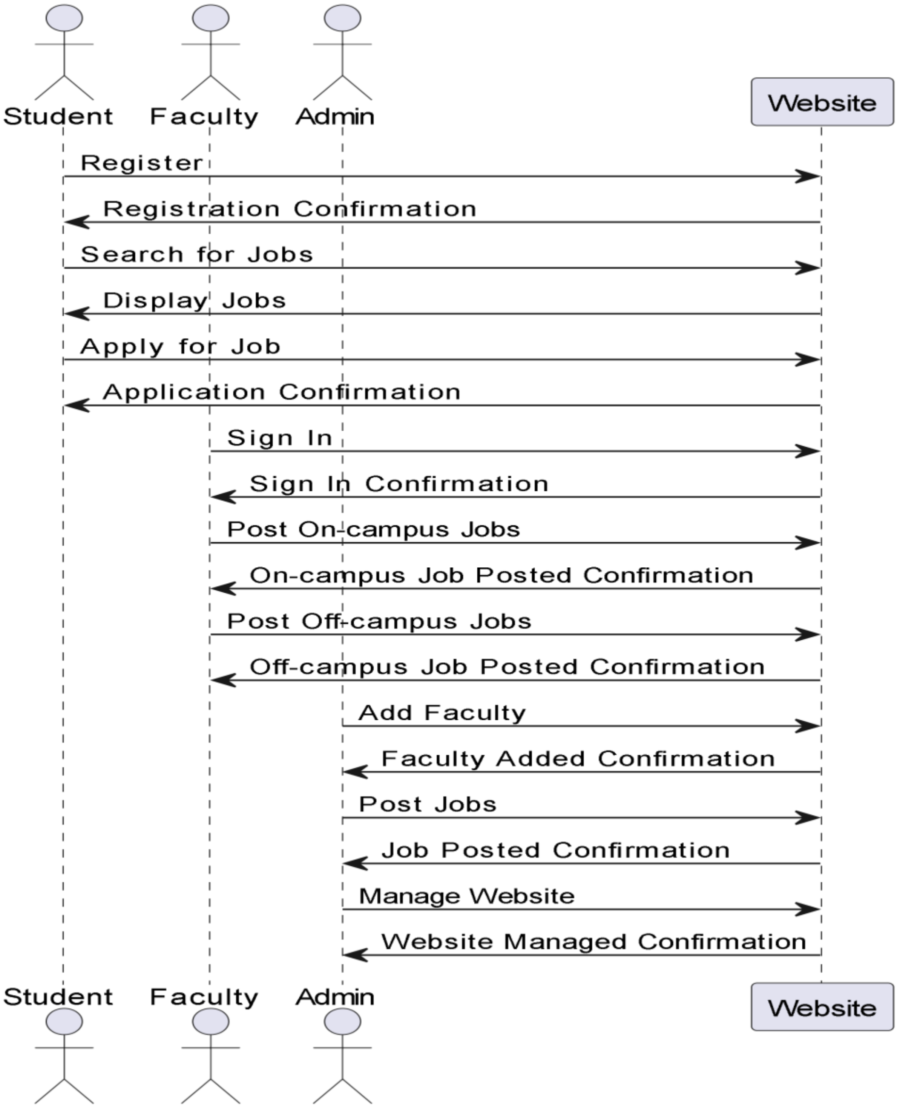


Fig 9 Sequence diagram

4.1.5 **Activity Diagram:**.

The activity diagram outlines the process flow for three roles: Student, Faculty, and Admin. The Student registers, searches, and applies for jobs. The Faculty signs in and posts on-campus and off-campus jobs. The Admin adds faculty, posts jobs, and manages the website. The diagram shows the sequential tasks for each role, ensuring efficient job management and application.

