A Project Report

on

Future Mesh Enhancing Employability Through Mentorship and Smart Job Matching

Submitted by

ABHISEK PANDA (22UG010159)

DEBABRATA MISHRA (22UG010273)

DIPTESH NARENDRA (22UG010357)

Under the Supervision of

Dr. Gitanjali Mishra

Assistant Professor
Department of CSE, GIET University, Gunupur (Odisha)

A Project Report submitted in partial fulfilment of the requirements for the degree of

B. TECH IN COMPUTER SCIENCE AND ENGINEERING (CSE)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SCHOOL OF ENGINEERING AND TECHNOLOGY,

GANDHI INSTITUTE OF ENGINEERING AND TECHNOLOGY UNIVERSITY

GUNUPUR 765022

NOV 2025

DEDICATION

We dedicate this project to our beloved parents, whose constant love, encouragement, and support have been our greatest strength throughout this journey.

We also dedicate this work to our respected teachers and mentors, who have guided us with their knowledge and wisdom, and to all our friends and classmates for their cooperation, motivation, and memorable companionship during our B.Tech journey.

Above all, we thank the Almighty for giving us the courage, patience, and perseverance to complete this work successfully.

GIET University, Gunupur

ABHISEK PANDA – 22UG010159
DEBABRATA MISHRA - 22UG010273
DIPTESH NARENDRA - 22UG010357

GANDHI INSTITUTE OF ENGINEERING AND TECHNOLOGY UNIVERSITY, GUNUPUR

DIST-RAYAGADA, (ODISHA) PIN-765022

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Certificate of Examination

ABHISEK PANDA	**22CSE072 (22UG010159)**
DEBABRATA MISHRA	**22CSE140 (22UG010273)**
DIPTESH NARENDRA	**22CSE224 (22UG010357)**

Future Mesh Enhancing Employability Through Mentorship and Smart Job Matching

We, the undersigned, after reviewing the project report mentioned above and the official record book(s) of the student, hereby state our approval of the **7th Semester Sessional Examination report** submitted in partial fulfilment of the requirements for the award of the degree of **B.Tech** in the **Department of Computer Science and Engineering** under **GIET University, Gunupur**. We are satisfied with the quality, correctness, and originality of the work.

______________________________ ______________________________

Project Guide Project Coordinator

____

Head of the Department

GANDHI INSTITUTE OF ENGINEERING AND TECHNOLOGY UNIVERSITY, GUNUPUR
DIST-RAYAGADA, (ODISHA) PIN-765022

BONAFIDE CERTIFICATE

The software system developed in this project, entitled **Future Mesh: Enhancing Employability Through Mentorship and Smart Job Matching,** was carried out in the Department of Computer Science and Engineering, School of Engineering and Technology. The work reported here is original and does not form part of any other project or dissertation based on which a degree or award was conferred on an earlier occasion or to any other scholar.

We understand the University's policy on plagiarism and declare that this software project and its documentation are my own work, except where specific acknowledgements have been made. The content has not been copied from any other source nor previously submitted for any award or assessment.

Signature of the Students

Countersigned by

Signature of the Supervisor

Dr. Gitanjali Mishra
SUPERVISOR
Assistant Professor
Department of Computer Science
and Engineering, GIET
University, Gunupur (Odisha)

GANDHI INSTITUTE OF ENGINEERING AND TECHNOLOGY UNIVERSITY, GUNUPUR

DIST-RAYAGADA, (ODISHA) PIN-765022

This is to certify that the **7th Semester Sessional Examination report** entitled **Future Mesh Enhancing Employability Through Mentorship and Smart Job Matching** submitted to the Department of Computer Science & Engineering, School of Engineering & Technology, GIET University, in partial fulfilment of the requirements for the award of the degree of BTECH in Computer Science & Engineering, is a record of project work carried out by **Abhisek Panda, Debabrata Mishra, Diptesh Narendra** under my supervision and guidance.

All help received by him from various sources has been duly acknowledged.

No part of this project has been submitted elsewhere for the award of any other degree.

Signature of the Supervisor

Dr. Gitanjali Mishra
SUPERVISOR
Assistant Professor
Department of Computer Science
and Engineering, GIET
University, Gunupur (Odisha)

GANDHI INSTITUTE OF ENGINEERING AND TECHNOLOGY UNIVERSITY, GUNUPUR

DIST-RAYAGADA, (ODISHA) PIN-765022

CANDIDATE'S DECLARATION

We hereby certify that the work which is being presented in the project entitled **Future Mesh Enhancing Employability Through Mentorship and Smart Job Matching** by **Abhisek Panda, Debabrata Mishra, Diptesh Narendra** in partial fulfilment of requirements for the award of degree of B.TECH CSE submitted in the Department of CSE under GIET UNIVERSITY, GUNUPUR, India is an authentic record of my own work carried out during a period From July 15 to October 20 under the supervision of **Dr. Gitanjali Mishra**. The matter presented in this project has not been submitted by me in any other University / Institute for the award of B.TECH Degree.

Signature of the Student

This is to certify that the above statement made by the candidate is correct to the best of my/our knowledge

Signature of the SUPERVISOR

The 7th Semester Sessional Examination of (Abhisek Panda, Debabrata Mishra, Diptesh Narendra) has been held on ………………………… and accepted.

Signature of Supervisor

Signature of B.TECH Coordinator.   Signature of H.O.D.

ACKNOWLEDGEMENT

We would like to place on record our deep sense of gratitude to **Dr. Gitanjali Mishra**, Dept. of Computer Science and Engineering, GIET UNIVERSITY, GUNUPUR, India for his generous guidance, help and useful suggestions.

We also convey our sincere gratitude to **Prof. Neelamadhab Padhy**, Deputy Dean Computational Science (Research) Dept. of Computer Science and Engineering, GIET UNIVERSITY, GUNUPUR, India, for his stimulating guidance, continuous encouragement and supervision throughout the course of present work.

We also wish to extend our thanks to our colleagues for attending our seminars and for their insightful comments and constructive suggestions, which helped improve the quality of this project.

We are extremely thankful to **Dr. Premanshu Sekhar Rath,** HOD, **Dr. K.Murali Gopal**, Dy. Dean, Computational Sciences, Dept. of Computer Science and Engineering, GIET UNIVERSITY, GUNUPUR, India and **Dr. A. V. N. L. Sharma**, Vice Chancellor, **Dr. N. V Jagannadha Rao**, Registrar, **Dr. P. Vijaya Kumar**, COE, GIET University for providing all the necessary facilities and support during our research work, for providing me infrastructural facilities to work in, without which this work would not have been possible.

Signature of the Students

ABSTRACT

Today's competitive job market, students often face difficulties in finding the right career opportunities and professional guidance despite being part of educational institutions with placement facilities. The major problem lies in the lack of a structured and intelligent platform that connects students, mentors, alumni, and companies in a meaningful and organized manner. To address this issue, FutureMesh has been designed as a unified web-based platform that enhances employability through mentorship and smart job matching.

This system aims to bridge the gap between education and industry by bringing together all stakeholders on a single platform. Students can register, fill in their profile and submit their resume, connect with mentors, and apply to the positions offered openings that match their qualifications. Alumni can register as mentors, interact directly with students through real-time chat, and guide them in improving their career paths. Training and Placement (TNP) Cells, HODs, and Admins can manage job postings, approve applications, and monitor placement statistics efficiently. Companies can post job openings targeting specific departments or institutions, ensuring that only qualified candidates are shortlisted automatically.

The entire platform functions on a role-based architecture with dashboards designed for each type of user—Student, Alumni, HOD, Admin, TNP Cell, and Super Admin. The system integrates features like mentorship chat, smart job filtering, profile management, application tracking, and analytics visualization. Built on a stack made up of React, Next.js, Node.js, Express.js, TypeScript, and a cloud-based MongoDB Atlas database, the application ensures scalability, security, and responsiveness.

FutureMesh stands as a modern and practical solution to strengthen university–industry collaboration, improve placement efficiency, and empower students with better guidance and employability skills. It combines mentorship, automation, and data-driven insights to create a digital ecosystem for professional growth.

KEYWORDS: Mentorship, Employability, Job Matching, Placement Automation, Alumni Interaction, Role-Based Dashboard, Node.js, React, MongoDB Atlas.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ACRONYMS

SL. NO	ACRONYM	ABBREVIATION
1	HOD	Head of Department
2	TNP	Training and Placement
3	HR	Human Resource
4	UI	User Interface
5	UX	User Experience
6	DB	Database
7	CRUD	Create, Read, Update, Delete
8	API	Application Programming Interface
9	SMTP	Simple Mail Transfer Protocol
10	OTP	One-Time Password
11	AI	Artificial Intelligence
12	UI/UX	User Interface / User Experience
13	JSON	JavaScript Object Notation
14	IDE	Integrated Development Environment
15	CSS	Cascading Style Sheets
16	JS	JavaScript
17	TS	TypeScript
18	DBMS	Database Management System
19	NoSQL	Non-Structured Query Language
20	CRUD Ops	CRUD Operations

CHAPTER 01

INTRODUCTION

1.1 PURPOSE OF THE PROJECT

FutureMesh project is to build a structured, intelligent, and reliable digital ecosystem that strengthens the bridge between students, alumni, companies, and university placement authorities. In most institutions today, placement activities and mentorship interactions operate through disconnected channels, such as emails, WhatsApp groups, spreadsheets, and informal alumni networks. This scattered environment creates delays, miscommunication, and lost opportunities for students who rely heavily on timely information and proper guidance.

A primary purpose of FutureMesh is to centralize all placement-related and mentorship activities under one unified platform. By integrating job postings, student eligibility filtering, alumni guidance, and real-time tracking features, the system aims to reduce the burden on Training & Placement cells and enhance the experience of students preparing for careers.

Another important purpose of the project is to empower students with a professional space where they can create complete digital profiles, upload resumes, showcase academic projects, and receive personalized job recommendations based on their skill set and academic background. With increasing competition in the job market, students need an environment that encourages them to grow continuously, stay updated on opportunities, and access mentorship when required.

FutureMesh is also designed to encourage alumni involvement by providing them with a dedicated mentoring interface. Alumni play a crucial role in guiding students through real-world insights, interview strategies, resume improvements, and professional advice. However, lack of a formal communication channel prevents structured mentorship. This project seeks to overcome this gap by giving alumni a platform where they can interact with students meaningfully and securely.

From the university's perspective, the purpose of the project includes reducing administrative workload through automation. Placement officers often spend significant time filtering eligible candidates, updating records manually, and tracking the progress of multiple job drives. FutureMesh automates these repetitive tasks through filters, approval mechanisms, and dashboards, allowing administrators to focus on improving overall placement outcomes.

For companies and recruiters, the system serves the purpose of offering a verified, targeted recruitment channel where they can reach eligible students directly. Instead of managing hundreds of unfiltered applications, recruiters gain access to department-wise, skill-wise, and CGPA-wise filtered candidate lists.

1.2 Motivation for the Project

The primary motivation behind developing this project arises from the growing need to connect academic learning with real-world employment opportunities. Despite achieving strong academic results, many students struggle to secure jobs that match their skills and interests. The root cause lies not in the lack of talent but in the absence of structured mentorship, career guidance, and visibility into relevant job openings.

Over the years, universities have recognized that even though they produce skilled graduates, many students remain uncertain about how to apply their knowledge effectively in professional environments. Traditional placement processes are often manual and time-consuming, leaving both students and recruiters dissatisfied. This motivated the development of a platform that automates these processes while creating a collaborative space where learning meets opportunity.

FutureMesh was conceived with the vision of building a digital bridge between students, alumni, placement departments, and companies. The platform empowers students to explore career options aligned with their skill sets and interests, while also providing access to mentorship from experienced alumni. Alumni, in turn, can share practical industry insights, helping students prepare better for interviews, internships, and job roles.

Another driving factor for this project is the inefficiency observed in the current Training and Placement (TNP) systems. Placement officers often spend a significant amount of time managing records, verifying student eligibility, and communicating with companies. These repetitive administrative tasks can easily be automated, allowing them to focus on improving placement strategies and strengthening industry relations.

The project is also motivated by the need to create a smart recommendation system that personalizes job opportunities for students. By analyzing academic performance, skills, and experience, the system can suggest the most relevant roles, increasing the chances of successful placement. Furthermore, it enables companies to directly target eligible candidates without filtering through unnecessary applications.

Lastly, FutureMesh promotes a sense of community and continuity within the institution. It encourages alumni engagement, enhances mentorship culture, and ensures that every student—regardless of their background—has access to proper guidance, growth opportunities, and a path toward employability.

In essence, the motivation for this project is to transform the existing disconnected placement environment into a structured, data-driven, and interactive ecosystem that benefits every stakeholder involved.

1.3 Problem Definition / Statement

In the present academic and professional landscape, the process of campus placement and student mentorship remains highly fragmented and inefficient. Students often rely on multiple uncoordinated sources such as social media groups, informal chats, or third-party job portals to find career opportunities and professional guidance. This scattered approach leads to misinformation, missed opportunities, and a lack of proper direction in career planning.

Placement officers and university administrators face additional challenges as most existing systems are manual or partially digitized. They are responsible for maintaining vast amounts of student and company data, tracking job applications, shortlisting candidates, and managing interview schedules — all of which are prone to human error and delays. Without an integrated system, it becomes difficult to generate meaningful analytics or monitor placement progress in real time.

Companies, on the other hand, struggle to identify and reach eligible candidates efficiently. They often receive applications from students who do not meet the eligibility criteria or belong to unrelated departments. This not only wastes time but also reduces the effectiveness of recruitment drives. The lack of a targeted communication channel between recruiters and the university further complicates the process.

Another major gap exists in mentorship and alumni engagement. While alumni networks exist on social media, they are informal and lack a structured approach to connecting students with experienced professionals. As a result, students miss the opportunity to gain valuable career advice, resume feedback, and interview preparation tips from individuals who once walked the same path.

The FutureMesh platform addresses these issues by introducing a centralized, intelligent, and role-based system that connects students, alumni, placement officers, and company recruiters in a single digital environment. It streamlines operations by automating tasks such as student shortlisting, application tracking, and job posting approvals. Additionally, it offers a built-in mentorship module that allows students to directly communicate with alumni for guidance, thereby encouraging continuous learning and career growth.

Through data-driven decision-making, FutureMesh enables universities to monitor placement performance, identify skills in demand, and bridge the gap between education and employability. This holistic approach ensures transparency, efficiency, and collaboration all participating groups.

1.4 Objectives of the Project

The primary objective of the FutureMesh project is to develop a unified and intelligent digital ecosystem that bridges the gap between students, universities, alumni, and companies. The platform focuses on enhancing employability through mentorship, smart job recommendations, and automated placement management. The specific objectives of the project are as follows:

- To design and develop a single integrated platform that connects students, alumni, placement officers, HODs, and company recruiters, ensuring smooth and transparent communication among all stakeholders.
- To allow students to create comprehensive digital profiles including academic details, skills, certifications, projects, and resumes, enabling them to apply for job roles that best match their eligibility and career goals.
- To establish a structured mentorship system where alumni can register as mentors and provide one-to-one guidance to students through real-time chat, discussions, and mentorship requests.
- To automate placement workflows such as job posting, student shortlisting, eligibility verification, and application tracking, reducing manual work for placement officers and minimizing errors.
- To implement role-based dashboards for Students, Alumni, HODs, Admins, and Super Admins, ensuring that each user accesses only relevant features and data according to their responsibilities.
- To generate detailed analytics and reports that help placement cells, HODs, and administrators make data-driven decisions regarding recruitment trends, student performance, and alumni participation.
- To enhance alumni–student engagement by creating an interactive space for mentorship, experience sharing, and career guidance, fostering a stronger academic–industry connection.
- To enable companies to post job openings and reach suitable candidates directly through the platform, filtering applications based on department, CGPA, and skill requirements.
- To include a project showcase and resume management module for students, allowing them to highlight technical skills, internships, and research projects for better visibility among recruiters.
- To integrate smart job recommendation features using predefined filters and data analytics so that students receive job suggestions aligned with their academic profile.

