MINI PROJECT-I REPORT

(22ITC07)

on

SkillSphere (Skill- Exchange Platform)

Submitted in partial fulfilment for the completion of

BE-IV Semester

in

INFORMATION TECHNOLOGY

By

M. Srujan Nayak (160123737189)
Imtisal Rangrez (160123737180)

Under the guidance of

Dr. E. Ramalakshmi
Assistant Professor,
Dept. of Information Technology

DEPARTMENT OF INFORMATION TECHNOLOGY

CHAITANYA BHARATHI INSTITUTE OF TECHNOLOGY
(Autonomous)
(Affiliated to Osmania University; Accredited by NBA(AICTE) and NAAC(UGC), ISO Certified 9001:2015)
GANDIPET, HYDERABAD – 500 075
Website: www.cbit.ac.in

2024-2025

Introduction

SkillSphere was born from the vision of creating a dynamic, AI-enhanced platform that connects students, mentors, and professionals to foster skill development, networking, and real-time collaboration. Our journey began with the recognition of the challenges faced by learners and mentors in managing connections, tracking progress, and accessing personalized guidance in an increasingly digital world. Over several months of ideation, prototyping, and development, we converged on the concept of a comprehensive ecosystem that seamlessly integrates profile management, mentorship workflows, real-time communication, and AI-driven insights.

Throughout the development process, we focused on building a user-centric platform that not only facilitates networking but also empowers users to grow their skills through meaningful interactions and intelligent assistance. Inspired by the need for personalized mentorship and collaborative learning, SkillSphere offers a rich set of features including secure authentication, detailed profile creation, skill endorsements, virtual rooms for chat and video conferencing, and AI-powered chatbots that provide personalized recommendations and support.

A robust profile and connection management system that enables users to showcase their skills, endorse peers, and build meaningful mentorship relationships. Real-time communication capabilities powered by Socket.IO, allowing users to create and join virtual rooms for seamless chat and video collaboration.

Acknowledgements

First and foremost, we would like to sincerely thank our project guide and faculty members for their unwavering support and mentorship throughout this journey. Their insightful feedback and encouragement pushed us to think critically, refine our ideas, and strive for excellence.

First and foremost, I would like to sincerely thank **Dr. E. Ramalakshmi , Professor of IT**, our guide and mentor throughout this project. You constantly encouraged us to think deeper, be more precise, and keep pushing our limits. Thank you for believing in our idea and consistently guiding us with your valuable inputs.

I'd also like to thank my teammate, **Imtisal Hussain Rangrez**. Imtisal, working with you has been an absolute pleasure. From late-night debugging to design discussions and AI integration struggles, we tackled every step of this journey together. Your focus, clarity, and calm approach under pressure kept the project moving forward.

This was my first real dive into Generative AI, and though it was challenging at first, I gradually became confident with prompt engineering. Learning what prompts worked and what didn't, is something I'll carry forward. It made me realize how powerful AI can be when applied thoughtfully.

Through this project, I not only strengthened my technical skills but also learned the importance of planning, communication, and collaboration. Thank you for giving us the platform to explore and innovate.

Srujan Nayak , 160123737189

Imtisal Rangrez, 160123737180

Literature Survey

The digital collaboration and networking space includes popular platforms like LinkedIn, Slack, and Microsoft Teams. While these tools offer useful features for professional networking and communication, they often lack dedicated mentorship support and personalized AI-driven insights. Most platforms focus on either networking or communication but do not combine both with real-time collaboration and skill development.

LinkedIn provides a broad professional network but does not support real-time chat or AI-based recommendations. Slack and Microsoft Teams offer strong communication tools but are primarily designed for workplace collaboration without mentorship features. These platforms also lack AI-powered personalized guidance.

SkillSphere’s Innovations:

1. **Comprehensive Profile and Connection Management:**
    - Allows users to create detailed profiles showcasing skills, roles, and experiences.
    - Enables sending, accepting, and managing connection requests to build meaningful mentorship and professional networks.