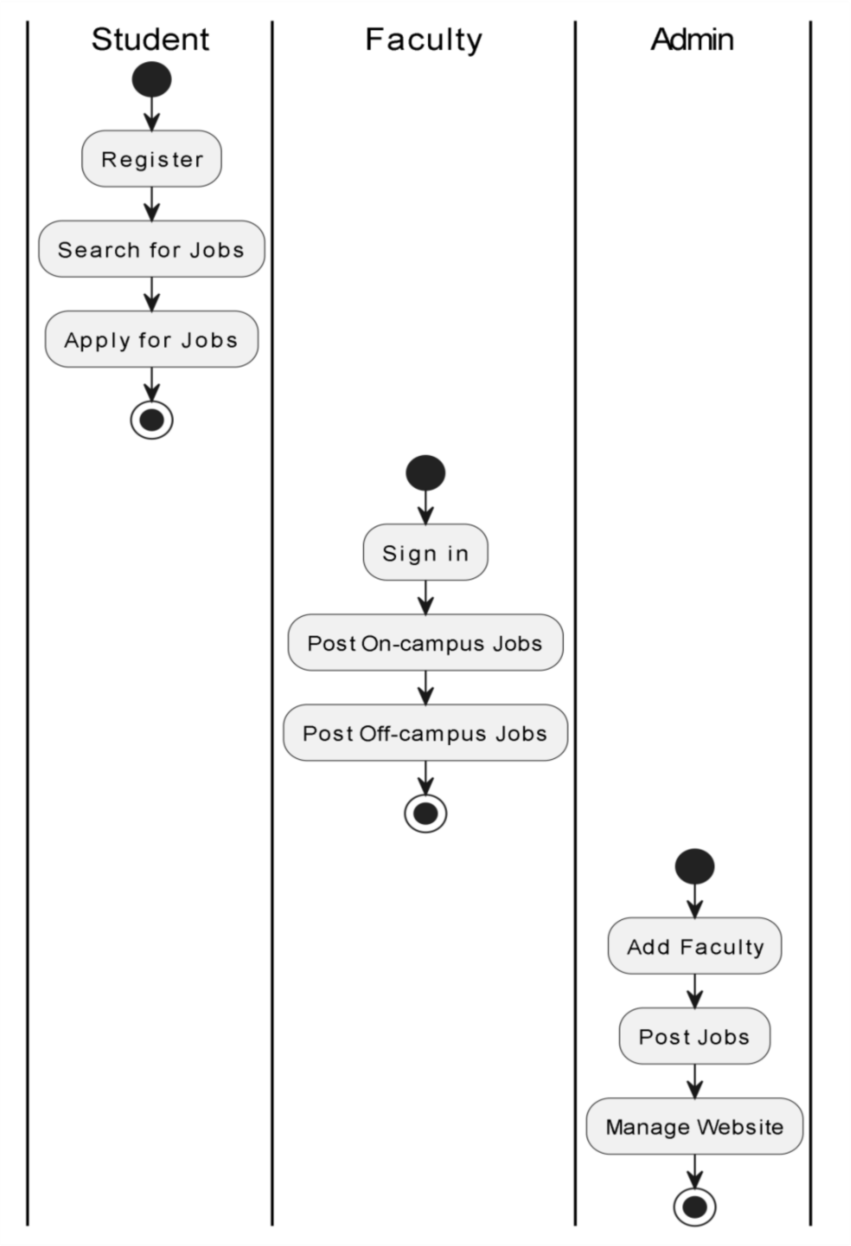


Fig 10 Activity Diagram

4.2 Explanation of Key Functions

4.2.1 HTML

Hyper Text Markup Language (HTML) is used for creating and visually representing a webpage. HTML adds "markup" to Standard English text. "Hyper Text" refers to links that connect Web pages to one another, making the World Wide Web what it is today. By creating and uploading Web pages to the Internet, you become an active participant in the World Wide Web. HTML supports visual images and other media as well. HTML is the language that describes the structure and the semantic content of a web document. Content within a web page is tagged with HTML elements such as <img>, <title>, <p>, <div>, <picture>, and so forth. These elements form the building blocks of a website.

4.2.2 CSS

Cascading Style Sheets, most of the time abbreviated as **CSS**, is a stylesheet language used to describe the presentation of a document written in HTML or XML (including various XML languages like SVG or XHTML). CSS describes how the structured element must be rendered on screen, on paper, in speech, or on other media. **CSS** is one of the core languages of the open web and has a standardized W3C specification. Developed in levels, CSS1 is now obsolete, CSS2.1 a recommendation and CSS3, now split into smaller modules, and is progressing on the standard track.

4.2.3 JS

JavaScript (JS) is a versatile, high-level programming language primarily used to create dynamic and interactive features on websites. It enables developers to build engaging web applications by manipulating HTML, CSS, and the Document Object Model (DOM). As a client-side scripting language, JavaScript runs directly in the browser, but with the advent of environments like Node.js, it is also widely used for server-side development. It supports event-driven, functional, and object-oriented programming styles, making it a cornerstone of modern web development.

4.3 Method of Implementation

4.3.1 Forms

The following pages provide various techniques for improving the accessibility of web forms:

Basic form hints: Adding hints and descriptions for invalid or required fields

Alerts: Using alerts to provide client-side validation error messages

Multi-part labels: Enabling complex form labels with a control inside each label.

□ **Student Module**:

- **Features**: Profile management, resume upload, and view opportunities.
- **Methods**: Register(), Update Resume(), Apply For Job ().

□ **Company Module**:

- **Features**: Post job openings, shortlist candidates, schedule interviews.
- **Methods**: Post Jo(), View Applications(), Schedule Interview().

□ **TnP Admin Module**:

- **Features**: Verify student profiles, approve job postings, generate reports.
- **Methods**: Approve Profile(), Approve Job Post(), Generate Report().

□ **Training Module**:

- **Features**: Organize training sessions, manage attendance, and provide feedback.
- **Methods**: Schedule Training(), Mark Attendance(), Give Feedback().

4.3.2 BasicForms

When implementing forms using traditional HTML form-related elements, it is important to provide labels for controls and to explicitly associate a label with its control. When a screen reader user navigates a page, the screen reader will describe form controls, but without a direct association between the control and its label, the screen reader has no way of knowing which label is the correct one.