1.5 Scope of the System

The scope of the FutureMesh system has been defined to cover every essential activity, interaction, and process involved in the academic placement and mentorship environment. The system is designed to offer a complete digital solution that brings together students, alumni, company recruiters, departmental authorities, and placement administrators under a single, coordinated platform. By automating the most frequently used processes and integrating real-time communication, the system ensures that the entire placement cycle—from job posting to student selection—runs smoothly, efficiently, and without the delays associated with traditional manual methods.

At its core, the platform provides students with an organized space where they can manage their academic profile, maintain updated resumes, showcase project work, apply for suitable job roles, and follow the progress of their applications. The system offers students a structured environment that enhances their career preparedness and helps them stay informed about new opportunities. Additionally, the mentorship section of FutureMesh expands the scope of learning by enabling students to connect with alumni who share relevant industry experience, offering guidance that goes beyond classroom knowledge.

For alumni, the system serves as a bridge that reconnects them with the institution in a meaningful way. Alumni often wish to contribute to student growth but lack a formal platform to do so. FutureMesh fills this void by providing a dedicated interface where alumni can respond to mentorship requests, engage in real-time discussions, and offer personalized career advice. This aspect of the system enables a stronger bond between past and current students, fostering a culture of professional support within the institution.

The role of company recruiters is equally important within the scope of the system. FutureMesh provides companies with a streamlined method for posting job opportunities, selecting eligibility criteria, reviewing candidate profiles, and managing recruitment activities. The system ensures that recruiters receive applications only from eligible candidates, reducing the time spent filtering and evaluating large volumes of student submissions. This creates a more focused and effective recruitment environment, benefiting both companies and applicants.

Department heads and Training & Placement (TNP) cells hold a central position in managing placement activities, and the system is designed to support them extensively. Through FutureMesh, HODs and placement officers can oversee job postings, approve or reject company requests, shortlist eligible students, and monitor the progress of recruitment drives. The system reduces paperwork, eliminates redundancy, and provides a clear structure for maintaining departmental placement records. This digital support allows academic authorities to manage responsibilities more effectively and with improved transparency.

The administrative scope of FutureMesh extends further through the inclusion of a Super Admin role, which provides full control over user management, platform security, feature monitoring, and data consistency. The platform ensures that all user interactions follow proper authorization rules and that sensitive information is protected at every level. Administrators can oversee activities across the entire system, making it possible to maintain a high standard of data integrity and operational reliability.

Beyond individual user roles, the broader scope of FutureMesh lies in its ability to support large-scale institutional networks. The system is created with a scalable architecture that can expand beyond a single institution and be implemented across multiple campuses, colleges, or even city-wide educational networks. This makes it possible to form a unified placement environment where multiple institutions can collaborate, exchange opportunities, and share resources through a single digital ecosystem. The platform also includes detailed analytical tools that provide valuable insights into placement performance, industry trends, student readiness, and alumni participation. These analytics help institutions evaluate their strengths, identify areas of improvement, and plan better academic and training strategies.

FutureMesh further extends its scope by ensuring secure authentication, reliable data protection, and smooth accessibility across devices. Its user interface is developed to be responsive on both desktop and mobile systems, making it easier for users to access essential features anytime and from anywhere. The system maintains data confidentiality, supports encrypted communication, and follows modern usability principles to create a platform that is not only functional but also comfortable for long-term use.

1.6 Methodology Overview

The development of the FutureMesh platform follows a structured and systematic methodology designed to ensure accuracy, clarity, and smooth implementation of all major features. The methodology focuses on understanding the needs of each stakeholder, transforming those needs into a practical design, and finally developing a working system that is stable, secure, and efficient. At the beginning of the project, a detailed requirement study was conducted to clearly understand the expectations of students, alumni, company recruiters, HODs, placement officers, and administrative authorities. This phase helped identify the challenges present in the traditional placement workflow and provided a clear vision of the features the new platform must accommodate. The information gathered in this phase formed the foundation on which every subsequent stage of development was built.

Once requirements were finalized, the next step was to design the system's overall structure. This included preparing the system architecture diagram, database models, UML diagrams, data flow diagrams, and user interface layouts. Each user role was studied separately to create dedicated dashboards that aligned with their responsibilities. The design phase also involved preparing wireframes and navigation flows that would guide the development of the user interface. Proper planning during this stage ensured that the platform would be easy to use, scalable, and capable of accommodating future enhancements without major structural changes.

The frontend development phase focused on creating visually clean, responsive, and interactive interfaces using modern web technologies such as React, Next.js, TypeScript, and JavaScript. These technologies were chosen because they offer fast performance, ensure code reusability, and support dynamic real-time updates—features that are crucial for modules like mentorship chat, job updates, and analytics dashboards. Special attention was given to consistency and ease of use, ensuring that each dashboard displayed information clearly and allowed quick navigation between different system sections.

Backend development was carried out using Node.js and Express.js, chosen for their speed, scalability, and ability to handle real-time interactions effectively. During this phase, all core functionality—such as user authentication, job posting, student shortlisting, mentorship communication, and analytics data retrieval—was implemented through secure and well-structured REST APIs. MongoDB Atlas was selected as the primary database because it supports flexible document-based storage, secure cloud hosting, and efficient handling of complex relationships between users, jobs, and mentorship activities. This ensured that all data was stored safely and could be retrieved quickly whenever required.

After completing both the frontend and backend components, the integration phase involved connecting the interfaces with the backend APIs and ensuring seamless communication between all modules. This phase played a critical role in validating the accuracy of data flow across the system. Modules such as job posting, student application tracking, alumni chat, notifications, and administrative controls were tested to confirm that information moved correctly between the user interface, the server, and the database. This stage helped identify minor mismatches and inconsistencies, which were corrected promptly to ensure smooth operation.

Testing formed another essential part of the methodology. Multiple forms of testing—including unit testing, integration testing, system testing, and user acceptance testing—were conducted to verify the reliability, performance, and security of the platform. Each module had to meet specific functional requirements, and any issues detected during this stage were resolved immediately. The mentorship chat was tested for responsiveness, while job workflows were checked for accuracy, ensuring that job postings, approvals, and shortlisting worked as intended. This phase ensured that the system was stable and ready for real-world use.

The final step in the methodology involved deployment, where the fully developed system was hosted securely on Vercel. This platform was chosen for its fast performance, easy integration with Next.js, and reliable deployment workflow. Deployment ensured that the platform became accessible to real users with proper security measures, role-based access control, and protected backend connections. This final stage ensured that the system was not only functional but also available online in a stable and scalable environment.

1.7 Organization of the Report

This report is organized into structured chapters to explain the system clearly:

- **Chapter 1** introduces the project background, motivation, objectives, and overall purpose.
- **Chapter 2** discusses existing systems, proposed solutions, feasibility studies, and requirement analysis.
- **Chapter 3** explains the system design through architecture diagrams, UML diagrams, ER models, and user interface layouts.
- **Chapter 4** details the implementation of the system, including technologies used and module development.
- **Chapter 5** covers testing procedures, test cases, results, and bug resolution.
- **Chapter 6** presents the results, output screenshots, and discussions.
- **Chapter 7** concludes with achievements, limitations, and future scope.

Each chapter builds progressively toward understanding the design, implementation, and evaluation of FutureMesh.

1.8 SUMMARY

This chapter highlighted the foundation of the FutureMesh project, starting with the purpose, technological background, methodology, and objectives. It also explained the motivation behind the system, the major problems identified in current placement environments, and how the report is organized.

The chapter emphasized the motivation for developing a structured digital platform by describing the shortcomings of existing systems and the communication gaps that currently affect campus recruitment and mentorship activities. It clearly defined the problem statement and demonstrated how these issues limit efficiency, transparency, and student support in many academic environments. Furthermore, the chapter detailed the methodology chosen for the project, providing an overview of how requirements were identified, system designs were formulated, and development stages were planned.

CHAPTER 02

System Analysis

2.1 Existing System and Its Limitations

In many educational institutions, the current system for managing placements, career guidance, and alumni interaction is still fragmented and outdated. Most colleges rely on a combination of manual record-keeping, basic web portals, and external job platforms to support student placements. Students usually depend on third-party job portals such as LinkedIn, Naukri, Indeed, or other commercial sites to search for opportunities, which may or may not be relevant to their academic background or skillset. Meanwhile, university placement cells maintain student records, company details, and application information using spreadsheets, emails, or partially automated internal systems. These tools lack the sophistication needed to handle large volumes of data or perform real-time communication effectively.

The interaction between students and alumni also remains largely informal and unstructured. Alumni guidance often happens through personal connections or social media, which does not ensure equal access for all students. Traditional systems do not provide a dedicated space for mentorship, making it difficult for students to receive consistent guidance on career decisions, interview preparation, or skill development. Because of this, the existing environment becomes disconnected, forcing students to switch between multiple platforms just to complete basic tasks related to placements and career planning.

For university staff, managing placement activities becomes challenging because most of the workflows, such as collecting resumes, verifying student eligibility, updating job listings, shortlisting students, and tracking application progress, require significant manual effort. As the number of students grows, these manual processes become slower and more prone to human errors. Training & Placement (TNP) cells often struggle to communicate updates efficiently, causing delays in sharing job notifications, interview schedules, or company requirements. This lack of automation reduces transparency and creates confusion among students during recruitment seasons.

Companies face their own difficulties when participating in campus recruitment through existing systems. Recruiters often depend on emails, offline forms, or simple portals that do not provide advanced filtering or structured student data. They cannot easily identify students based on department, year of graduation, required skills, CGPA, or experience. As a result, companies receive a large number of unqualified applications, increasing the time required to shortlist candidates. Without a centralized system linking all stakeholders, companies find it hard to coordinate with academic departments, leading to unpredictable delays and inefficient hiring processes.

The limitations of the existing system originate from its lack of integration and absence of intelligent automation. There is no unified platform that connects students, alumni, departments, and companies in real time. The system does not support personalized job recommendations or structured mentorship sessions. Institutions also lack the ability to generate quick analytics on placement performance, student eligibility distribution, or skill gaps across departments. Most importantly, the existing tools cannot scale effectively when large batches of students or multiple departments need to be managed simultaneously.

Another critical gap is the absence of secure role-based access. Many current systems offer the same interface to all users, without distinguishing between administrative roles, student privileges, alumni functions, or company rights. This leads to confusion, data overload, and security concerns. In some cases, sensitive student information is stored without proper access control, making it vulnerable to unauthorized use. Additionally, existing portals often fail to accommodate new technologies such as real-time chat, automated workflows, or cloud-based scale-up features.

Because of these limitations, students frequently miss out on important updates or opportunities, alumni remain disconnected from the institution, placement officers handle repetitive administrative tasks unnecessarily, and companies face difficulty finding suitable candidates. The lack of a centralized digital framework affects the overall efficiency of the placement ecosystem and restricts institutions from providing high-quality career support to their students. This situation clearly indicates the need for a modern, integrated, and intelligent system like FutureMesh, which can bridge all gaps and offer a comprehensive solution tailored to today’s placement and mentorship requirements.

2.2 Proposed System Overview

The proposed system, FutureMesh, is envisioned as an integrated digital framework that brings together all major aspects of campus placements, mentorship, and professional networking into a single unified platform. Unlike the fragmented traditional systems, FutureMesh connects students, alumni, academic departments, placement authorities, and company recruiters through dedicated dashboards and well-defined access controls. The primary purpose of the system is to eliminate the communication gaps that currently exist and to create a seamless channel through which career-related activities can be carried out efficiently. By combining automation, real-time communication, and intelligent data processing, the system aims to offer a smoother, faster, and more transparent placement environment for every stakeholder involved.

At the core of FutureMesh is its role-based structure, which ensures that each user interacts with the system according to their responsibilities and permissions. Students receive a personalized interface where they can create detailed profiles, upload resumes, explore job opportunities, and track the progress of their applications. The platform further supports them by offering a structured mentorship system that connects them with alumni based on their academic interests, skills, or career goals. This connection allows students to receive timely guidance, interview tips, and insights into real industry expectations. Such interactions help bridge the knowledge gap that often separates academic preparation from professional requirements.

For alumni, the system serves as a platform to reconnect with their institution in a meaningful way. They are provided with tools to communicate directly with students, share their experiences, provide career-related suggestions, and support the next generation of learners. This structured mentorship framework ensures that students have access to guidance whenever they need it, and alumni can contribute without the need for informal channels or external platforms. The mentorship module also records Chats, maintaining professionalism and data security throughout the communication process.

The involvement of company recruiters is another central component of the proposed system. Companies can create profiles and post job openings directly on the platform. They are given access to relevant student data through intelligent filtering mechanisms that sort candidates based on department, CGPA, skills, certifications, or experience. Instead of sifting through hundreds of unqualified applications, companies receive a curated list of eligible candidates, saving time and improving the overall efficiency of recruitment drives. This structured process also ensures fairness, as every candidate is evaluated based on transparent and predefined criteria.

Academic authorities such as HODs and TNP cells are provided with dashboards tailored to administrative and managerial tasks. They can verify student profiles, review job postings from companies, approve or reject listings based on institutional norms, and shortlist students for various openings. These tools significantly reduce the manual workload that typically burdens placement departments. The system enables them to track the status of each job posting, monitor student participation, and manage communications in a centralized, organized manner. Automated notifications ensure that students receive timely updates regarding shortlisting, interview schedules, or announcements from the placement cell.

Administrators and super administrators hold complete oversight of the system. Their dashboards include user management controls, data monitoring tools, and access to analytics that summarize placement performance, mentorship participation, and student engagement. They can track the overall health of the platform, resolve technical issues, and verify organizational alignment. The availability of automated analytics supports strategic decision-making and helps improve placement practices over time.

The technical foundation of FutureMesh is built on modern, scalable technologies such as React, Next.js, TypeScript, Node.js, Express.js, and MongoDB Atlas. These tools have been selected for their stability, fast performance, and ability to handle real-time data interactions. The platform supports dynamic page rendering, asynchronous data flow, and cloud-based storage, ensuring that thousands of users can interact without performance degradation. Strong security practices, including encrypted communication and role-based access, protect user data from unauthorized access.

2.3 FEASIBILITY STUDY

A feasibility study is essential to evaluate whether the proposed system can be developed, implemented, and maintained successfully. For FutureMesh, feasibility was assessed across three primary dimensions—technical feasibility, operational feasibility, and economic feasibility. Each dimension highlights the strengths of the proposed system and confirms that the project is practical, sustainable, and suitable for institutional deployment. The study ensures that the chosen technologies, user processes, and financial aspects align with the long-term goals of the platform.

2.3.1 TECHNICAL FEASIBILITY

FutureMesh is technically feasible because it is built on modern, widely adopted, and stable technologies that support scalability, reliability, and long-term maintainability. The selection of **React** and **Next.js** for the frontend ensures that users experience fast loading times, smooth navigation, and responsive layouts across different devices. These technologies are well-known for their developer-friendly structure and reusable components, which simplify interface updates and reduce the time required for future enhancements.

On the backend, the use of Node.js and Express.js enables high-performance server-side operations. These frameworks are capable of handling concurrent requests efficiently, an essential requirement in a system serving many students, alumni, companies, and administrators simultaneously. The asynchronous architecture of Node.js helps the system manage real-time modules, such as mentorship chat and live notifications, without delays or bottlenecks.

Data management in FutureMesh is supported by MongoDB Atlas, a cloud-based NoSQL database known for its scalability, high availability, and secure data storage features. Its document-oriented structure allows flexible storage of student profiles, job postings, messages, and analytics data. The built-in security options, backup features, and automated scaling make it suitable for a system expected to grow with new institutions and user roles.