2. **Real-Time Communication and Virtual Rooms:**
    - Supports creation and joining of virtual rooms with password protection and participant limits.
    - Provides real-time chat and video conferencing powered by Socket.IO for seamless collaboration.

3. **AI-Powered Chatbot and Personalized Recommendations:**
    - Integrates OpenAI GPT-3.5 to offer an intelligent chatbot for user assistance and FAQs.
    - Generates personalized skill-based mentor matching and actionable recommendations to support user growth

4. **Responsive and Interactive User Interface:**
    - Implements a modern, responsive design using React, Tailwind CSS, and Framer Motion.
    - Enhances user experience with smooth animations, custom cursors, and toast notifications.

SkillSphere not only connects users but actively supports their skill development and collaboration through a unified, AI-enhanced platform.

System Requirements

ChronoFlow was developed using modern full-stack technologies, emphasizing performance, modularity, and AI integration. The application is based on the **MERN stack** with additional tools for GenAI support and testing.

Prerequisites:

- Node.js (v14 or higher)
- MongoDB (NoSQL database)

Frontend:

- **React.js**: Builds dynamic user interfaces with reusable components.
- **Vite**: Optimizes build and development workflow.
- **Axios**: Manages HTTP requests.
- **Tailwind CSS / Material UI**: Ensures responsive and attractive design.

Backend:

- **Express.js (Node.js)**: Handles routing, authentication, and business logic.
- **Mongoose ORM**: Simplifies MongoDB interaction through schemas and models.
- **JWT**: Ensures secure token-based authentication.
- **Bcrypt**: Safeguards passwords through encryption.

External Tools:

- **Gemini API**: Powers GenAI insights.
- **dotenv**: Manages environment variables securely.
- **Postman**: Used for backend API testing.
- **MongoDB Compass**: Provides a GUI for managing and validating DB operations.

Dev Tools:

- **VS Code**: Primary code editor.
- **GitHub**: Version control and project collaboration.

System Design and Methodology

SkillSphere is built on a modular, full-stack architecture that separates concerns between the frontend, backend, and database layers. The system follows a RESTful client-server model, enabling clear communication via HTTP methods and promoting scalability and maintainability.

The frontend is developed using React.js, allowing for a component-based approach to building dynamic and reusable user interfaces. Styling is managed through Tailwind CSS and Material UI, ensuring a clean, responsive design. Communication with the backend is handled using Axios, which performs asynchronous API calls to fetch and update data in real time.

On the backend, we used Node.js with the Express.js framework to manage routes, business logic, and external integrations. This layer is responsible for handling:

- Password security using bcrypt
- API calls to Google's Gemini API for generating AI-based insights
- CRUD operations for tasks, timetables, feedback, and user profiles

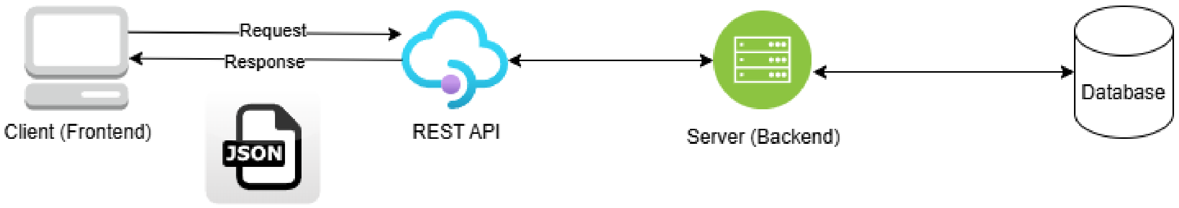


Fig. 4.1 client server architecture using REST APIs

Database Schema

The MongoDB schema was carefully designed to support ChronoFlow's mentor-mentee relationships and AI-driven analytics. Below are the key collections and their structures:

1. Users Collection
Stores users profiles with:

- Credentials (hashed passwords)
- Contact information
- Profile Details

2. Connection Collection
Manages connection requests and statuses:

- Request status (pending, accepted, rejected)
- Assigned mentor reference
- Skill endorsements between users

3. Room Collection
Stores virtual room detail:

- Room ID and password protection
- Participant limits and session durations
- List of current participants

4. Message Collection
Stores messages:

- Sender reference
- Receiver reference

Generative AI Integration

Generative AI lies at the heart of ChronoFlow's innovation. Rather than relying on static data, the system uses **Google's Gemini API** to analyze user behavior, mentor feedback, and task performance to generate weekly insights in natural language.