The example below shows a simple form with labels. Note that each <input> element has an id, and each <label> element has a for attribute indicating the id of the associated

<input>. Eg:

```
<form>
        <ul>
                   <li>
                   <input id="subject-1" type="checkbox" value="Bharat"/>
                  <label for="wine-1">BIET</label>
                  </li>
</form>
```

4.3.3 Alerts:

You have a form — a contact form, for example — that you want to put some accessible error checking into. Common problems are e-mail addresses that are not valid, or a name that does not contain at least a first and a surname.

Checking the validity and notifying the user consists of several steps:

1. Checking if the e-mail address or entered name are valid. To keep it simple, we'll check whether the e-mail address contains the "@" symbol, and if the name entry contains at least 1 space characters" ".
2. Setting the field's aria-invalid attribute and giving it a value of "true".
3. Notifying the user via an alert that the value entered was incorrect. Instead of using an intrusive dialog box created by the JavaScript 'alert' function, we'll use a simple WAI- ARIA widget to do it. This notifies the user, but lets them continue interacting with the form without any interruptions.

All of this happens when the input loses focus, meaning in the “**onblur**” handler.

The JavaScript code I wrote looks like this, inserted above the closing “head” tag: Ex:

```
<script
 type="application/javascript">
 function removeOldAlert ()
 {
   var                    oldAlert                   =
   document.getElementById("alert");            if
   (oldAlert)
   document.body.removeChild(oldAlert);
 }
</script>
```

SAMPLE SOURCE CODE

<!doctype html>

```
<html lang="en">
  <head>
    <meta charset="UTF-8" />

    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>BIET-TnP</title>
  </head>
  <body>
    <div id="root"></div>
    <script type="module" src="/src/main.jsx"></script>
  </body>
</html>
```

index.css

```
@tailwind base;
@tailwind components;
@tailwind utilities;

:root {
  font-family: Inter, system-ui, Avenir, Helvetica, Arial, sans-serif;
  line-height: 1.5;
  font-weight: 400;

  color-scheme: light dark;
  background-color: white;
  color: #050505;

  font-synthesis: none;
  text-rendering: optimizeLegibility;
  -webkit-font-smoothing: antialiased;
  -moz-osx-font-smoothing: grayscale;
}
```

main.jsx

```
import React from 'react'
import ReactDOM from 'react-dom/client'
import App from './App.jsx'
import './index.css'
import { Provider } from 'react-redux'
import store from '../store.js'

ReactDOM.createRoot(document.getElementById('root')).render(

    <Provider store={store}>
    <App />
    </Provider>

)
```

facultyModel.js

```
const mongoose=require('mongoose');

const facultySchema= new mongoose.Schema({
    name: {
        type:String,
        required: true
    },
    email: {
        type:String,
        required: true,
        unique: true
    },
    password: {
        type:String,
        required: true
    },
    role: {
        type:String,
        required: true,

    }
})

const Faculty= mongoose.model('Faculty',facultySchema);
```

```
module.exports=Faculty;
```

offCampusModel.js

```
const mongoose=require('mongoose');

const offCampusSchema= new mongoose.Schema({
    position: {
        type:String,
        required: true
    },
    companyname: {
        type:String,
        required: true,

    },
    location: {
        type:String,
        default:"Not mentioned"
    },
    salary: {
        type:String,
        default:"Not mentioned"
    },
    batch:{
        type:Array,
        default:[],
    },
    jobtype: {
        type:String,
        default:"Not mentioned"
    },
    branch: {
        type:String,
        default:'All Branches'
    },
    src: {
        type:String,
        required: true
    },
    postedOn:{
        type:Date,
        default:Date.now(),
    },
    postedBy:{
        type:mongoose.Schema.Types.ObjectId,
        ref:'Faculty',
       required:true
    },
    autoDelete:{
        type:Date,
        default:null,
    },

})
offCampusSchema.index({autoDelete:1},{expireAfterSeconds:0});

const offcampus= mongoose.model('offcampus',offCampusSchema);

module.exports=offcampus;
```

onCampusModel.js

```
const mongoose=require('mongoose');

const onCampusSchema= new mongoose.Schema({
    position: {
        type:String,
        required: true
    },
    companyname: {
        type:String,
        required: true,

    },
    description: {
        type:String,
        required: true,

    },
    location: {
        type:String,
        default:"Not mentioned"
    },
    salary: {
        type:String,
        default:"Not mentioned"
    },
    batch:{
        type:Array,
        default:[],
    },
    jobtype: {
        type:String,
        default:"Not mentioned"
    },
    branch: {
        type:String,
        default:'All Branches'
    },
    src: {
        type:String,
        required: true
    },

    postedOn:{
        type:Date,
        default:Date.now(),
    },
    postedBy:{
        type:mongoose.Schema.Types.ObjectId,
        ref:'Faculty',
       required:true
    },
    autoDelete:{
        type:Date,
        default:null,
    },

})
onCampusSchema.index({autoDelete:1},{expireAfterSeconds:0});

const oncampus= mongoose.model('oncampus',onCampusSchema);

module.exports=oncampus;
```

userModel.js

```
const mongoose=require('mongoose');

const userSchema= new mongoose.Schema({
    name: {
        type:String,
        required: true
    },
    email: {
        type:String,
        required: true,
        unique: true
    },
    password: {
        type:String,
        required: true
    },
    role: {
        type:String,
        required: true,
    }
})

const User= mongoose.model('User',userSchema);

module.exports=User;
```