All selected technologies are compatible with one another and integrate seamlessly through REST APIs. This ensures smooth data exchange and consistent performance across various modules. Additionally, since these technologies follow industry standards, they are future-ready and can be upgraded easily as new versions or features become available. The technical foundation therefore supports reliability, flexibility, and long-term sustainability of the platform.

2.3.2 OPERATIONAL FEASIBILITY

Operational feasibility examines how effectively the system can be used in real-world conditions by different stakeholders. FutureMesh is designed to fit naturally within the daily routines of students, placement officers, company recruiters, alumni, and university administrators. The platform simplifies existing workflows by centralizing all placement-related activities, reducing dependence on emails, spreadsheets, or external job portals.

The system is intentionally developed with a clean and intuitive interface, ensuring that users require minimal training to navigate its features. Students can effortlessly create profiles, upload resumes, apply to jobs, and request mentorship without needing technical expertise. Similarly, alumni can initiate and manage mentorship conversations easily, making the system convenient even for users who are less familiar with modern digital platforms.

HODs, TNP cells, and admins benefit from dashboards that present information in a well-organized manner, allowing them to handle job postings, approvals, and student shortlisting quickly. The system's ability to automate notifications and generate placement reports helps reduce manual workload, making everyday operations more efficient. Since the platform is accessible from any device with an internet connection, stakeholders can perform tasks remotely without depending on a physical office setup.

Furthermore, the system supports long-term operational stability because its workflows align with existing academic and administrative processes. The system's modular design ensures that new features can be integrated smoothly as institutional needs change. Overall, FutureMesh enhances operational efficiency by offering a structured, consistent, and user-friendly environment for all placement activities.

2.3.3 ECONOMIC FEASIBILITY

FutureMesh proves to be economically feasible due to its use of open-source technologies and low-maintenance infrastructure. The core development tools—including React, Next.js, Node.js, Express.js, and MongoDB—do not require expensive licensing fees. This eliminates the financial burden typically associated with proprietary software and makes the platform affordable for academic institutions.

The database is hosted on MongoDB Atlas, which offers flexible pricing models and allows institutions to scale their usage based on actual requirements. This pay-as-you-grow approach prevents unnecessary expenditure and supports budget-friendly expansion when the system is adopted by more users or multiple institutions. Deployment on modern cloud platforms such as Vercel also reduces hosting costs while offering reliable uptime, automated deployment pipelines, and built-in performance optimization.

Hardware investment is minimal since the platform operates entirely online. Institutions do not need to purchase servers or additional equipment; existing computers and internet facilities are sufficient.

2.4 Requirements Analysis

2.4.1 Functional Requirements

- User Authentication: Login and registration for different roles (Student, Alumni, HOD, Admin, Company).
- Profile Management: Users can update personal information, upload resumes, and manage details.
- Job Posting and Application: Companies can post jobs, and students can apply if eligible.
- Mentorship System: Alumni can send and receive mentorship requests, and students can chat directly.
- Application Tracking: Students can check application status (Applied, Under Review, Shortlisted, Interview).
- Admin & HOD Control: Admins can manage users and approve job posts.
- Analytics and Reports: Placement data visualized using charts and graphs.
- Notification System: In-platform and email notifications for updates.

2.4.2 Non-Functional Requirements

- Performance: The system should handle multiple users and requests simultaneously.
- Usability: Interfaces should be simple, intuitive, and mobile-friendly.
- Reliability: Data should remain consistent and accessible at all times.
- Security: Role-based access and encrypted data handling ensure privacy.
- Scalability: Can be expanded to support multiple institutions and departments.
- Maintainability: Modular code structure for easy updates and debugging.

2.5 Problem Context

Below is the textual explanation of the use case context:

- Actors: Student, Alumni, HOD, Admin, Company HR, Super Admin.
- Student: Registers → Creates Profile → Applies for Jobs → Chats with Alumni → Tracks Application Status.
- Alumni: Registers → Accepts Mentorship Requests → Guides Students via Chat.
- HOD: Reviews Job Posts → Shortlists Students → Sends Notifications.
- Admin/TNP: Approves Jobs → Monitors Applications → Generates Reports.
- Company HR: Posts Job → Reviews Applications → Shortlists Eligible Students.
- Super Admin: Monitors entire platform and user activities.

Each module interacts with the database through secure APIs ensuring accurate data flow and management.

CHAPTER 03

System Design

3.1 System Architecture Diagram (FlowChart)

The structure of the FutureMesh platform follows a clear multi-layer web design. The system is divided into three major parts that work together to deliver all its features: the interface used by the users, the server that handles the platform’s logic, and the database where all information is stored.

The user interface is created with React, Next.js, and TypeScript. This section controls everything the user interacts with, including dashboards, profile pages, chat screens, forms, and other visual elements. It is designed to run smoothly for students, alumni, department heads, administrators, and company representatives, offering a clean and responsive experience on different devices.

Behind the interface is the server layer, which runs on Node.js and Express.js. This part handles incoming requests, manages login and verification, processes job postings and student applications, and oversees the messaging features used in the mentorship module.

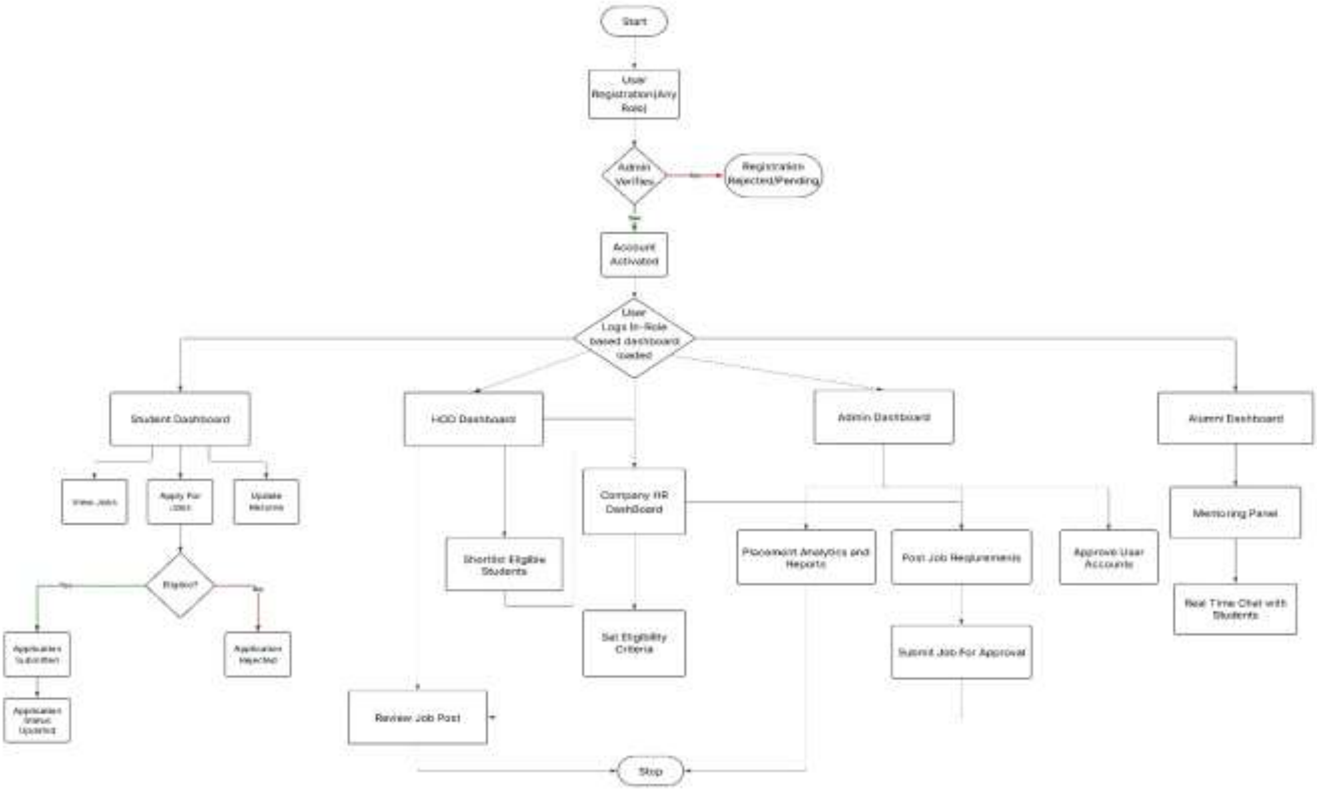


Figure 1.1 (Source: Developed by project team)

3.2 UML Diagrams

3.2.1 Use Case Diagram

The use case diagram shows how each type of user connects with and operates within the system. The key roles involved in these interactions are students, alumni, department heads, administrative staff, the super administrator, and company recruiters.

Students can register, update profiles, apply for jobs, and chat with mentors.

Alumni can register as mentors, respond to student requests, and guide through chat.

HODs and Admins manage job postings, shortlists, and approvals.

Company HRs post jobs and review student applications.

Super Admin oversees all modules and user activities.

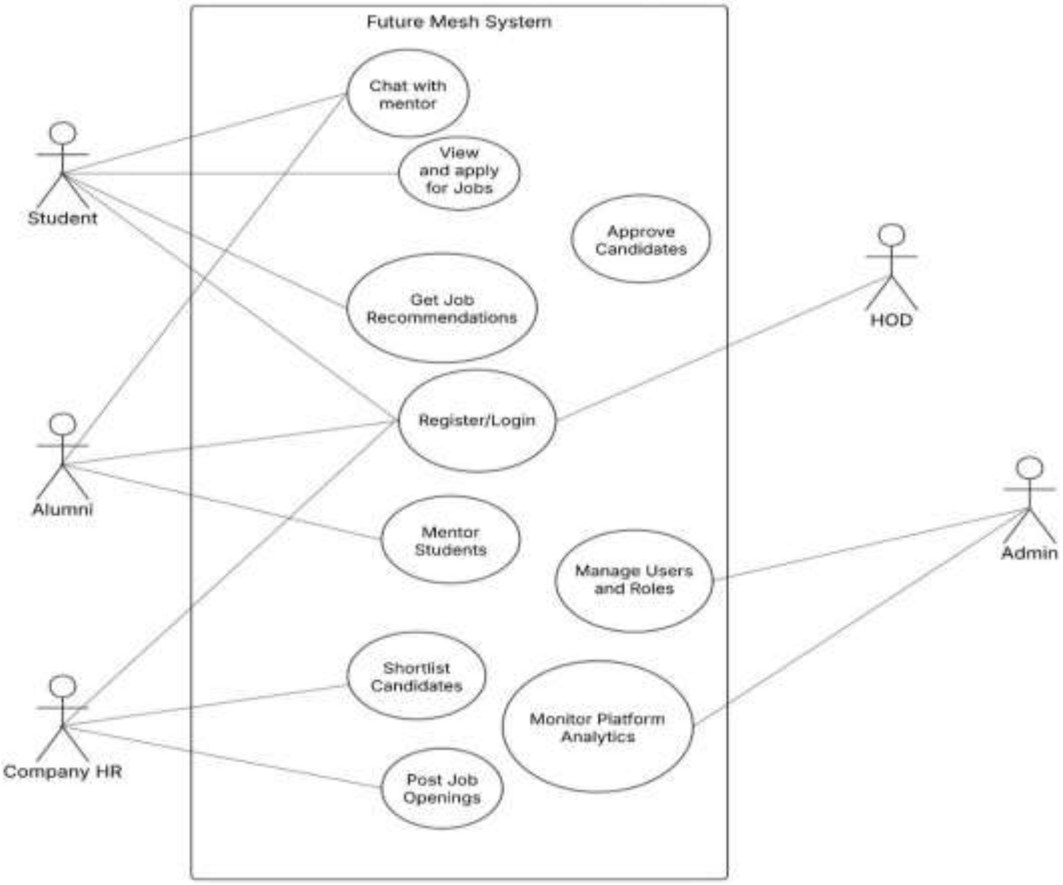


Figure 2.1 (Source: Created by project team)

3.2.2 Class Diagram

The class diagram outlines the core components of the system and illustrates how they are connected to one another. It includes important classes such as User, Student, Alumni, HOD, Admin, JobPost, Application, Chat, and MentorshipRequest, each representing a specific part of the platform's functionality. Each class defines specific attributes and methods. Relationships such as one-to-many (Admin to JobPosts) and many-to-many (Student to JobApplications) are maintained.

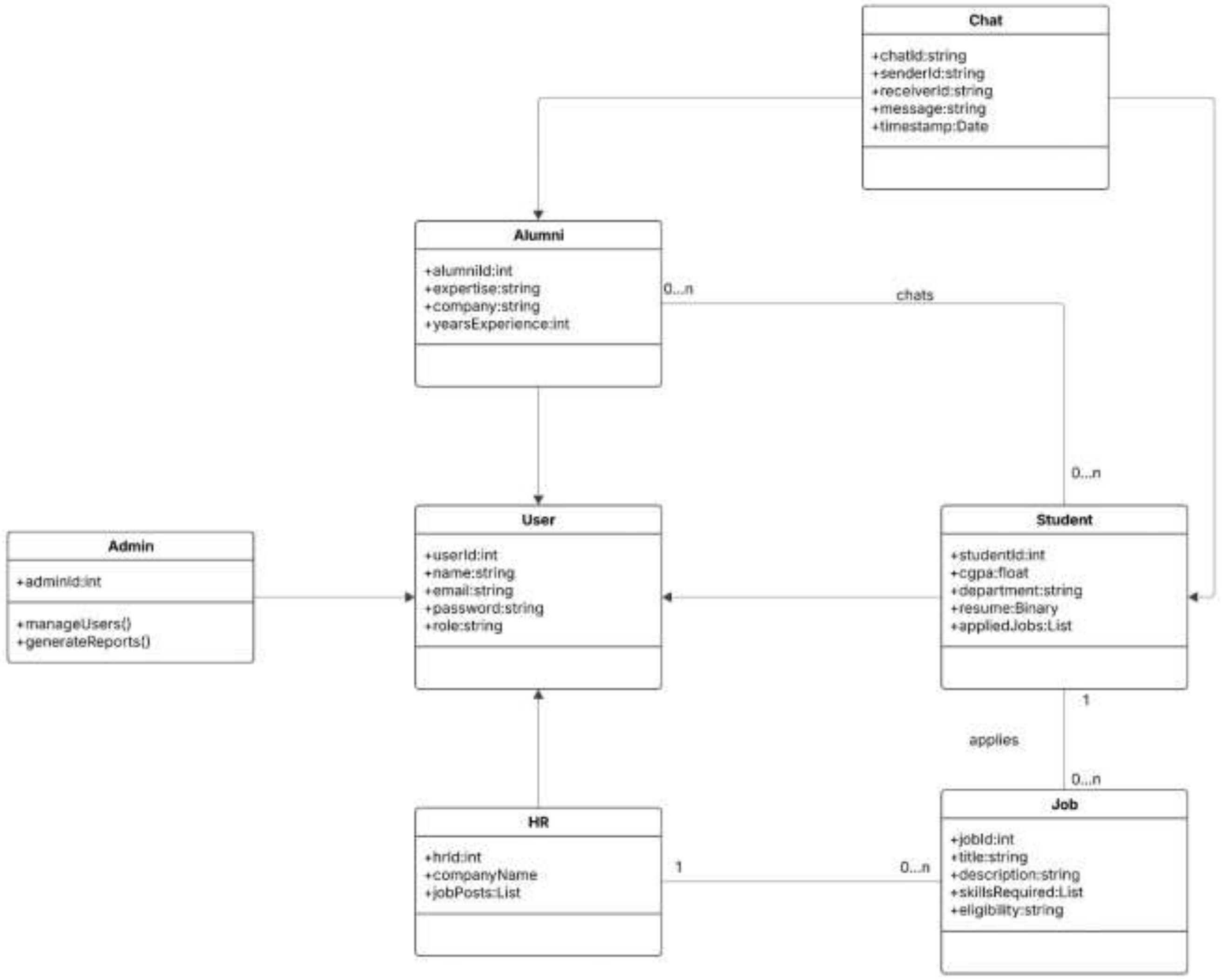


Figure 2.2 (Source: Created by project team)

3.2.3 Sequence Diagram

The sequence diagram explains the step-by-step interaction between users and the system. Example:

A student logs in and views job posts.

