Annexure-E

GANDHI INSTITUTE OF ENGINEERING AND TECHNOLOGY UNIVERSITY, ODISHA, GUNUPUR

Department of Computer Science & Engineering

Statement of Purpose – Major Project-I

1. Title of the Project:

FutureMesh: Enhancing Employability Through Mentorship and Smart Job Matching

Domain/Area of Specialization: Education Technology (EdTech), Career Development, and Placement Automation

2. Abstract:

In today's fast-changing job environment, many students face challenges in finding the right career guidance and job opportunities. Despite having access to alumni, mentors, and university resources, students often feel disconnected from the industry. At the same time, universities and placement departments struggle to manage job postings, shortlist students, and track applications efficiently.

FutureMesh is a complete solution designed to bridge this gap between students, colleges, and companies. Built in line with Smart India Hackathon Problem Statement SIH1630, this platform allows:

- Students to receive personalized job suggestions, mentorship from alumni, and real-time updates for interviews or opportunities.
- Companies to post jobs targeted to specific departments or universities.
- HODs and Admins to track student applications, shortlist candidates, and analyze placement performance using smart analytics.
- Alumni to share career experiences and mentor students directly.

The platform supports role-based login for different users and includes a project showcase system, academic record tracking, resume-building suggestions, and real-time chat. It is built using React, Next.js, MongoDB, HTML/CSS/JavaScript, and integrates tools like APScheduler for scheduled tasks and SMTP for sending emails and updates.

3. Objectives:

- To connect students with verified mentors and industry professionals for career guidance.

- To help universities manage job postings, shortlisting, and placement activities efficiently.
- To provide students with smart job recommendations based on their academic records and skills.
- To allow companies to post targeted job roles and receive filtered applications.
- To offer a real-time chat and collaboration platform for students, mentors, and alumni.

4. Introduction / Problem Statement:

Many students struggle to find the right career path due to limited access to mentorship and lack of visibility into job openings. Universities often face difficulty in coordinating placements and tracking student applications manually. On the other hand, companies look for streamlined ways to hire the right talent from specific departments or institutions. There is a need for a centralized platform that simplifies mentorship, career guidance, and placement for all parties involved — students, universities, alumni, and industry professionals. Future Mesh aims to solve this problem by offering an intelligent, role-based system that brings everything under one digital roof.

5. Expected Input & Expected Output:

Expected Inputs:

- Student profile data (academic records, skills, resume, project details)
- Job postings by companies (role, skills required, eligibility criteria)
- Mentor and alumni registrations
- Departmental data (student batches, university approvals)

Expected Outputs:

- Job recommendations for students based on profile and eligibility
- Application status tracking for students and administrators
- Shortlisting of students by HODs
- Chat and mentorship connections with relevant alumni
- Dashboards showing analytics, placement data, and trends

6. Proposed Methodology / Techniques to be Used:

1. Requirement Understanding and Planning

The first thing we did was understand the needs of all users – students, alumni, HODs, university admins, and company HRs. We listed down what each of them should be able to do on the platform. Based on that, we broke the full project into small tasks. We also created system flow diagrams and a rough structure to understand how each part of the platform will connect with the others.

2. Choosing the Right Technology

We selected tools and technologies that are simple to use but powerful enough to support the features we want. Here's what we are using:

- HTML, CSS, JavaScript for the front-end
- Flask (Python) for the back-end
- MongoDB Atlas for storing all data
- Flask-SocketIO for real-time features like chat
- APScheduler to handle interview reminders and scheduling
- Chart.js or Plotly.js for showing stats and reports
- Vercel for hosting the project

This combination makes our application lightweight, fast, and easy to maintain.

3. Designing the User Interface

We are designing clean and simple dashboards for each type of user. Every role (student, alumni, HOD, HR, admin) will have a separate dashboard with only the features they need. The design is made mobile-friendly, so users can access it on any device. Our main focus is on keeping the layout easy for everyone to understand and use.

4. Role-Based Access System

To make sure that users only access the features they are supposed to, we have added role-based access. This means:

- Students can apply to jobs and chat with alumni.
- HODs can view job posts and select eligible students.
- HRs can post jobs and view student profiles.
- University admins can approve job posts and track placement progress.

5. Job Posting and Student Shortlisting

Companies can post jobs by selecting target universities, departments, and eligibility criteria like skills, CGPA, and graduation year. Once the job is posted, the university admin checks and approves it. After that, the job is sent to the respective department's HOD, who then filters students based on the criteria and sends notifications to them.

6. Student Job Application and Tracking

Students will only see the jobs they are eligible for. They can apply easily and track the status of their application – whether it's under review, shortlisted, rejected, or scheduled for interview. Notifications will be shown inside the platform and sent through emails as well.

7. Real-Time Chat Between Students and Alumni

We are adding a chat feature so students can talk directly to alumni. This can help them get career guidance and tips. Chats will be stored safely in the database. We are also giving smart suggestions so that students can connect with the most relevant alumni based on their department and interests.