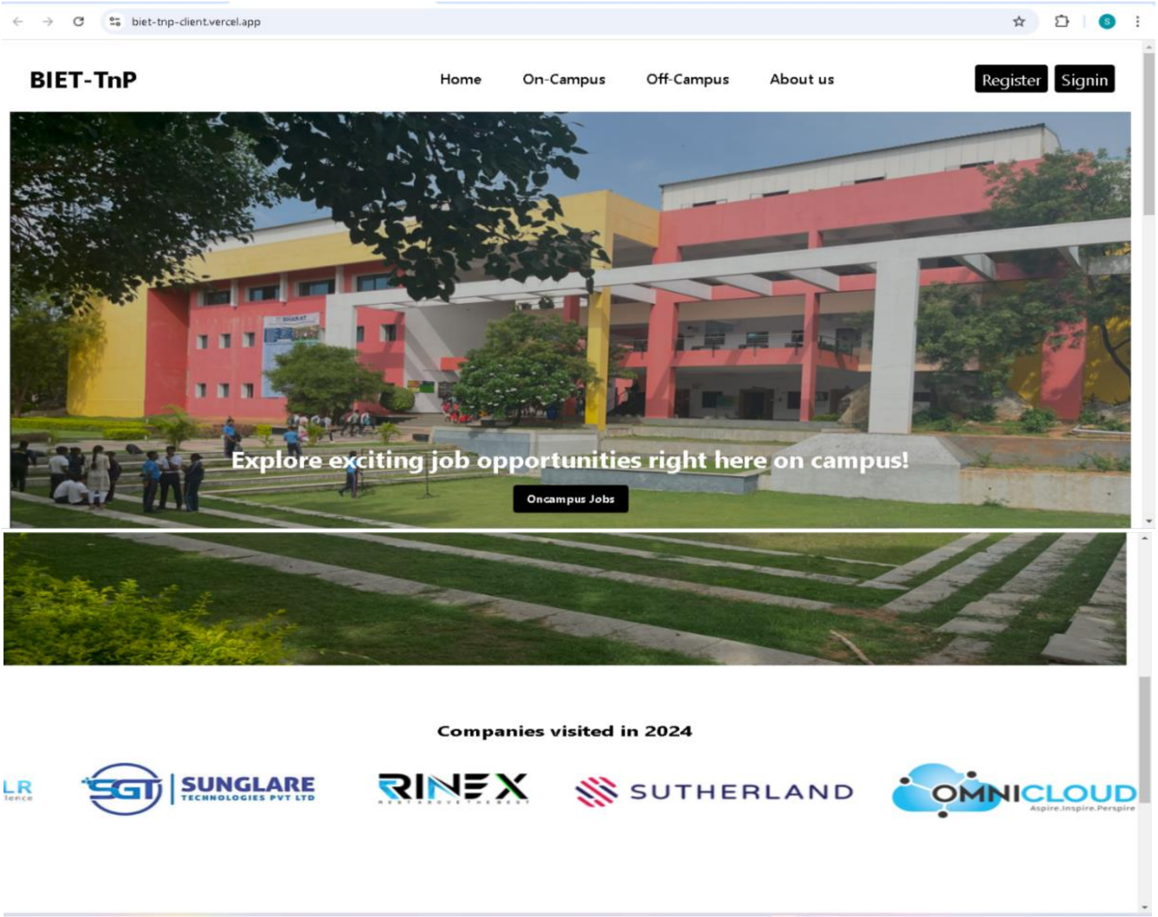
adminController.js

```
const Faculty = require('../models/facultyModel');
const bcrypt = require('bcrypt');

async function add(req, res) {
    const { name, email, password, role } = req.body;
    try {
        const user = await Faculty.findOne({ email });
        if (user) {
            return res.status(400).json({ message: "User already exists", status: true });
        }
        const hashedPassword = await bcrypt.hash(password, 10);
        const newUser = new Faculty({ name, email, password: hashedPassword, role });
        await newUser.save();

        res.status(201).json({ message: 'User added successfully', status: true });
    } catch (error) {
        console.error('Error registering user:', error.message);
        res.status(500).json({ message: 'Server error', status: false });
    }
}

module.exports = {add} ;
```

4.4 Output Screens

Home page:

This Home Page Includes the Index of the webpage which gives brief information of all the topics which are included in the website.