The system fetches available jobs through the backend API.

The student applies for a job.

The backend validates eligibility and stores the application in the database.

The Admin and HOD are notified of the new application.

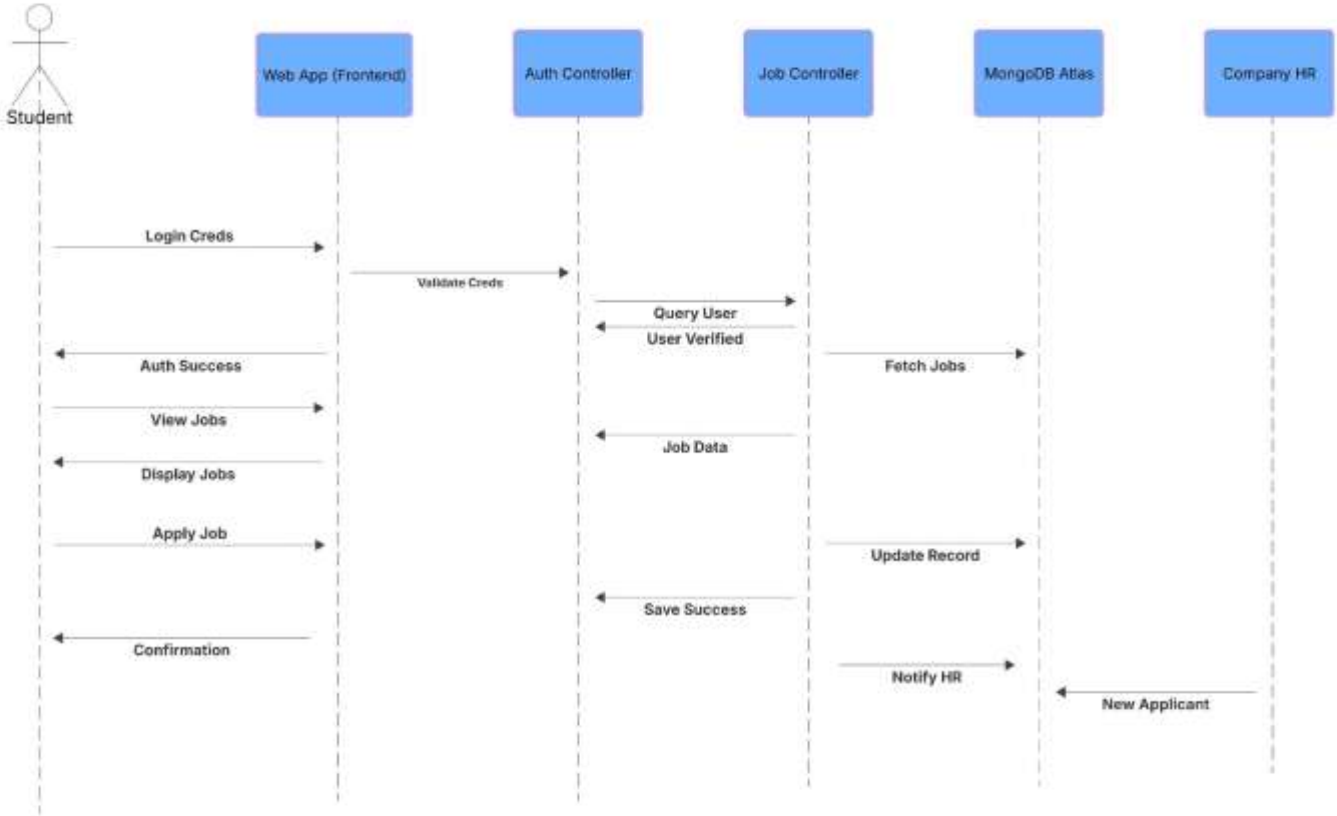


Figure 2.3 (Source: Designed based on project workflow)

3.2.4 Activity Diagram

The activity diagram presents the step-by-step flow of how a job process is handled in the system. It begins with the user login, followed by profile setup, job application submission, HOD shortlisting, and Admin approval. For mentorship, it shows the request sent by the student, acceptance by alumni, and chat interaction.

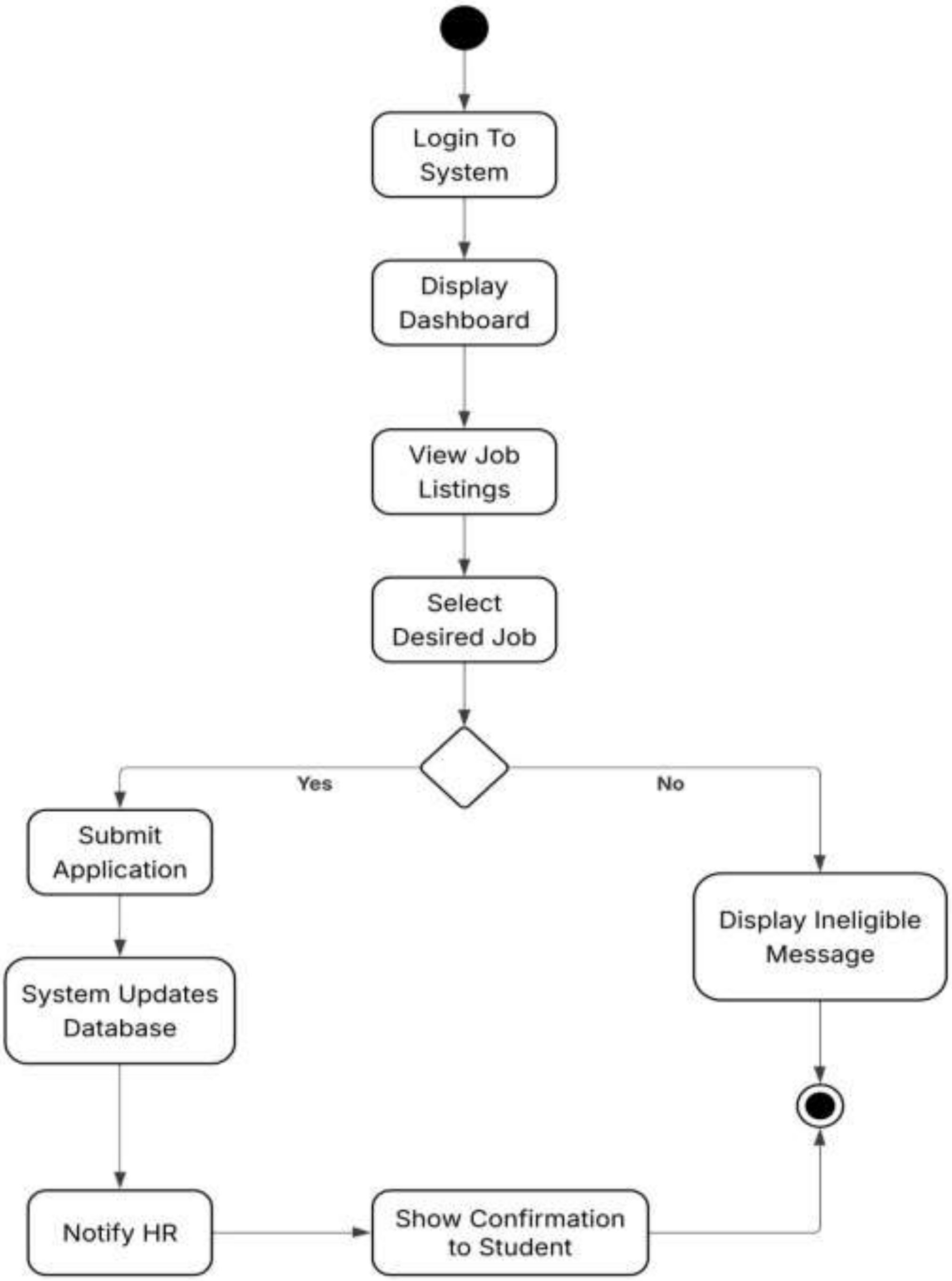


Figure 2.4 (Source: Designed based on project workflow)

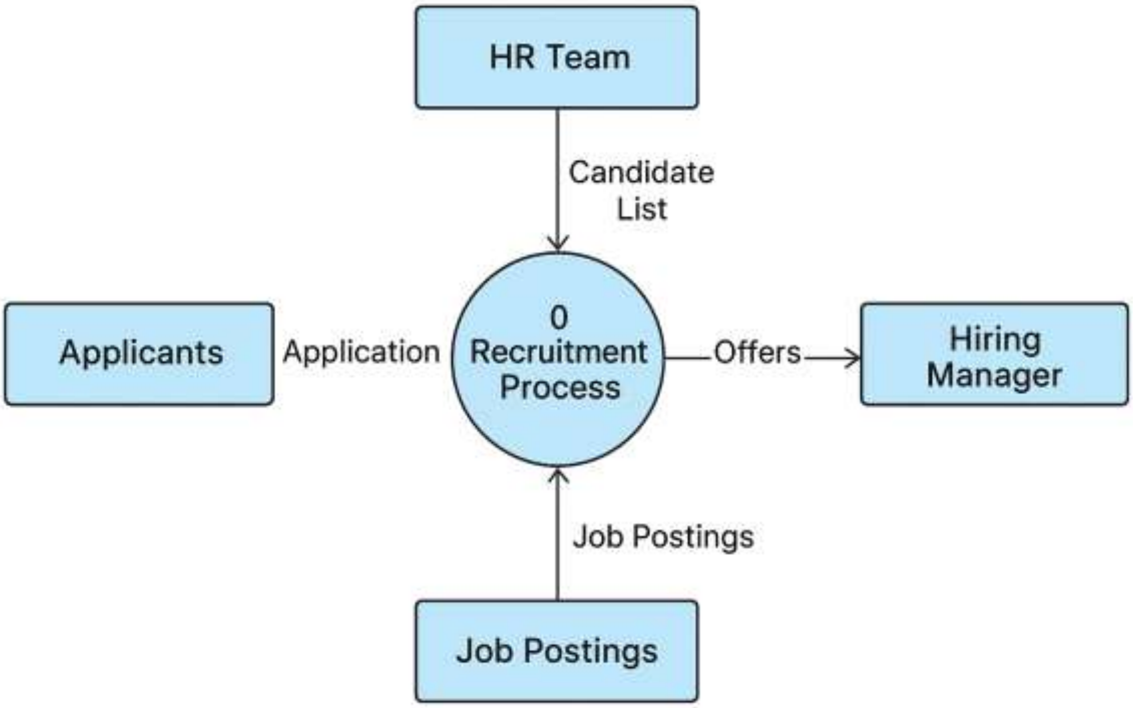
3.2.5 Data Flow Diagram (DFD)

The DFD illustrates the movement of information across the system.

At Level 0, it highlights the basic interaction between users and the platform.

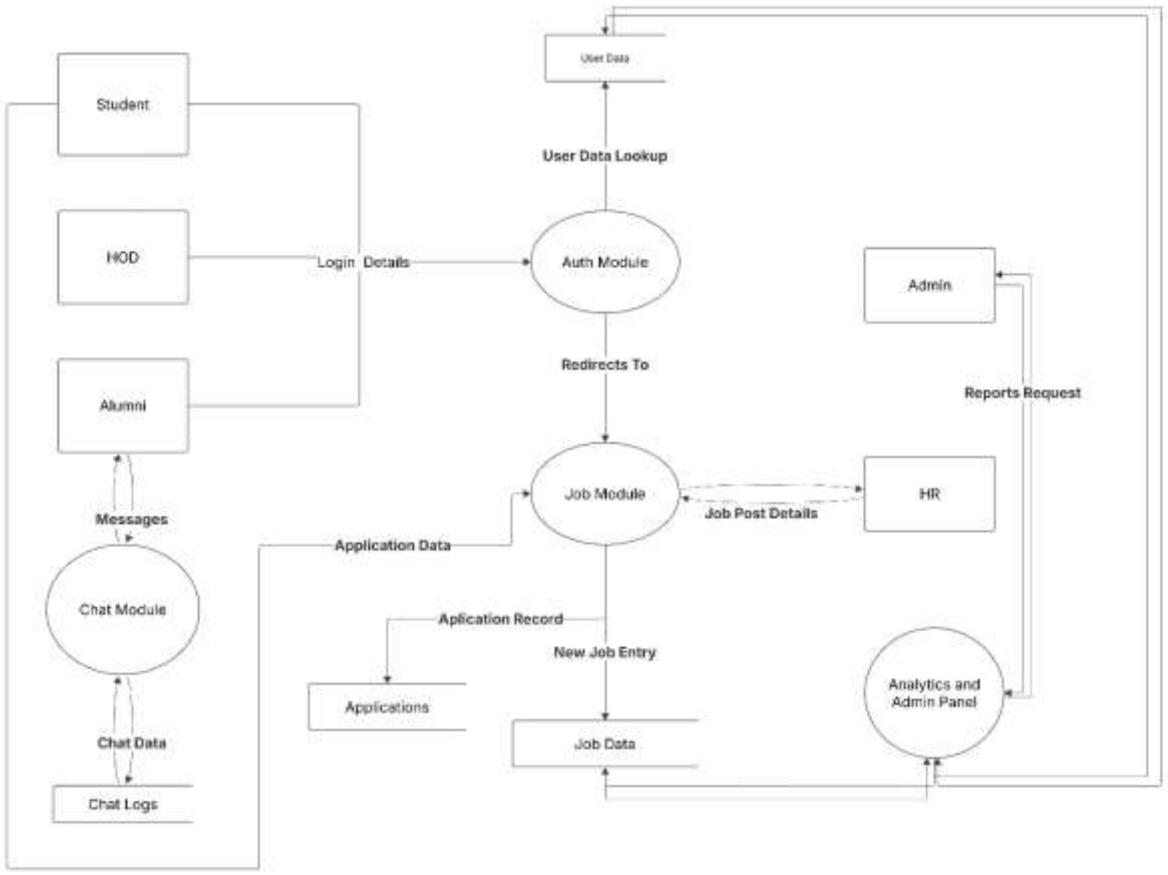
Level 1: Details processes like registration, job posting, application, mentorship, and analytics.

Data stores include user profiles, job data, mentorship chats, and placement reports. Information flows between users and the database through controlled backend APIs.



LEVEL – 0

Figure 3.1 (Source: Designed based on project workflow)



LEVEL - 1

Figure 3.2 (Source: Designed based on project workflow)

3.3 Database Design

3.3.1 ER Diagram

The Entity-Relationship Diagram represents the database structure. Main entities include Users, JobPosts, Applications, Chats, MentorshipRequests, and Departments.

Relationships:

One Admin manages many JobPosts.

One Student can have multiple Applications.

One Alumni can handle multiple MentorshipRequests.

Departments connect Students, HODs, and JobPosts.