Key steps in the AI integration process:

1. **Data preparation:** Detailed statistical data like task completions, delays, quadrant distribution, and mentor feedback are collected, formatted and is given to AI.

2. **Prompt engineering:** A structured prompt is created, instructing Gemini to return a JSON-formatted analysis with insights like missed urgent tasks or suggested improvements.
3. **API call and response:** The Gemini API returns structured JSON output, which is parsed, cleaned, and displayed to the user.
4. **Cleaning and returning:** The output received from AI is cleaned and formatted to JSON so that it can be returned to frontend for displaying to the user. Key insights are rendered in the frontend as graphs, charts, and actionable suggestions.

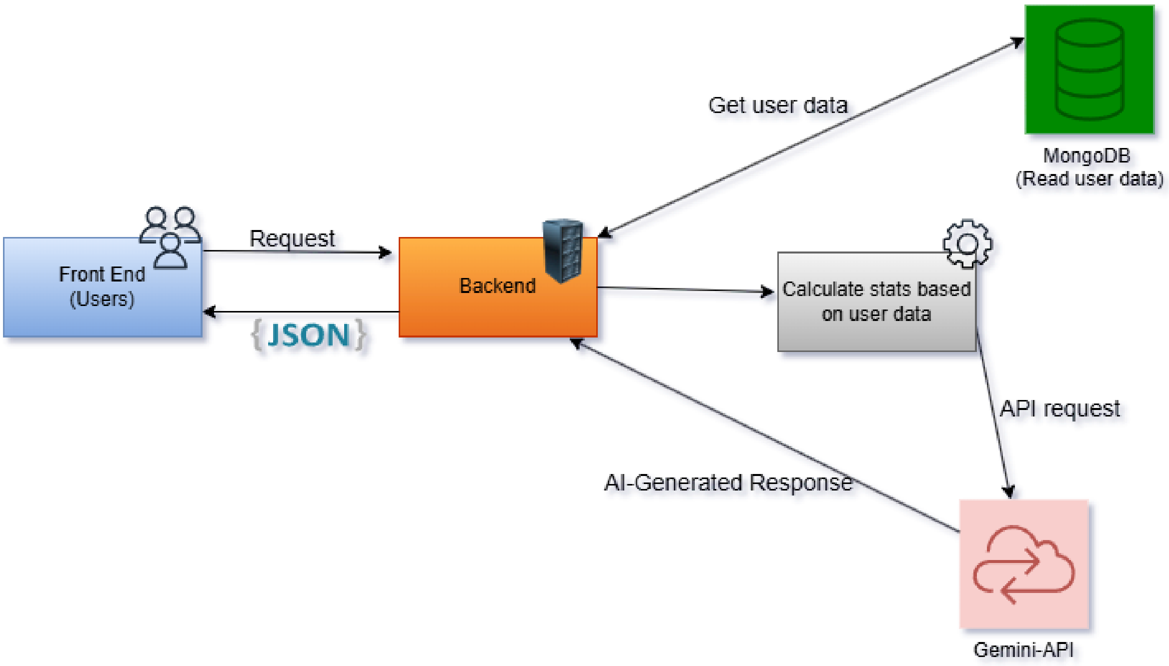


Fig 4.4 Gen-AI working diagram

Benefits of this approach include:

1. Automated performance reviews with minimal manual effort.
2. Personalized suggestions tailored to the user's patterns.
3. Context-aware feedback that improves over time.
4. Support for mentors, who can rely on AI-generated summaries to guide mentees.