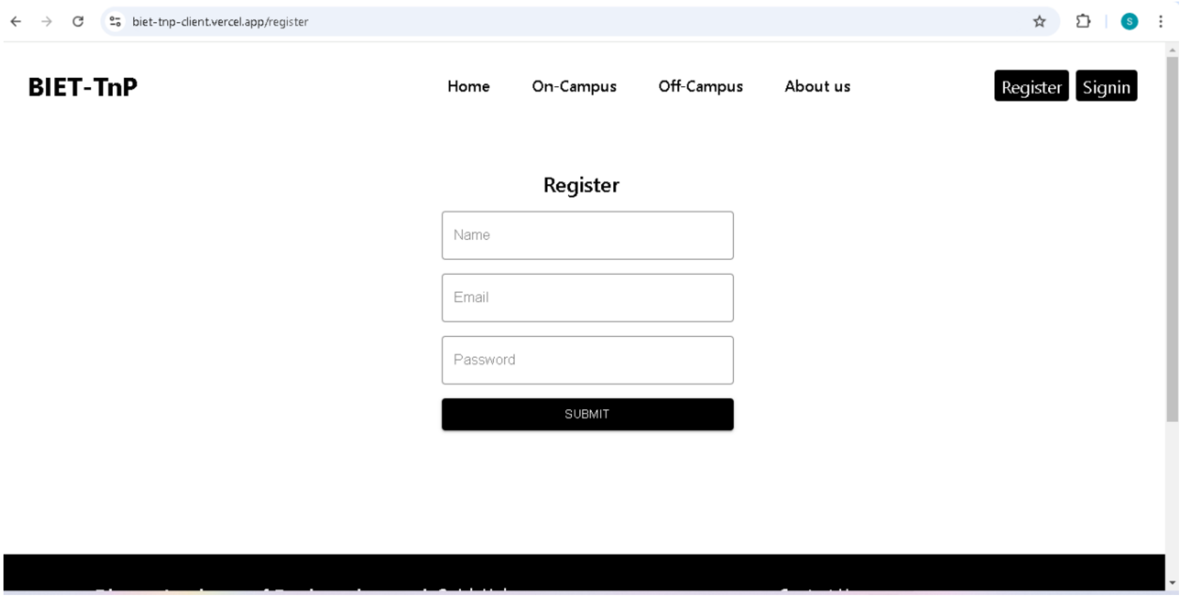


Explore exciting job opportunities right here on campus!

Companies visited in 2024

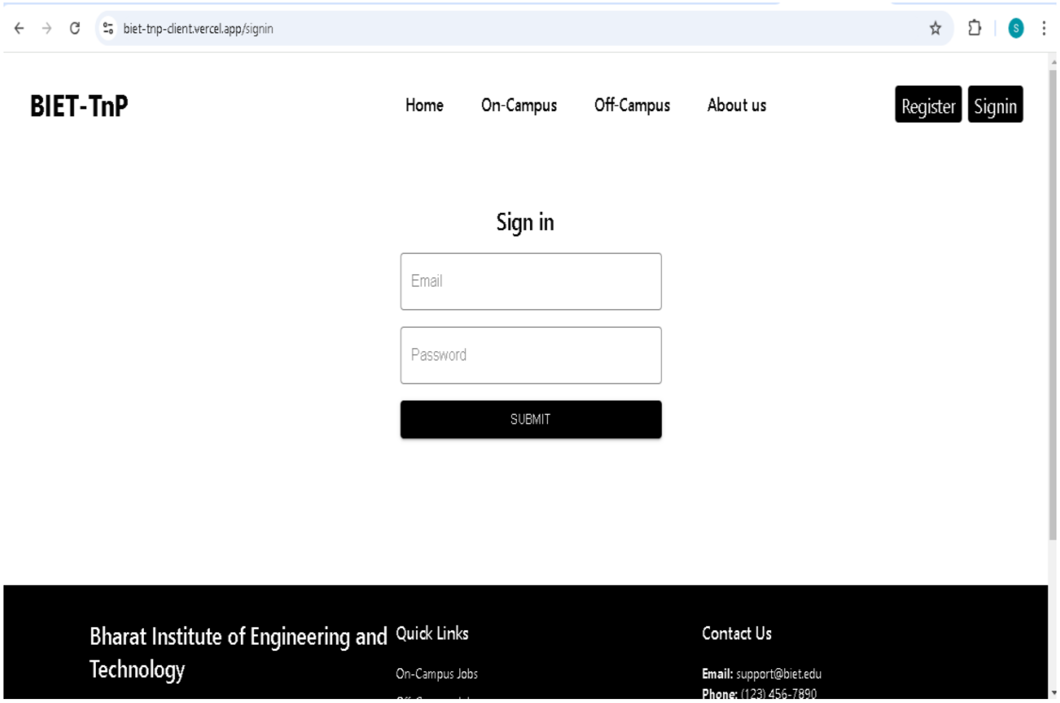
Student Register Page:

The registration page of your website provides a user-friendly interface for new users to create an account. It typically includes fields for essential information such as name, email address, and password, ensuring a straightforward and secure registration process.