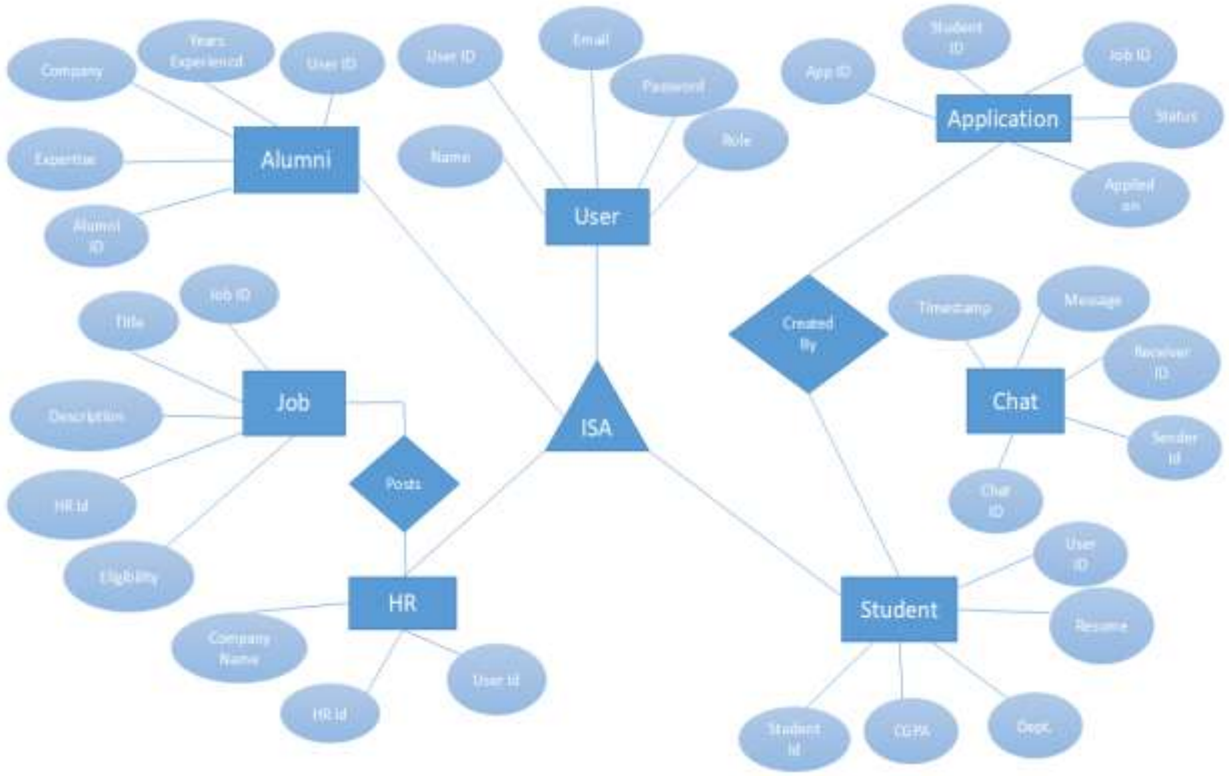


Figure 4.1 (Source: Database structure prepared by project team)

3.3.2 Table Structures

Each collection (table) in MongoDB holds structured documents. For example:

Users: user_id, name, email, role, password, department.

JobPosts: job_id, company_name, eligibility, description, status.

Applications: application_id, student_id, job_id, status.

Chats: chat_id, sender_id, receiver_id, message, timestamp.

3.3.3 Relationships and Keys

Relationships and Keys

In the FutureMesh system, each table or collection in the database is connected to others using unique identifiers. These identifiers make sure that data can be linked correctly and safely between users, job posts, applications, and mentorship records.

Every entry in the database has a primary key, which is a unique value used to identify that record. In MongoDB, this is usually the ObjectID. Relationships between different collections are created using foreign keys or references to these unique IDs. This ensures that information stays consistent and accurate throughout the platform.

Main Relationships:

- User–Profile Relationship:
  Each user (Student, Alumni, HOD, Admin, HR) has one main profile record stored in the Users collection. This relationship is one-to-one.
- Admin–JobPost Relationship:
  Each Admin or TNP Cell member can create and manage multiple job posts. This is a one-to-many relationship since one Admin manages many job entries.
- Student–Application Relationship:
  A single student can apply for several job posts, and each job post can have many student applications. Hence, this is a many-to-many relationship.
- Alumni–Mentorship Relationship:
  One alumni member can mentor multiple students, and a student can have guidance from more than one alumni. This forms another many-to-many relationship, managed through a separate Mentorship collection.
- JobPost–Application Relationship:
  Each job post can have multiple applications linked to it. Each application includes references to both the job ID and the student ID, creating a one-to-many connection between job posts and applications.

- Department–User Relationship:
  Each department is linked with several users like students, alumni, and HODs. This is also a one-to-many relationship because one department includes many users.

- Chat Relationship:
  Each chat message connects a sender and a receiver (student–alumni or student–mentor). The Chat collection stores references to both users' IDs, forming a many-to-many communication pattern.

Keys Used:

- Primary Key: _id (Unique ObjectID generated by MongoDB).
- Foreign Keys/References:
  - user_id → Links to the Users collection.
  - job_id → Links to the JobPosts collection.
  - application_id → Links to Applications collection.
  - alumni_id and student_id → Used in Mentorship and Chat collections.
  - department_id → Connects users to their respective departments.

All these relationships ensure smooth data flow, maintain consistency, and allow the system to display correct information for each user role.

3.4 User Interface Design

3.4.1 Screen Layouts

The system includes modern, responsive layouts for each user role.

Student Dashboard: Profile, Resume Upload, Job Listings, Mentorship Chat.

Alumni Dashboard: Mentorship Requests, Chat Section, Profile Settings.

Admin Dashboard: Job Approvals, User Management, Analytics.

HOD Dashboard: Shortlisting Panel, Department Reports.

Super Admin Dashboard: Full system overview and performance tracking.

3.4.2 Navigation Flow

The navigation flow of the FutureMesh platform is designed to be simple, intuitive, and role-specific so that users can move across different modules without confusion. Once a user logs in, the system automatically identifies their role—Student, Alumni, HOD, TNP Officer, Admin, or Company HR—and redirects them to the corresponding dashboard. This ensures that every user receives only the tools and features that match their responsibilities.

The platform uses a clean sidebar-based layout where major options are arranged in a clear, vertical structure. Each menu item, such as Profile, Jobs, Mentorship, Chat, Notifications, Applications, Analytics, and Admin Controls, opens to its own dedicated page. This avoids overcrowding the interface and makes each section easy to understand. The sidebar remains visible across all screens, allowing users to switch sections without returning to the homepage.

Smooth transitions help users navigate from one module to another without delays. Students can move from their job recommendations page to their application status with a single click. Alumni can switch quickly between mentorship requests and chat conversations. Placement officers can shift from job approvals to student shortlisting without leaving their dashboard.

Every page is linked through a logical flow that mirrors real placement and mentorship activities. For example, clicking on a job takes the user to the job details page, which then leads to the application or eligibility section. Similarly, mentorship requests open directly into chat windows, helping students and alumni stay connected.

Navigation is also optimized for clarity and error-free movement. Back buttons, breadcrumb paths, and clear labels ensure that users always know where they are and how to return to previous pages. Mobile-responsive navigation is also supported, allowing the menu to collapse into a compact layout on smaller screens while still maintaining full functionality.

Overall, the navigation flow is structured to provide a seamless experience across all modules of FutureMesh. It reduces complexity, improves user engagement, and ensures that every action—whether applying for a job, approving a posting, or chatting with a mentor—can be completed quickly and efficiently

CHAPTER 04

SYSTEM IMPLEMENTATION

4.1 Description of Technologies, Frameworks, and Tools Used

The FutureMesh platform is built using modern web development technologies that ensure speed, scalability, and ease of use. The system is designed as a full-stack web application with both front-end and back-end integration.

- Programming Languages:
  - TypeScript and JavaScript: Used in both front-end and back-end for handling logic, data exchange, and dynamic behavior. TypeScript provides type safety, while JavaScript handles the main execution.
- Frameworks and Libraries:
  - React and Next.js: Used to design a responsive and user-friendly interface. React manages UI components, while Next.js provides server-side rendering and better performance.
  - Node.js and Express.js: Used in the backend to create RESTful APIs, handle requests, and manage business logic securely.
- Database:
  - MongoDB Atlas: A cloud-based NoSQL database that stores user profiles, job data, chat messages, mentorship requests, and other placement-related information.
- Development Tools:
  - Visual Studio Code: Used as the primary editor for coding, testing, and fixing errors.
  - Git and GitHub: Helps manage code changes and supports smooth teamwork during development.
  - Vercel: Used for hosting the front-end application securely and efficientl

4.2 Implementation Details (Module-Wise)

- Landing Page and Authentication Module:
  This includes the main home page, registration, and login systems for all user roles — Student, Alumni, HOD, Admin, and Super Admin. The authentication is secured using JWT (JSON Web Token) and encrypted passwords.

- Super Admin Module:
  Handles overall control of the platform. The Super Admin can monitor users, view activity statistics, approve or suspend accounts, and oversee job postings and mentorship interactions.

- Student Module:
  Students can create profiles, upload resumes, apply for jobs, and request mentorship. Their dashboard also shows application status, chat access, and career analytics.

- Alumni Module:
  Alumni can register as mentors, accept mentorship requests, and communicate directly with students using real-time chat. They can update their professional details and guide students based on their experience.

- HOD and Admin Modules:
  The HOD dashboard allows student shortlisting and tracking department-wise placement records.
  Admins manage job posts, verify eligibility, and generate reports for placement analysis.

- Company HR Module:
  Allows companies to create accounts, post jobs, set eligibility criteria, and review student applications.

- Mentorship and Chat Module:
  Connects students and alumni through a live chat system powered by Socket.io. It allows guidance sessions, query discussions, and mentorship tracking.

4.3 Screenshots of Developed Modules / Interfaces

(Screenshots are attached separately in the report, showing major sections such as Student Dashboard, Alumni Dashboard, Admin Panel, HOD Panel, Job Posting Page, and Mentorship Chat Window.)

Each screenshot demonstrates the functionality of the module — for example:

- Student Dashboard: Displays profile details, job listings, and resume upload section.
- Alumni Dashboard: Shows mentorship requests and chat interface.
- Admin Panel: Displays user and job management tools.

4.4 Explanation of Each Module’s Functionality

- Student Dashboard: Allows students to update their profiles, view eligible job posts, and communicate with mentors.
- Alumni Dashboard: Lets alumni respond to mentorship requests and chat with students for career guidance.
- HOD Dashboard: Displays department data, shortlisted students, and job notifications.
- Admin Dashboard: Used to manage the platform, approve jobs, and monitor all user activities.
- Company Dashboard: Used to post jobs, view applications, and shortlist candidates.
- Super Admin Dashboard: Provides full control with analytics and system-wide visibility.

4.5 Integration of Front-End, Back-End, and Database

The front-end communicates with the back-end using RESTful APIs. When a user performs an action (like logging in, applying for a job, or sending a chat), the request goes to the backend through the API. The Node.js and Express.js server processes the request, interacts with the MongoDB Atlas database, and sends the result back to the front-end.

The integration ensures:

- Smooth data exchange between UI and database.
- Secure authentication and role-based data access.
- Real-time chat and notifications.
- Automatic synchronization of job posts, mentorship activities, and reports.
- This complete integration forms the backbone of the FutureMesh platform, ensuring it runs efficiently and securely for all users.

CHAPTER 05

TESTING

5.1 Objectives of Testing

The primary objective of testing in the FutureMesh project is to ensure that every component of the platform functions reliably, accurately, and consistently before being introduced for real institutional use. Since FutureMesh integrates multiple user roles—students, alumni, company HR, HOD, TNP cell, and admin—testing plays a crucial role in validating that the system remains stable under various conditions. The purpose of testing is not only to detect errors but also to confirm that the platform delivers a smooth and secure experience for all users.

Testing was carried out to validate the behaviour of each module, examine the interaction between different components, and ensure that the system performs well from both functional and performance perspectives. The goal was to ensure that essential features such as authentication, job posting, student shortlisting, mentorship chat, and analytics dashboards operate without any failures. It also ensured that data flows correctly from the user interface to the backend APIs and is stored safely in the database.

The testing activities in FutureMesh specifically aimed to:

- Verify functional correctness of each module, confirming that buttons, forms, dashboards, and workflows behave exactly as designed.
- Ensure smooth integration among connected features such as login → dashboard → job application → tracking → notifications.
- Validate accuracy of user actions, confirming that each input produces the expected result without unexpected behaviour.
- Detect and eliminate bugs early, preventing future issues that could disrupt user experience or platform stability.
- Confirm data consistency, ensuring that information remains accurate when stored, retrieved, or updated across multiple user roles.
- Check platform performance, including loading speed, response time, and behaviour during real-time chat or multiple simultaneous operations.
- Ensure role-based security, verifying that each user has access only to their permitted features and no unauthorized access occurs.

- Enhance usability and user experience, ensuring that dashboards, forms, and workflows are clear, easy to navigate, and intuitive.
- Validate compatibility, confirming that the platform works correctly on different devices, screen sizes, and browsers.
- Check robustness under load, ensuring that the system remains stable when multiple users perform actions simultaneously, such as sending messages or applying for jobs.

Overall, the objective of testing in the FutureMesh project is to ensure that the entire system is functional, secure, stable, efficient, and user-friendly. Testing confirms that FutureMesh meets the expectations of its intended users and is ready for deployment in an actual institutional environment.

5.2 Types of Testing Performed

1. Unit Testing

- Each functional module of FutureMesh was tested separately to ensure that its internal logic behaved exactly as expected. This included checking form validations, API responses, error handling, and UI interactions.
- The login and registration modules were tested for cases such as empty fields, incorrect credentials, duplicate email entries, and invalid formats.
- The mentorship chat module was tested to confirm that messages were delivered instantly, displayed in the correct order, and stored safely in the database.
- The job posting form was checked for correct field validation, eligibility criteria input, and error messages. Unit testing helped identify minor logical issues early, ensuring that each component worked correctly before integration.

2. Integration Testing

After verifying individual modules, their interactions were tested to ensure correct data flow between the frontend, backend APIs, and the database.

- The student application flow was tested by submitting an application from the student dashboard and confirming that the entry appeared correctly in both the recruiter dashboard and the admin panel.

- Role-based navigation was tested to ensure that login credentials directed each user (Student, Alumni, HOD, TNP, Admin, Company) to the appropriate dashboard.
- The job approval workflow was checked end-to-end: Company HR → Admin → HOD → Student. Integration testing ensured that modules communicated without conflicts and that combined operations were stable.

3. System Testing

System testing examined the entire platform as a complete working unit. All functionalities were tested together to confirm that the platform functioned according to requirements.

- Full workflows—such as registration → login → profile update → job apply → tracking → mentorship—were executed without interruptions.
- All dashboards were checked for loading speed, UI consistency, and error-free navigation.
- Stress checks ensured that multiple users could interact (e.g., send messages, apply for jobs) without system crashes.This step verified overall system reliability, performance, and behaviour under real usage.

4. User Acceptance Testing (UAT)

A controlled sample of real users—students, alumni, HODs, and placement staff—were invited to test the platform. They interacted with the actual system and provided genuine feedback.

- Students tested job application flows, profile creation, mentorship request forms, and real-time chat.
- Alumni tested the mentoring dashboard, chat interface, and student connection requests.
- Placement staff tested job approval, shortlisting, and analytics features. Based on their suggestions, improvements were made to button placements, message alignment, form clarity, and notification visibility. UAT ensured that the system was comfortable, intuitive, and suited to real users.

5. Performance Testing

The system was checked for response time, load handling, and smooth operation under multiple active users.

- Chat messages were tested for speed during high activity.

- Dashboards were assessed for loading time when many records were stored.
- Large job lists and student data entries were tested to ensure no slowdowns. This validated that FutureMesh performs efficiently under normal and peak usage.

6. Security Testing

Basic security checks were carried out to ensure that user data remained safe.

- Unauthorized access attempts were tested to confirm that users could not access restricted dashboards.
- Passwords and sensitive data were checked to ensure they were not stored or displayed in plain text.