8. Student Project and Resume Section

Each student will have a profile where they can upload their academic projects, tech stack used, certificates, and hackathon experience. HRs can see these profiles and make better hiring decisions by looking beyond grades. This also helps students show their practical skills.

9. Dashboard for Analytics and Reports

Admins will have access to dashboards where they can see various stats like:

- Total placements by department
- Most in-demand skills
- Job post trends
- Alumni participation

10. Testing and Fixing Issues

After all modules are completed, we will test the entire platform. We will check each part separately and also how they work together. If any issues or bugs are found, they will be fixed immediately. We will also make sure the site works well on different devices and browsers.

11. Deployment and Live Hosting

Once everything is working fine, we will upload the project to Vercel. We will set up a custom domain and connect all parts — front-end, back-end, and database. Email services and chat functionality will also be activated for the live version.

7. Novelty / Contribution:

It is not just job portal. What makes it different is how it brings together students, alumni, college faculty, and companies all on one platform. Most existing systems only focus on job listings or resume uploads. But FutureMesh goes deeper—it creates a space where students can actually grow, connect, and get real help for their careers.

One of the key features is the mentorship connection. Students can talk directly to alumni who studied the same course or belong to the same department. This allows students to learn from real experiences and prepare better for jobs or interviews.

Another major contribution is the smart job shortlisting system. Instead of manually going through every resume, the platform lets HODs and admins filter students based on CGPA, skills, and projects. This makes the selection process faster and fairer.

FutureMesh also supports a project portfolio section. Students can upload their real academic work and certifications. This gives companies more insights into the student's actual skills, not just their marks. It helps those who are talented but may not have top grades to still get good job opportunities.

The real-time communication system allows smooth and fast chat between different users. Whether it's students chatting with mentors or companies giving interview updates, the chat system keeps everyone informed and connected.

The platform also gives detailed analytics to universities. They can see what kind of jobs are being posted, which departments are doing well, what skills are in demand, and how alumni are contributing. This helps them improve their placement strategies and guide students better.

8. Database Description :

We are using MongoDB Atlas as our main database. This database is hosted on the cloud, which means it is accessible through the internet but is also protected with proper security measures to keep all user data safe.

We have taken steps to make sure that the data stored in our system—like student profiles, job applications, chat messages, and company information—is secure and private. Only the people who are supposed to see certain information will be able to access it. For example:

- A student can only see their own profile and the jobs they applied for.

- A company can only view the applications submitted for their own job posts.
- An HOD can only access the students from their own department.
- Chat messages are private and only visible to the two people involved in the conversation.

We also use authentication and role-based access control, which means each user logs in with a username and password and only gets access to the features and data meant for their role (like student, alumni, HOD, or company).

Additionally, all data exchanged between users and the server is encrypted, which adds another layer of protection. The database also has backup options, so even if something goes wrong, the data can be recovered safely.

We have made sure that no personal data is exposed publicly, and the structure of the database is designed in such a way that information stays organized, protected, and easy to manage.

9. Expected Outcomes / Deliverables:

- A working web-based prototype accessible by students, companies, HODs, admins, and mentors
- Real-time chat system for alumni-student interaction
- Job application tracking and smart recommendations module
- Role-based dashboards for all users
- Analytics reports showing placement stats, skill gaps, and trends

10. References / Tools & Technologies :

- Flask (Python) – We used Flask as our backend framework because it's lightweight and simple to integrate with other components.
- MongoDB – This NoSQL database helped us store and manage student profiles, job postings, and application records efficiently.
- Flask-SocketIO – This tool made it possible to add real-time chat between students and mentors/alumni.
- APScheduler – It was used to schedule automatic interview reminders and notifications through email.
- Chart.js and Plotly.js – These libraries helped us create graphs and dashboards for analytics and visual reports.

- SMTP with Python – We used SMTP (Simple Mail Transfer Protocol) to send important email alerts and job updates to students.
- Vercel – This was used for hosting our frontend securely and quickly.

2. Smart India Hackathon (SIH) Reference

This project idea is based on the Mentor Connect problem statement (Code: SIH1630) from the Smart India Hackathon 2025. We referred to the official SIH problem statement document from the government website while preparing our solution.

3. Technologies and Libraries

- Git and GitHub – For version control and team collaboration.
- HTML, CSS, and JavaScript – Used to build the web interface for all users.
- For future development, we plan to use:
    - spaCy – A Python library for natural language processing. We'll use it to extract useful information from resumes.
    - PyPDF2 – Another Python tool we plan to use to read resume files (PDF) and analyze them.

Future Integration:

- Dialogflow or OpenAI API (Chatbot for mentorship guidance)
- Resume Parser using Python NLP libraries (spaCy, PyPDF2)

Signature of the student **Signature of the supervisor**