Faculty Signin Page:

The sign-in page of your website, BIET-TnP, provides a user-friendly interface for users to access their accounts securely.It features input fields for username and password, along with a 'Sign In' button to submit credentials.The page ensures secure authentication, allowing users to access personalized features and services upon successful login.



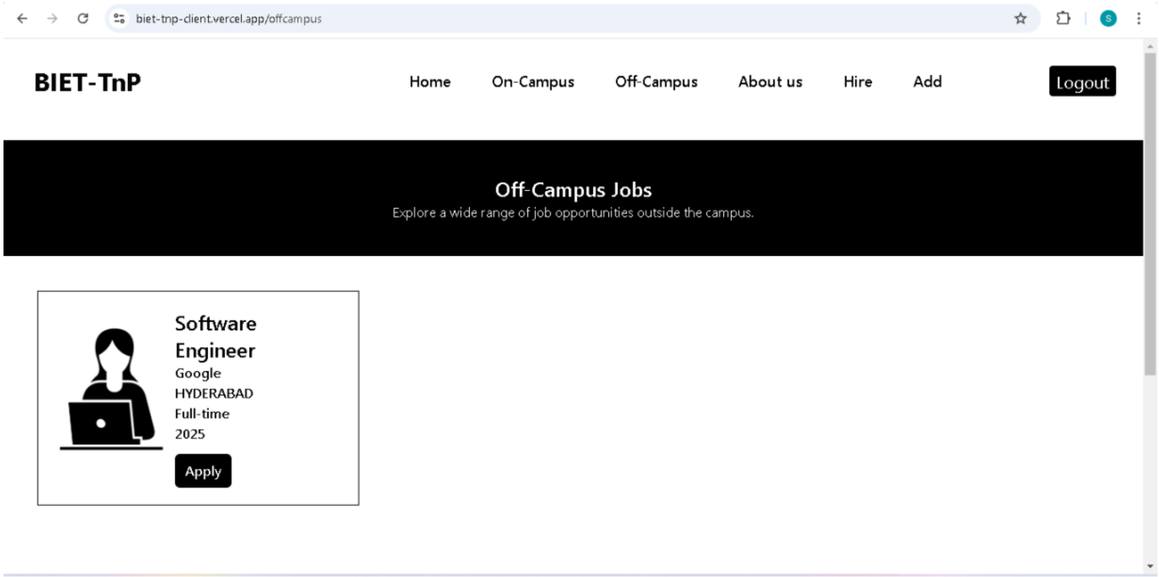
Bharat Institute of Engineering and Technology

Email: support@biet.edu
Phone: (123) 456-7890

Jobs Page:

The "Job Off Campus" page on your website provides students with information about employment opportunities available outside the campus premises.It includes details about various companies, job roles, application procedures, and eligibility criteria, enabling students to explore and apply for positions that align with their career aspirations.□

The "Job On Campus" page offers insights into recruitment drives conducted within the campus.It features schedules of upcoming placement events, participating companies, job descriptions, and necessary preparations, assisting students in effectively engaging with potential employers during campus recruitment activities.



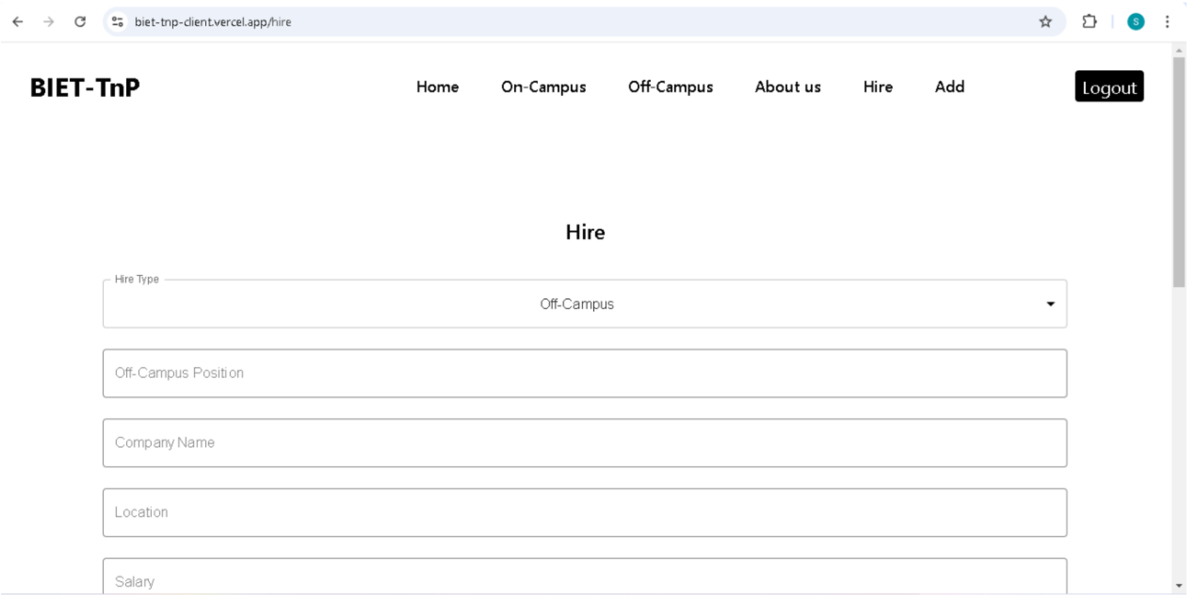
Off-Campus Jobs

Explore a wide range of job opportunities outside the campus.