5.3 Test Cases and Results

Test Case ID	Module	Test Description	Expected Output	Actual Result	Status
TC01	Login	Verify login with valid credentials	Dashboard loads successfully	As expected	Passed
TC02	Registration	Test new user registration	Account created and confirmation shown	As expected	Passed
TC03	Profile Update	Update personal details	Updated information saved	As expected	Passed
TC04	Job Application	Apply for a valid job	Application stored and status updated	As expected	Passed
TC05	Mentorship Chat	Send message to alumni	Message delivered in real-time	As expected	Passed
TC06	Job Posting	Company posts a new job	Job visible after Admin approval	As expected	Passed
TC07	Admin Approval	Approve pending job	Job becomes active	As expected	Passed

Table 5.1 (Source: Testing documentation prepared by project team)

5.4 Bug Reports and Resolution Summary

Bug ID	Description	Reason	Action Taken	Status
B01	Login not redirecting properly	Session not maintained	Fixed by adding session handling in backend	Resolved
B02	Messages not updating in chat	Socket event missing	Added real-time listener and reconnection logic	Resolved
B03	Profile picture not loading	Wrong file path	Corrected upload directory in backend	Resolved
B04	Duplicate job entries in Admin panel	API duplication	Added unique job ID validation	Resolved
B05	Delay in notification updates	Missing async handling	Optimized event handling and API response	Resolved

Table 5.2 (Source: QA report compiled by project team)

5.5 Summary

All the modules of the FutureMesh system were tested carefully to ensure smooth functioning and reliability. Each component—such as user authentication, role-based dashboards, job posting and approval workflows, mentorship chat, application tracking, and analytics—underwent multiple rounds of testing. The results confirmed that all features respond accurately to user actions and maintain consistency across different user roles.

The system was checked for errors, performance issues, and interface problems. Any minor bugs identified during testing were fixed immediately, and the modules were retested to verify the corrections. The testing process also examined how well different modules interacted with each other, ensuring that data moved correctly between the front-end, back-end, and database.

Special attention was given to verifying the stability of real-time features such as notifications and mentorship chat. These features were tested under various user loads to confirm that the system remained responsive without delays.

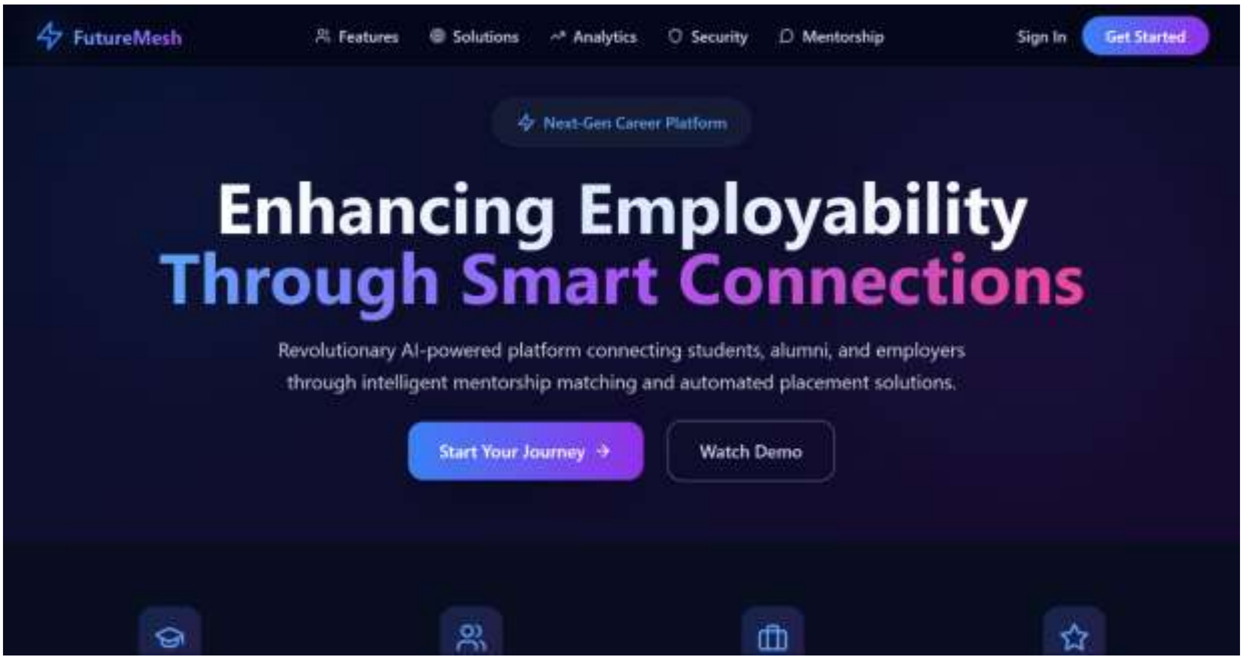
CHAPTER 06

Results and Discussion

6.1 Output Screenshots and Explanation

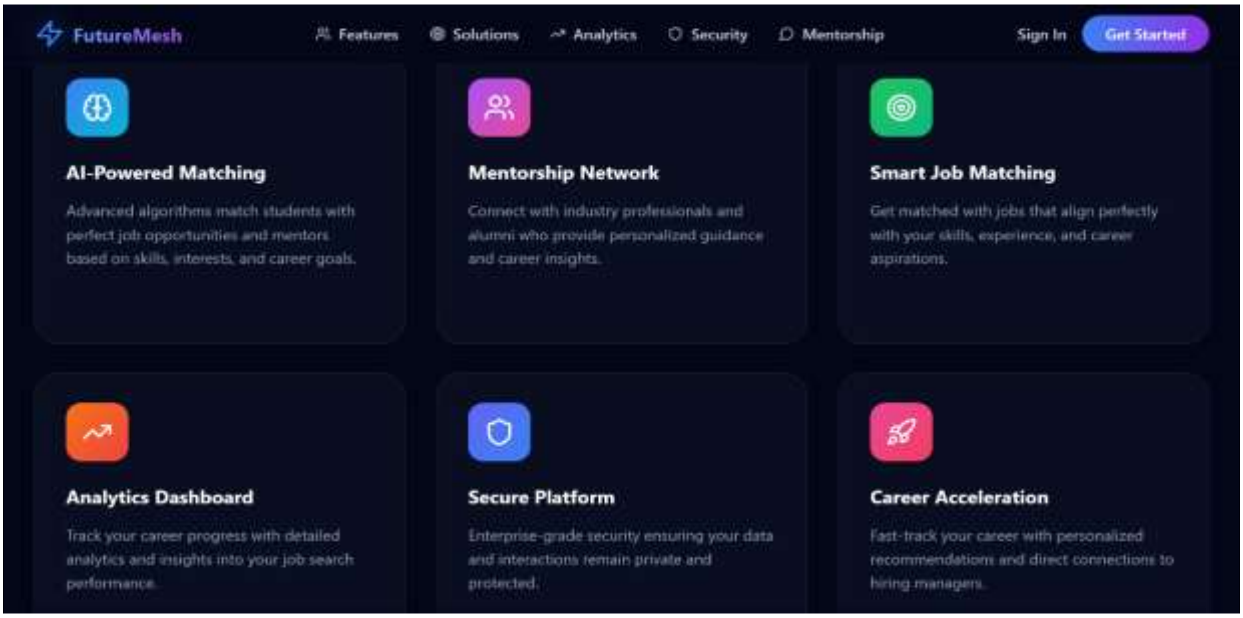
The developed **FutureMesh** system successfully displays the working of all modules through clear and interactive interfaces. Each screen demonstrates the functionality for different user roles.

- **Landing Page:** Shows the main introduction of the platform with login and registration options for Students, Alumni, HODs, Admins, and Company HRs.



Enhancing Employability Through Smart Connections

Revolutionary AI-powered platform connecting students, alumni, and employers through intelligent mentorship matching and automated placement solutions.



AI-Powered Matching

Advanced algorithms match students with perfect job opportunities and mentors based on skills, interests, and career goals.

Mentorship Network

Connect with industry professionals and alumni who provide personalized guidance and career insights.

Smart Job Matching

Get matched with jobs that align perfectly with your skills, experience, and career aspirations.

Analytics Dashboard

Track your career progress with detailed analytics and insights into your job search performance.

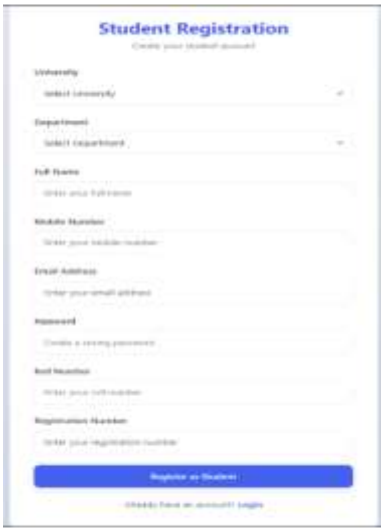
Secure Platform

Enterprise-grade security ensuring your data and interactions remain private and protected.

Career Acceleration

Fast-track your career with personalized recommendations and direct connections to hiring managers.

- **Login and Registration Page:** Allows users to securely log in or create an account based on their role.



Student Registration

Create your student account

University

Select University

Department

Select Department

Full Name

Enter your full name

Mobile Number

Enter your mobile number

Email Address

Enter your email address

Password

Create a strong password

Roll Number

Enter your roll number

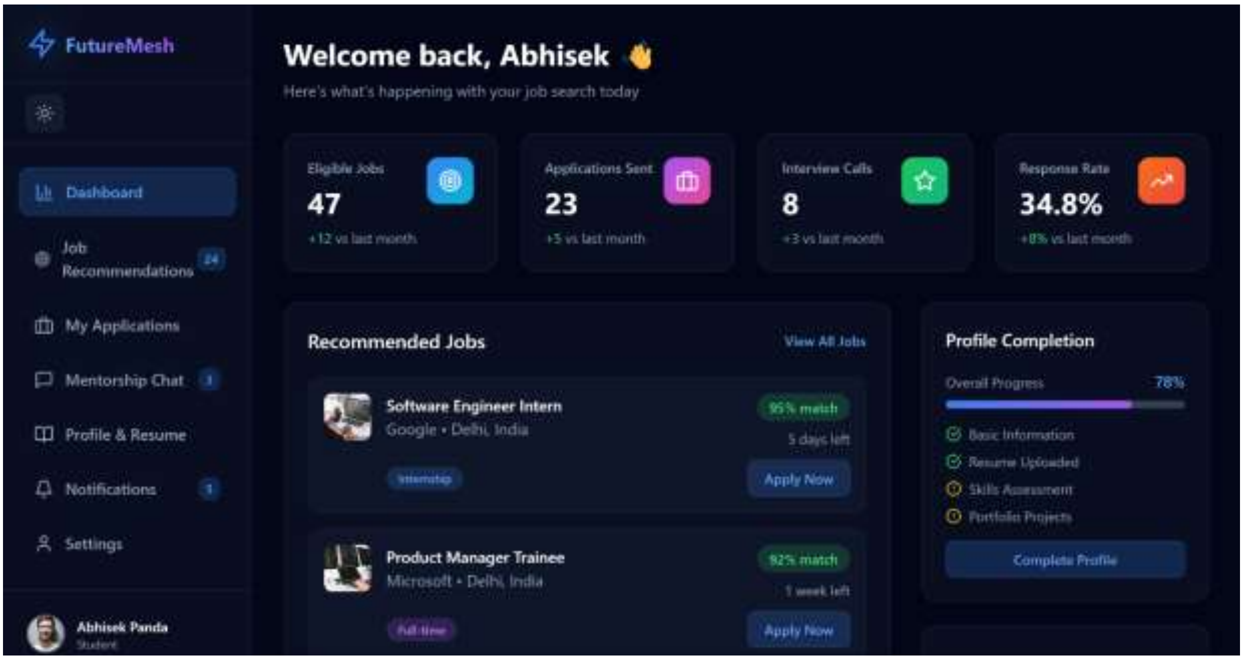
Registration Number

Enter your registration number

Register as Student

Already have an account? Login

- **Student Dashboard:** Displays job listings, profile details, resume upload section, and mentorship chat feature.



Abhisek Panda
Student

Welcome back, Abhisek 👋

Here's what's happening with your job search today

Eligible Jobs
47
+12 vs last month

Applications Sent
23
+5 vs last month

Interview Calls
8
+3 vs last month

Response Rate
34.8%
+8% vs last month

Recommended Jobs

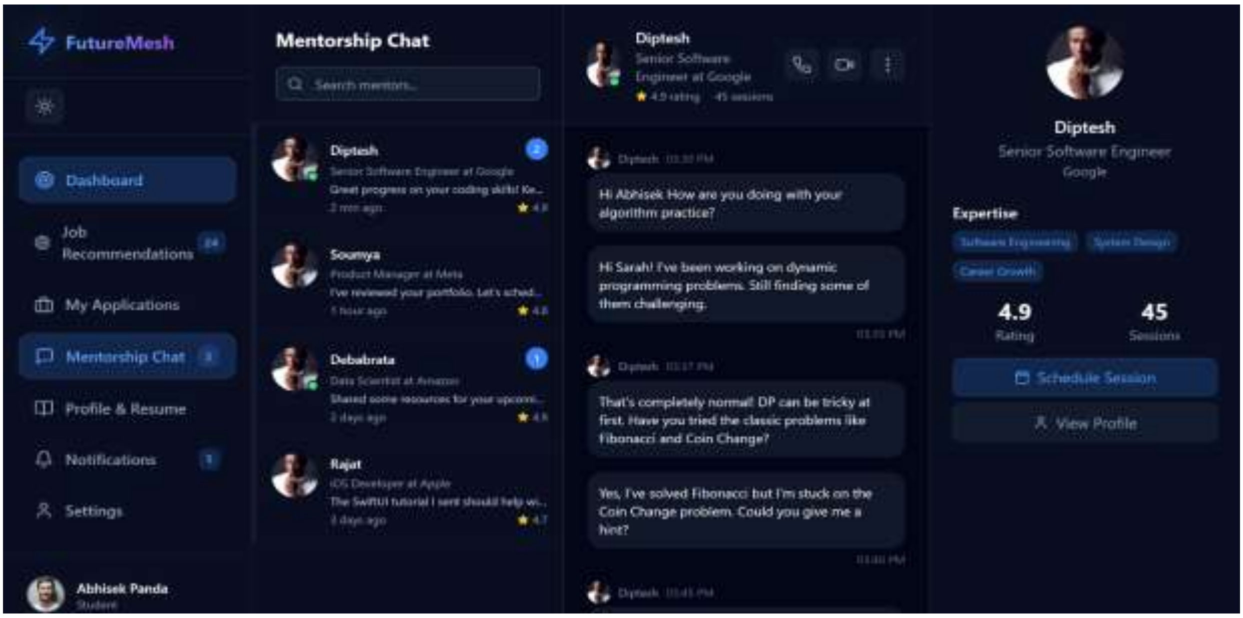
Software Engineer Intern
Google • Delhi, India
95% match
5 days left
Internship

Product Manager Trainee
Microsoft • Delhi, India
92% match
1 week left
Full-time

Profile Completion

Overall Progress 78%

✓ Basic Information
✓ Resume Uploaded
○ Skills Assessment
○ Portfolio Projects



Diptesh
Senior Software Engineer at Google
Great progress on your coding skills! Ke...
2 min ago ★ 4.9

Soumya
Product Manager at Meta
I've reviewed your portfolio. Let's sched...
1 hour ago ★ 4.8

Debabrata
Data Scientist at Amazon
Shared some resources for your upcomi...
2 days ago ★ 4.9

Rajat
iOS Developer at Apple
The SwiftUI tutorial I sent should help wi...
3 days ago ★ 4.7

Diptesh
Senior Software Engineer at Google
★ 4.9 rating 45 sessions

Diptesh 03:30 PM
Hi Abhisek How are you doing with your algorithm practice?

Hi Sarah! I've been working on dynamic programming problems. Still finding some of them challenging.
03:35 PM

Diptesh 03:37 PM
That's completely normal! DP can be tricky at first. Have you tried the classic problems like Fibonacci and Coin Change?