Software Engineer
Google
HYDERABAD
Full-time
2025

Hire Page:

The "Hire" page on your website serves as a platform for potential employers to connect with talented students from BIET.It provides detailed information about the institution's training and placement services, highlighting the skills and competencies of the students.Employers can access resources to facilitate recruitment processes, ensuring a seamless experience in hiring qualified candidates.□



biet-tnp-client.vercel.app/hire

Hire

Hire Type: Off-Campus

5. RESULT ANALYSIS

4.5 RESULT ANALYSIS

Introduction

Training and Placement website refers to the process of evaluating the effectiveness of the platform in achieving its primary objectives. This involves analyzing data, user interactions, job placements, and feedback to assess how well the website is functioning and identify areas for improvement. The goal is to provide insights that can help both students and employers, as well as administrators, in making better decisions for future placements.

4.5.1 Design of Test cases and scenarios

4.5.1.1 Unit testing

Unit testing for the **Training and Placement Website** focuses on verifying the functionality of individual components to ensure they work as expected. Key features, such as the job listing posting by faculty, student registration, and the admin management of the platform, are tested for accuracy and performance. Test cases are designed to check if all functionalities—like adding job opportunities, creating user accounts, and role-based access—perform without errors. The tests also assess the response time and stability of the platform under different scenarios to ensure reliability. By conducting thorough unit testing, we ensure that each module functions correctly before integrating them into the larger system .

4.5.1.2 Integration testing

Integration testing for the **Training and Placement Website** focuses on validating the interaction between different modules and components to ensure seamless functionality across the entire platform. This testing phase checks the flow of data between the admin, faculty, and student roles, ensuring that job listings posted by faculty are correctly displayed to students and that students can apply for jobs without any issues. It also verifies that the admin can manage user accounts and job listings effectively. Integration testing ensures that the platform's database, user interfaces, and backend systems work together harmoniously, providing a smooth experience for all users. By addressing any issues in the interaction between components, we ensure that the entire system functions as expected in real-world usage scenarios.

4.5.1.3 Functional test

These test cases ensure that the basic functionalities of the website work as expected for different users (students, employers, and admins).

1.1. Registration and Login

Scenario 1: Student Registration

Test Case: Verify that a student can register by providing all the required details.

Input: Name, email, password, course details, contact information.

Expected Output: Account is created successfully, and a confirmation email is sent.

Scenario 2: Login (Student/Employer)

Test Case: Verify that registered users (students and employers) can log in using their credentials.

Input: Email and password.

Expected Output: User is successfully logged in and redirected to the dashboard.

Scenario 3: Invalid Login Attempt

Test Case: Verify that users receive an error message if login credentials are incorrect.

Input: Invalid email or password.

Expected Output: An error message is displayed, and the user is prompted to re-enter credentials.

4.5.1.4 System Test

System testing for this project involves evaluating the entire platform as a whole to ensure that all components work together seamlessly. It verifies the integration of front-end and back-end functionalities, ensuring that features such as user registration, job listing management, and role-based access function correctly across all user roles. The testing also includes performance, security, and usability assessments to confirm that the platform meets both business and technical requirements, ensuring reliability and a smooth user experience.

White Box Testing

White box testing for this project involves testing the internal code and logic of key features like student registration, job listings, and admin management. It ensures that all code paths, functions, and branches work as expected. Test cases focus on verifying control flow, data flow, and handling of edge cases such as invalid inputs. This testing ensures the system's internal components are secure, efficient, and function correctly.

Black Box Testing

Black box testing for the **Training and Placement Website** focuses on evaluating the functionality of the system without considering the internal code or structure. Test cases are designed based on the platform's requirements, user interfaces, and expected behavior, including features such as student registration, job listing posting, and role-based access. The goal is to verify that the system responds correctly to various inputs, including valid and invalid data, and that the expected outputs are generated. This testing ensures that the website meets its functional requirements, performs as intended, and provides a seamless user experience, without delving into the underlying code logic.

1. Unit Testing

Unit testing is conducted during the development phase to ensure that each individual module or component of the **Training and Placement Website** works correctly. This testing ensures that the core functions, such as user registration, job listing posting, and role-based access, are working as expected. Developers test small units of code to validate their behavior before integrating them into the larger system.

2. Test Strategy and Approach

Field testing for the **Training and Placement Website** will be performed manually, focusing on real-world scenarios. Functional tests will be written in detail to verify that the core features work as intended. These tests will cover tasks like job listing posting, student registration, and user role management, ensuring the platform's usability and accuracy.