Yes, I've solved Fibonacci but I'm stuck on the Coin Change problem. Could you give me a hint?
03:40 PM

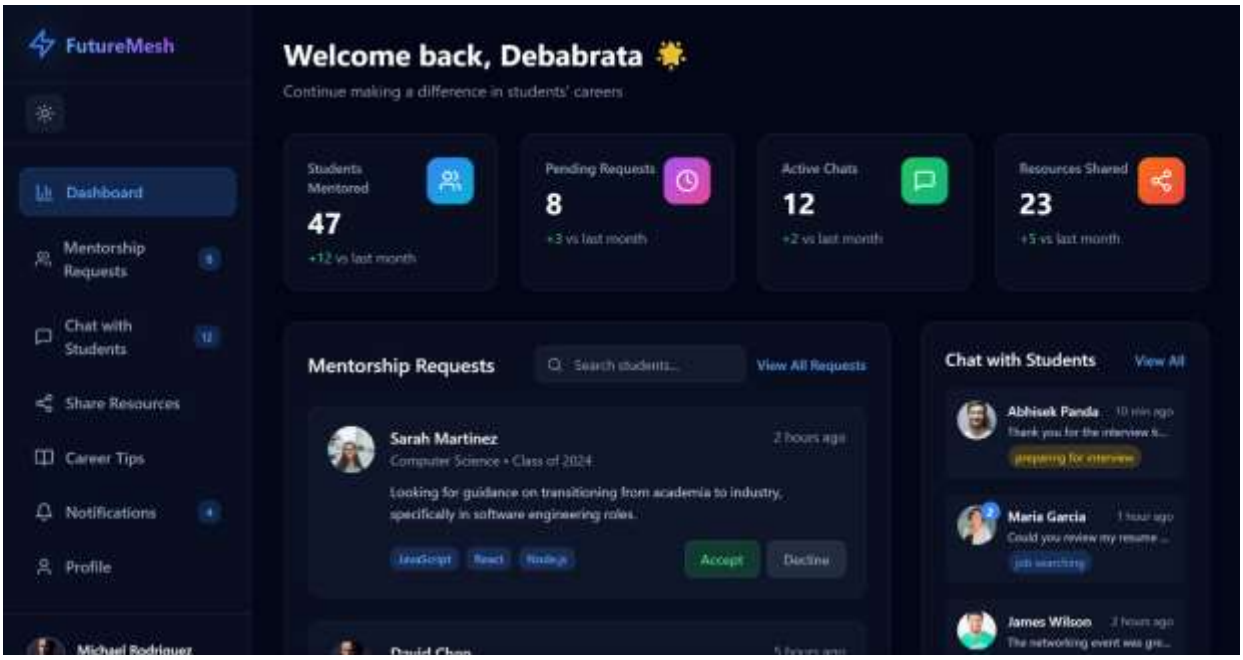
Diptesh 03:45 PM

Diptesh
Senior Software Engineer
Google

Expertise
Software Engineering System Design Career Growth

4.9 Rating
45 Sessions

- **Alumni Dashboard:** Shows mentorship requests received from students and provides a chat window for communication.



Welcome back, Debabrata

Continue making a difference in students' careers

Students Mentored 47 +12 vs last month

Pending Requests 8 +3 vs last month

Active Chats 12 +2 vs last month

Resources Shared 23 +5 vs last month

Mentorship Requests

Sarah Martinez 2 hours ago
Computer Science • Class of 2024
Looking for guidance on transitioning from academia to industry, specifically in software engineering roles.
JavaScript React Node.js

David Chen 5 hours ago

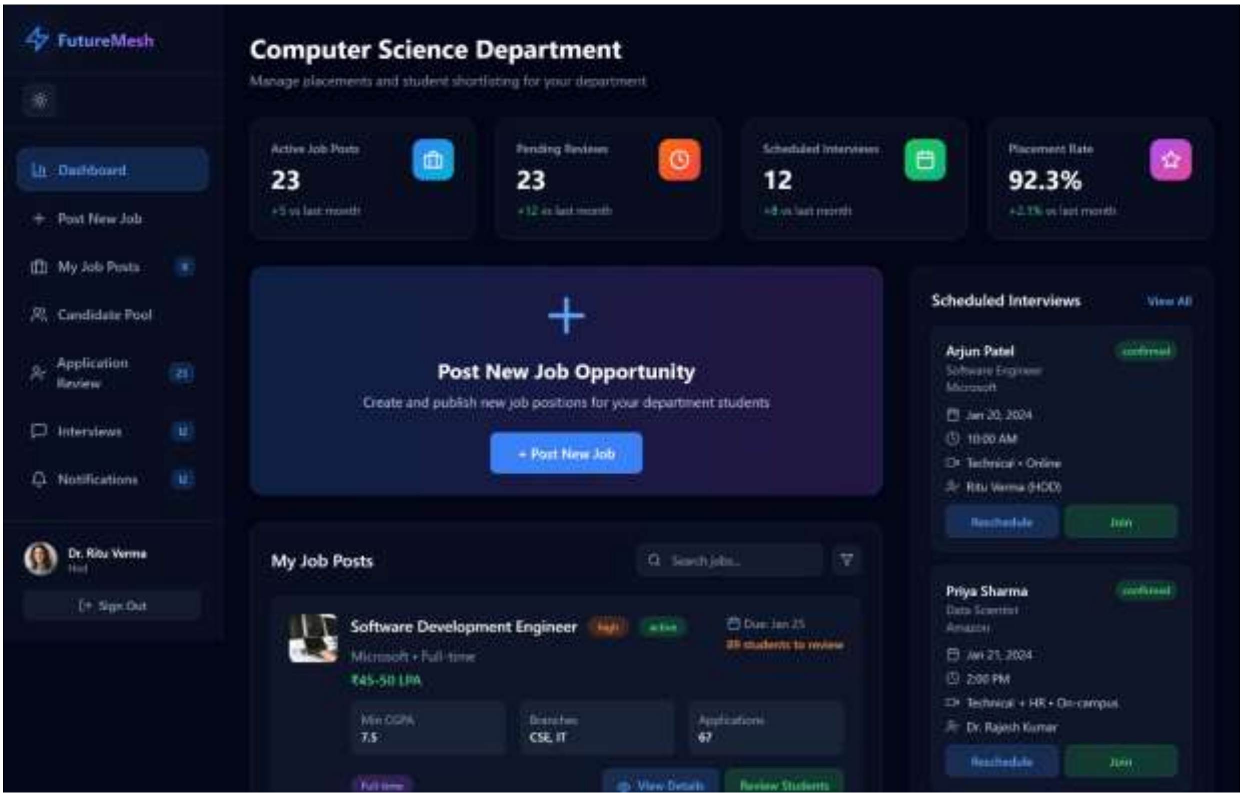
Chat with Students

Abhisek Panda 10 min ago
Thank you for the interview ti...
preparing for interview

Maria Garcia 1 hour ago
Could you review my resume ...
job searching

James Wilson 3 hours ago
The networking event was gre...

- **HOD Dashboard:** Displays department-wise student data, job approval requests, and shortlisted candidates.



Computer Science Department

Manage placements and student shortlisting for your department

Active Job Posts 23 +5 vs last month

Pending Reviews 23 +12 vs last month

Scheduled Interviews 12 +8 vs last month

Placement Rate 92.3% +2.1% vs last month

Post New Job Opportunity

Create and publish new job positions for your department students

My Job Posts

Software Development Engineer high active
Microsoft • Full-time
₹45-50 LPA
Due: Jan 25
89 students to review

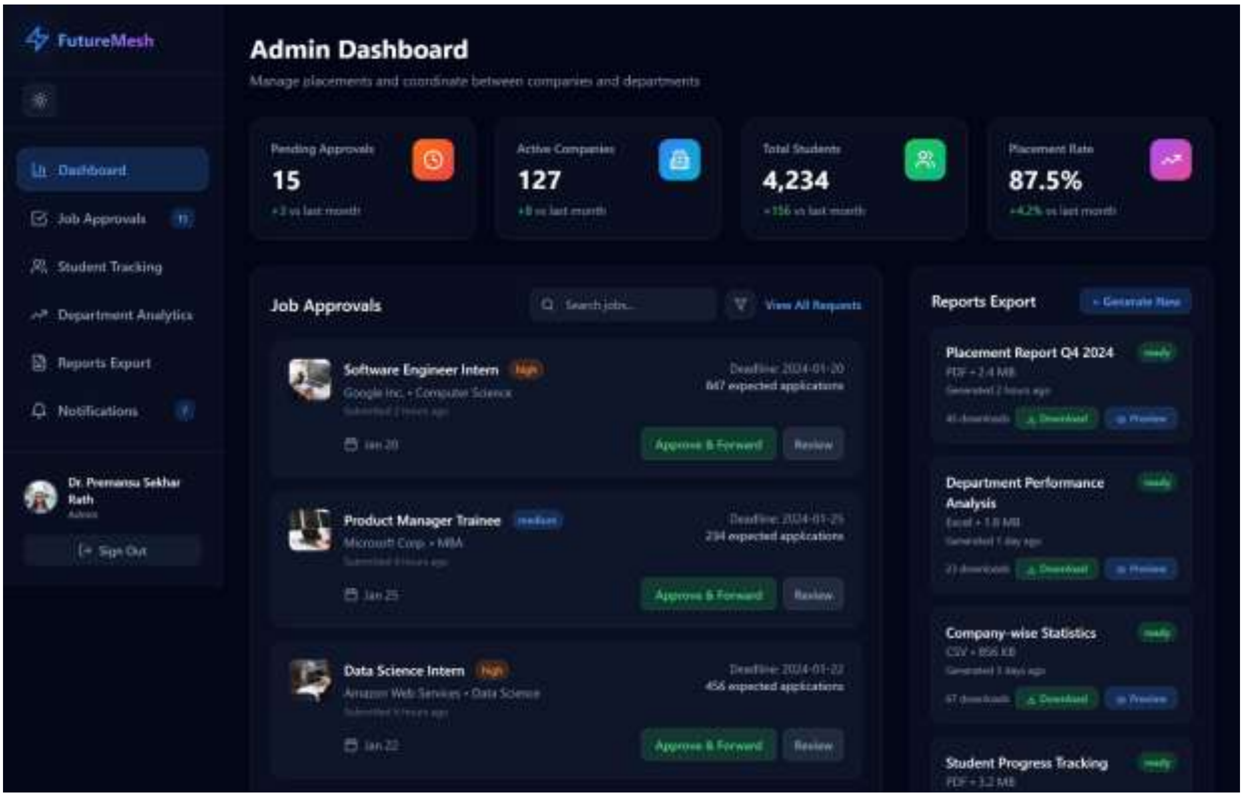
Min CGPA 7.5 | Branches CSE, IT | Applications 67

Scheduled Interviews

Arjun Patel confirmed
Software Engineer
Microsoft
Jan 20, 2024
10:00 AM
Technical • Online
Ritu Verma (HOD)

Priya Sharma confirmed
Data Scientist
Amazon
Jan 21, 2024
2:00 PM
Technical + HR • On-campus
Dr. Rajesh Kumar

- **Admin Dashboard:** Controls user management, job approval, and placement reports.



Admin Dashboard

Manage placements and coordinate between companies and departments

Pending Approvals 15 +3 vs last month

Active Companies 127 +8 vs last month

Total Students 4,234 +156 vs last month

Placement Rate 87.5% +4.2% vs last month

Job Approvals

Software Engineer Intern high
Google Inc. • Computer Science
Submitted 2 hours ago
Jan 20
Deadline: 2024-01-20
847 expected applications

Product Manager Trainee medium
Microsoft Corp. • MBA
Submitted 4 hours ago
Jan 25
Deadline: 2024-01-25
234 expected applications

Data Science Intern high
Amazon Web Services • Data Science
Submitted 6 hours ago
Jan 22
Deadline: 2024-01-22
456 expected applications

Reports Export

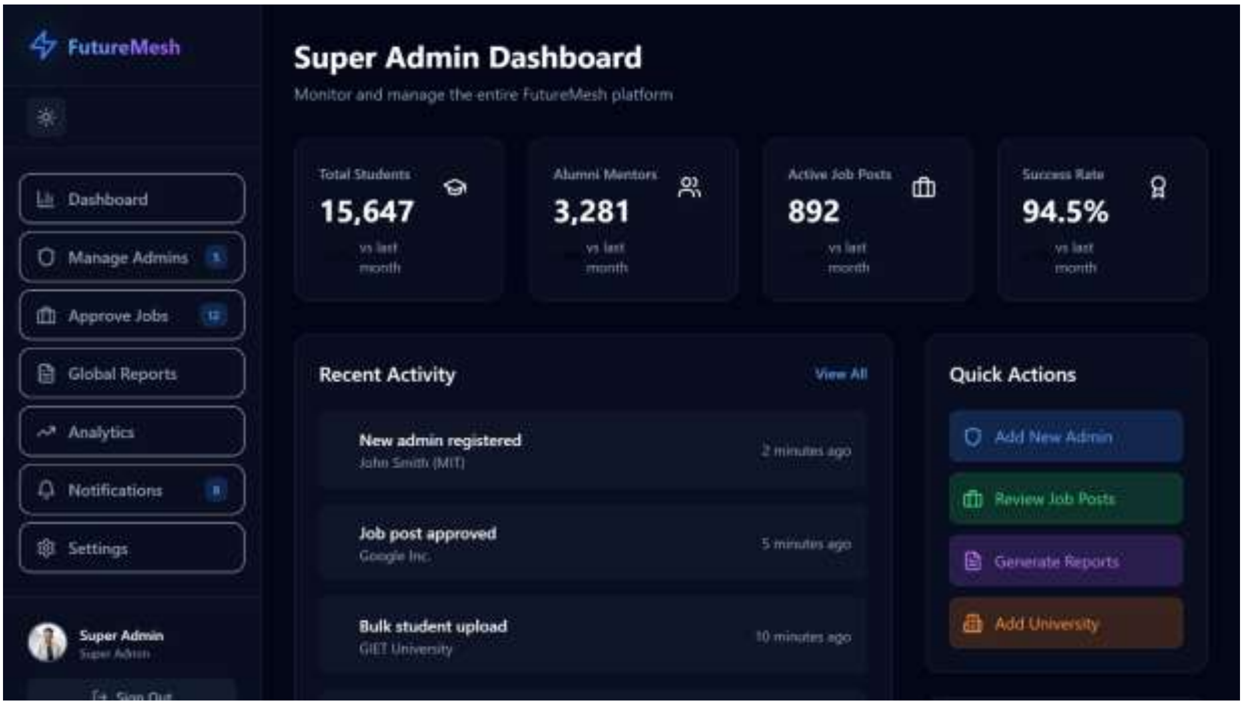
Placement Report Q4 2024 ready
PDF • 2.4 MB
Generated 2 hours ago
45 downloads

Department Performance Analysis ready
Excel • 1.8 MB
Generated 1 day ago
23 downloads

Company-wise Statistics ready
CSV • 856 KB
Generated 3 days ago
67 downloads

Student Progress Tracking ready
PDF • 3.2 MB

- **Super Admin Dashboard:** Gives complete system control with performance statistics and analytics.



Super Admin Dashboard

Monitor and manage the entire FutureMesh platform

Total Students 15,647 vs last month

Alumni Mentors 3,281 vs last month

Active Job Posts 892 vs last month

Success Rate 94.5% vs last month

Recent Activity

New admin registered
John Smith (MIT) 2 minutes ago

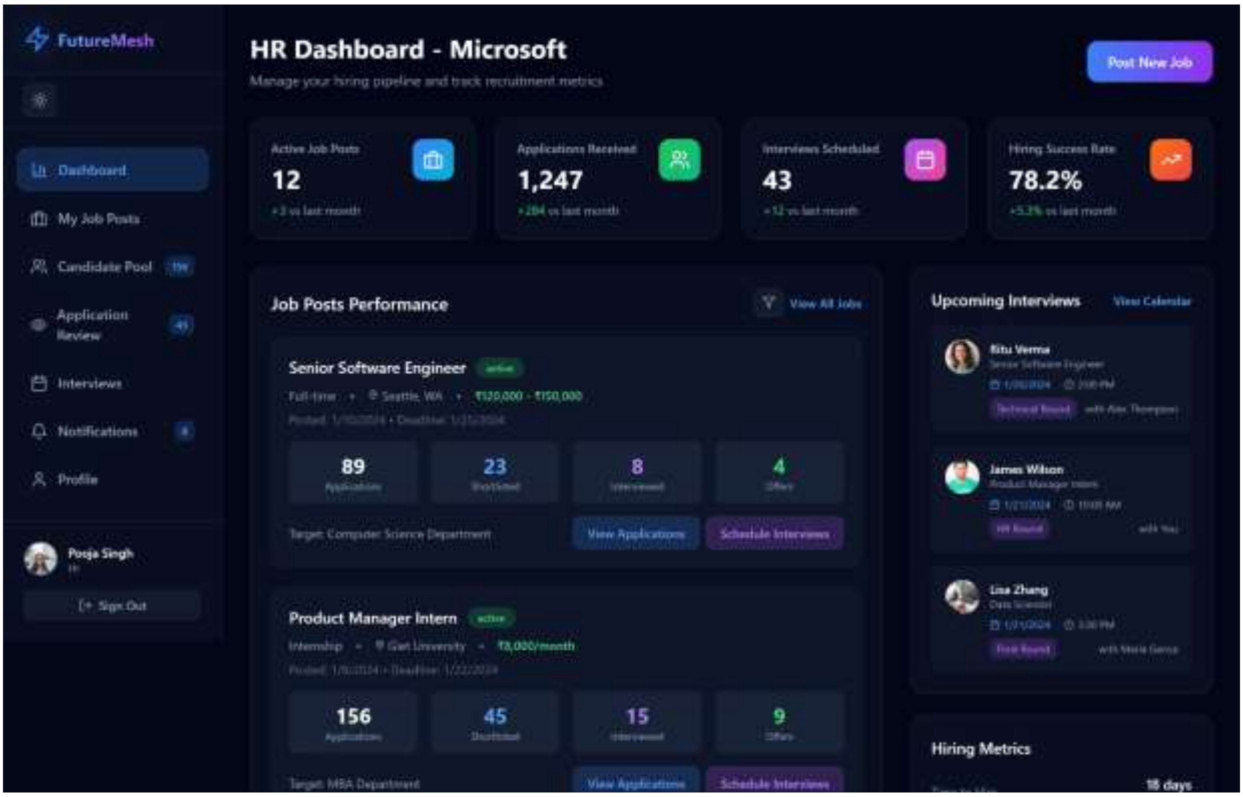
Job post approved
Google Inc. 5 minutes ago

Bulk student upload
GIET University 10 minutes ago

Quick Actions

Super Admin
Super Admin

- **Company HR Dashboard:** Allows posting jobs, setting eligibility criteria, and viewing applications.



HR Dashboard - Microsoft

Manage your hiring pipeline and track recruitment metrics

Active Job Posts 12 +3 vs last month

Applications Received 1,247 +284 vs last month

Interviews Scheduled 43 +12 vs last month

Hiring Success Rate 78.2% +5.3% vs last month

Job Posts Performance

Senior Software Engineer active
Full-time • Seattle, WA • ₹120,000 - ₹150,000
Posted: 1/10/2024 • Deadline: 1/25/2024
89 Applications 23 Shortlisted 8 Interviewed 4 Offers
Target: Computer Science Department

Product Manager Intern active
Internship • Giet University • ₹8,000/month
Posted: 1/8/2024 • Deadline: 1/22/2024
156 Applications 45 Shortlisted 15 Interviewed 9 Offers
Target: MBA Department

Upcoming Interviews

Ritu Verma
Senior Software Engineer
1/20/2024 2:00 PM
Technical Round with Alex Thompson

James Wilson
Product Manager Intern
1/21/2024 10:00 AM
HR Round with You

Lisa Zhang
Data Scientist
1/1[illegible]/2024 3:30 PM
Final Round with Maria Garcia

Hiring Metrics
Time to Hire 18 days

Pooja Singh
HR

Each of these outputs confirms that the platform is responsive, easy to navigate, and visually consistent for all users.

6.2 Comparison with Existing System / Expected Outcomes

In the existing system, most college placement and mentorship activities are managed manually or through basic portals. These systems often lack proper communication, mentorship interaction, and real-time tracking.

Comparison Summary:

Features	**Existing System**	**FutureMesh System**
Job Posting	Manual or limited system	Automated online job posting by HRs
Student Application Tracking	Difficult to manage manually	Real-time tracking with status updates
Mentorship	Not available	Built-in chat between student and alumni
Data Management	Spreadsheet-based	Centralized cloud database (MongoDB Atlas)
Reports & Analytics	Rarely generated	Interactive dashboards and graphs
Role Access	Common for all users	Role-based secured access for each type of user

Expected Outcomes:

- Improved communication between students, alumni, and university staff.
- Faster job posting, approval, and application process.
- Real-time mentorship and guidance for students.
- Easy monitoring of placement trends and performance analytics.

6.3 Performance Metrics or User Feedback

The system was tested by a few users, including students and alumni. The feedback was positive, especially regarding ease of navigation and the chat feature.

User Feedback Summary:

- Students found it easy to apply for jobs and track their applications.
- Alumni appreciated the mentorship request and chat module.
- Admins reported smooth operation of job approvals and data viewing.

Performance Metrics:

- Average page load time: Less than 2 seconds.
- Data retrieval time from the database: Less than 1 second.
- Chat message delivery: Instant (real-time with Socket.io).
- No major crashes or performance issues were observed.

6.4 Analysis of Results

The system achieved its main goals by connecting all stakeholders on one digital platform.

- The automation of job posting and shortlisting reduced manual workload.
- The mentorship module successfully created a link between alumni and students.
- Reports and analytics helped visualize placement data department-wise.
- The role-based dashboards ensured that data remains secure and well organized.

Overall, the FutureMesh platform proved to be efficient, user-friendly, and reliable. It meets the objectives set at the beginning of the project and is ready for further enhancement or deployment at an institutional level.

CHAPTER 07

Conclusion and Future Work

7.1 Summary of Achievements and Contributions

The project FutureMesh Enhancing Employability Through Mentorship and Smart Job Matching successfully achieved its main goals by creating a unified platform that connects students, alumni, training and placement officers, and companies.

Through this system, different users can now perform their tasks efficiently within their own dashboards. Students can register, upload resumes, apply for jobs, and connect with alumni for career guidance. Alumni can act as mentors, share their experiences, and communicate directly with students. The Admin, HOD, and TNP dashboards allow proper management of job postings, student records, and placement analytics.

The platform introduced smart job filtering, real-time mentorship chat, and automated workflows that reduced manual work in the placement process. It also improved communication between institutions and companies by creating a structured and transparent system.

By combining mentorship, automation, and analytics, this project made a meaningful contribution to improving employability and bridging the gap between academic institutions and the job market.

7.2 Limitations of the Current System

While the system performs efficiently, a few limitations still exist in its current version:

1. The platform currently supports only web access; there is no dedicated mobile application.
2. Real-time notifications are limited to in-platform messages and emails; push notifications are not yet included.
3. Chat history between students and alumni is stored but lacks advanced features like file sharing or video communication.
4. The job recommendation system works on basic filters and has not yet integrated advanced AI-based matching.
5. The system has been tested on sample data and requires more large-scale user testing before full deployment.

7.3 Future Enhancement Possibilities

The project can be further improved in several ways to make it more dynamic and impactful:

- AI-Based Job Recommendations: Integrate AI or machine learning to suggest jobs and mentors more accurately based on profile analysis.
- Video Mentorship Feature: Add video calling or virtual meeting options for better mentor–student interaction.
- Resume Analyzer: Introduce an automatic resume evaluation tool to help students improve their profiles.
- Gamification and Reward System: Encourage user engagement by rewarding active mentors and top-performing students.
- Integration with External Job Portals: Sync job data from trusted sources like LinkedIn or Naukri for broader exposure.
- Advanced Analytics: Develop deeper insights using graphical reports to monitor student performance, placement rates, and mentor engagement trends.

REFERENCES

[1] Gupta, R., & Sharma, K. (2023). *AI-powered job recommendation systems for university placement portals.* International Journal of Emerging Trends in Computer Science, 15(2), 45–53.

[2] Patel, M., & Roy, A. (2022). *Enhancing employability through digital mentorship platforms.* Journal of Education and Technology Integration, 10(4), 112–120.

[3] Kumar, V., & Singh, D. (2023). *Role-based dashboards for academic placement management using modern web frameworks.* International Journal of Software Engineering and Development, 18(3), 221–233.

[4] Bansal, P., & Mehta, R. (2021). *Design and implementation of smart campus recruitment systems using MERN stack.* Journal of Advanced Computing Applications, 9(5), 87–96.

[5] Das, A., & Mohanty, S. (2024). *Integration of alumni mentorship and AI-driven job matching in academic institutions.* International Journal of Computational Intelligence and Smart Systems, 11(1), 34–42.

[6] Choudhury, S., & Sahu, P. (2022). *Bridging the gap between students and industries through intelligent placement systems.* Journal of Information Management and Digital Learning, 16(2), 65–74.

[7] Nayak, T., & Jena, R. (2023). *Secure user authentication in multi-role web applications using JWT and Node.js.* International Journal of Cybersecurity and Data Science, 7(4), 176–184.

[8] Verma, R., & Mishra, P. (2024). *Development of real-time mentorship chat applications for higher education institutions.* Journal of Communication Systems and Software Innovation, 5(2), 99–108.

[9] Patra, B., & Sethi, A. (2022). *MongoDB Atlas-based scalable backend architecture for institutional web platforms.* International Journal of Data Engineering, 12(3), 141–149.

[10] Rath, D., & Behera, L. (2024). *Smart integration of student data and analytics dashboards for effective placement management.* Journal of Information Technology Research, 8(1), 53–62

BIOGRAPHICAL SKETCH

1. Abhisek Panda
Roll No: 22CSE072
Registration No: 22UG010159
Department: Computer Science and Engineering
Institution: GIET University, Gunupur, Odisha
Biographical Note:-
Abhisek Panda is an undergraduate student of Computer Science and Engineering at GIET University, Gunupur. His area of interest lies in full-stack web development, modern UI/UX design, and smart system integration. He has developed multiple academic and open-source projects focusing on real-world problem solving through technology. His major project, FutureMesh, aims to enhance employability through mentorship and data-driven job recommendations. Abhisek aspires to pursue a career in software engineering and contribute to innovative digital transformation solutions.

2. Debabrata Mishra
Roll No: 22CSE140
Registration No: 22UG010273
Department: Computer Science and Engineering
Institution: GIET University, Gunupur, Odisha
Biographical Note:-
Debabrata Mishra is a dedicated and analytical student specializing in backend development, API integration, and database management. He has worked extensively on the FutureMesh platform, focusing on database design, user authentication, and secure data flow. His interests include exploring scalable cloud technologies and improving data-driven decision systems. He is committed to continuous learning and aims to become a proficient software developer in the field of modern web technologies.

3. Diptesh Narendra

Roll No: 22CSE224

Registration No: 22UG010357

Department: Computer Science and Engineering

Institution: GIET University, Gunupur, Odisha

Biographical Note:-

Diptesh Narendra is a passionate learner and frontend developer with a keen interest in UI/UX design, responsive web layouts, and performance optimization. In the FutureMesh project, he contributed to building user dashboards and ensuring cross-platform functionality. He has an eye for detail and focuses on delivering seamless and accessible user experiences. His long-term goal is to work in user-centric software development and contribute to creating impactful digital interfaces.

Name: Dr. Gitanjali Mishra

Designation: Associate Professor

Department: Computer Science and Engineering

Institution: Gandhi Institute of Engineering and Technology (GIET) University, Gunupur, Odisha

Biographical Note:-

Dr. Gitanjali Mishra is currently serving as an Associate Professor in the Department of Computer Science and Engineering at GIET University, Gunupur, Odisha. She has several years of academic and research experience in computer science, with expertise in areas such as software engineering, database systems, artificial intelligence, and data analytics.

Throughout her career, Dr. Mishra has guided numerous undergraduate and postgraduate students in their major and minor projects, encouraging innovation and practical problem-solving. She has contributed to academic growth through research publications, seminars, and active participation in curriculum development.

Her mentorship in this project has been instrumental in providing technical direction, ensuring structured development, and maintaining the academic quality of the work. She continues to inspire students to explore emerging technologies and pursue excellence in their professional and research endeavors.

Software_development_Project_7Sem.pdf

GIET

RESEARCH

Gandhi Institute of Engineering and Technology University Odisha, Gunupur

Document Details

Submission ID

trn:oid:::1:3425931941

Submission Date

Nov 27, 2025, 10:34 AM GMT+5:30

Download Date

Nov 27, 2025, 10:41 AM GMT+5:30

File Name

Software_development_Project_7Sem.pdf

File Size

2.6 MB

55 Pages

10,437 Words

64,440 Characters

*% detected as AI

The percentage indicates the combined amount of likely AI-generated text as well as likely AI-generated text that was also likely AI-paraphrased.

Caution: Review required.

It is essential to understand the limitations of AI detection before making decisions about a student's work. We encourage you to learn more about Turnitin's AI detection capabilities before using the tool.

Detection Groups

AI-generated only *%
Likely AI-generated text from a large-language model.

0 AI-generated text that was AI-paraphrased 0%
Likely AI-generated text that was likely revised using an AI-paraphrase tool or word spinner.

Disclaimer
Our AI writing assessment is designed to help educators identify text that might be prepared by a generative AI tool. Our AI writing assessment may not always be accurate (i.e., our AI models may produce either false positive results or false negative results), so it should not be used as the sole basis for adverse actions against a student. It takes further scrutiny and human judgment in conjunction with an organization's application of its specific academic policies to determine whether any academic misconduct has occurred.

Frequently Asked Questions

How should I interpret Turnitin's AI writing percentage and false positives?
The percentage shown in the AI writing report is the amount of qualifying text within the submission that Turnitin's AI writing detection model determines was either likely AI-generated text from a large-language model or likely AI-generated text that was likely revised using an AI paraphrase tool or word spinner.

False positives (incorrectly flagging human-written text as AI-generated) are a possibility in AI models.

AI detection scores under 20%, which we do not surface in new reports, have a higher likelihood of false positives. To reduce the likelihood of misinterpretation, no score or highlights are attributed and are indicated with an asterisk in the report (*%).

The AI writing percentage should not be the sole basis to determine whether misconduct has occurred. The reviewer/instructor should use the percentage as a means to start a formative conversation with their student and/or use it to examine the submitted assignment in accordance with their school's policies.

What does 'qualifying text' mean?
Our model only processes qualifying text in the form of long-form writing. Long-form writing means individual sentences contained in paragraphs that make up a longer piece of written work, such as an essay, a dissertation, or an article, etc. Qualifying text that has been determined to be likely AI-generated will be highlighted in cyan in the submission, and likely AI-generated and then likely AI-paraphrased will be highlighted purple.

Non-qualifying text, such as bullet points, annotated bibliographies, etc., will not be processed and can create disparity between the submission highlights and the percentage shown.

PROJECT_REPORT_Software_development.pdf

by . .

Submission date: 18-Nov-2025 10:17AM (UTC+0530)

Submission ID: 2791112265

File name: PROJECT_REPORT_Software_development.pdf (877K)

Word count: 8129

Character count: 48733

PROJECT_REPORT_Software_development.pdf

ORIGINALITY REPORT

7%	1%	1%	1%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www.coursehero.com Internet Source	<2%
2	Submitted to Victoria University Student Paper	<1%
3	Submitted to Tezpur University Student Paper	<1%
4	www.mdpi.com Internet Source	<1%
5	bcetgsp.ac.in Internet Source	<1%
6	Submitted to Gandhi Institute of Engineering and Technology Student Paper	<1%
7	www.slideshare.net Internet Source	<1%

Exclude quotes	On	Exclude matches	< 10 words
Exclude bibliography	On		