3. Test Objectives

1. All user inputs (job listings, student registrations) must be properly processed.
2. Links and navigation must direct users to the correct pages.
3. The system must respond to user actions without delays or errors.

4. Features to be Tested

1. Ensure that all entries (job listings, student data) are in the correct format.
2. Prevent the creation of duplicate job listings or student registrations.
3. Verify that all links, such as "Apply Now" and "View Details," direct users to the correct pages.

5. Integration Testing

Software integration testing for the **Training and Placement Website** will focus on checking the interaction between the platform's modules, such as job posting, user registration, and admin management. The goal is to ensure that each module interacts smoothly without errors, especially when data flows between different components, like job listings being posted by faculty and accessed by students. This testing will identify and resolve interface defects to ensure seamless integration across the platform.

5.2 Test Results

All the test cases mentioned above passed successfully. No defects encountered.

5.2.1 Acceptance Testing

Acceptance testing for the **Training and Placement Website** ensures that the system meets all specified requirements and functions as expected in real-world scenarios. It involves verifying key features such as job posting, student registration, and user role management to ensure they align with business goals. End users, including students, faculty, and admins, will perform tests to validate that the platform is intuitive, reliable, and ready for deployment. The goal is to confirm that the system is fully operational and satisfies all user needs before going live.

5.2.2 Results

All the test cases mentioned above passed successfully. No defects encounter

6. CONCLUSION

6.CONCLUSION

The Training and Placement Website has been successfully designed, developed, and tested to meet the
needs of students, faculty, and administrators. The platform effectively bridges the gap between educational institutions and the job market by providing a seamless, user-friendly interface that facilitates
job postings, student applications, training programs, and notifications
.
Key features such as user registration, job posting and application, Off campus and On campus jobs ensuring smooth interaction for all stakeholders. The security mechanisms, including encrypted passwords, role-based access control ensure that sensitive information is handled safely and securely. The
platform has been tested under real-world conditions, with both functional and non-functional testing conducted to verify the system's performance, reliability, and scalability. . Usability testing confirmed that the platform is intuitive and easy to navigate, enhancing the overall user experience for students, employers, and admins alike. Security features, such as login authentication and session timeout, were thoroughly tested and verified to protect against unauthorized access.

Overall, the Training and Placement Website is a comprehensive, robust, and secure platform that streamlines the recruitment process. It meets the needs of all users while maintaining a high standard of
usability, performance, and security. With successful testing results and key functionalities in place, the
website is now ready to be deployed for use, offering valuable services to students, employers, and institutions aiming to enhance the employment prospects of students.

7.REFERENCES

7.REFERENCES

[1]. Aneena Felix, Ajeena Sunny, Angelin Saji, "Placement Management System for Campus Recruitment," International Journal of Innovative Science and Research Technology, Volume 5, Issue 5, May 2020.

[2]. Akshata Bhalgat, Ina Datta, Abhishek Kolkar, Aditya Mate, "Training & Placement Management System," International Engineering Research Journal (IERJ), Volume 2, Issue 10, Dec 2017.

[3]. Rahul Kumar Modi, Dr. S.B.Vanjale, Akshit Jain, Supreet Raj, "Smart Training & Placement System," International Journal of Computer Science and Technology (IJCST), Vol. 8, Issue 2, April - June 2017.

[4]. Praneeth Ambiti, Navaneeth Kumar B, K Hema, Vamsi Kandula, Kishore Buddha, "Student Analysis System for Training and Placement," International Journal of Recent Technology and Engineering (IJRTE), Volume-8, Issue6, March 2020.

[5]. Mrs. Srividhya V R, Santhosh Kumar H, "Online Training and Placement Management System," International Journal of Engineering Research & Technology (IJERT), Volume 4, Issue 22, ICACT - 2016 Conference Proceedings.

[6]. Spoorthi M S, Kavana V, Koushik S N, Veena M, "A Review on Placement Management System," International Journal of Creative Research Thoughts (IJCRT), Volume 9, Issue 7, July 2021.

[7]. C. K. Patil, K. G. Patel, "Study of Implementation of Online Placement System," International Journal of Advanced Technology in Engineering and Science (IJATES), Vol. No.4, Issue No. 03, Mar 2016.

[8]. Sandeep S. Chorge, Ganesh A. Bhabad, Balaji D. Chate3, Rahul T. Take, Nitin R. Shinde, "Online Training and Placement Portal with Android Application," International Journal of Advance Research and Innovative Ideas in Education (IJARIIE), Vol-3, Issue-3, 2017.

[9]. FU Yue, "A Study of Student Information Management Software," International Conference of Online Analysis and Computing Science (ICOACS), IEEE, May 2016.

[10]. Dipin Budhrani, Vivek Mulchandani, Yugchhaya Galphat, "Student Information Management System," International Journal of Engineering Development and Research (IJEDR), Volume 6, Issue 1, 2018.