Challenges Faced

- **Structured JSON Output Handling:** Ensuring Gemini API consistently returned machine-readable JSON was critical for system integration. Early responses included free-form text or syntax errors, causing frontend failures. We enforced structured outputs through strict prompt templates
- **Prompt Size Limitation:** Gemini's token limit (~15K characters) caused truncation for users with extensive task histories. To resolve this, task data was chunked into smaller batches, prioritized by urgency (Eisenhower Matrix), and summarized pre-API call (e.g., "5 urgent tasks delayed").
- **API Rate Limits in Free Tier:** The Gemini API's free plan imposes strict usage constraints, allowing only 60 requests per minute. This limitation could create significant bottlenecks during periods of high system activity, particularly when multiple users attempted to generate insights simultaneously.

- **Role-based Access Control**: Preventing unauthorized actions (like mentees accessing mentor controls) required both frontend and backend validations.

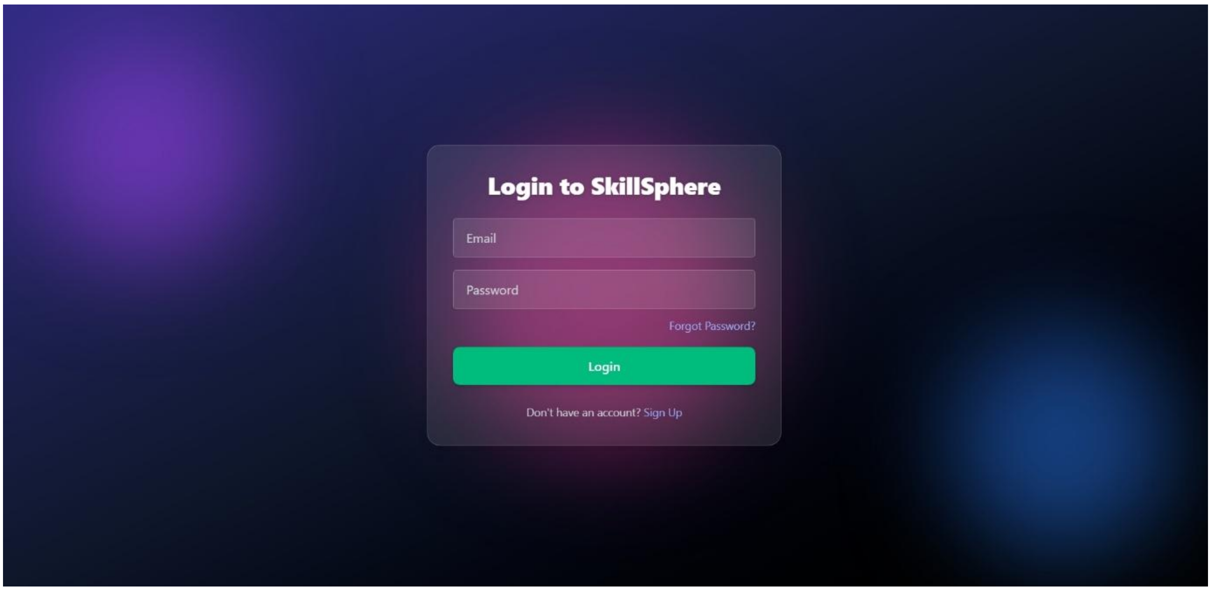
- **Database indexing**: Some MongoDB queries were inefficient, especially with increasing task data.

Implementation Status

The implementation of SkillSphere has progressed significantly, with all major features successfully developed, tested, and integrated into a fully functioning full-stack application. The system is now capable of handling real-time communication, secure user authentication, mentorship workflows, and AI-powered assistance. Both the frontend and backend layers are seamlessly connected through RESTful APIs and Socket.IO, ensuring smooth and responsive user interactions.

Core Features Implemented:

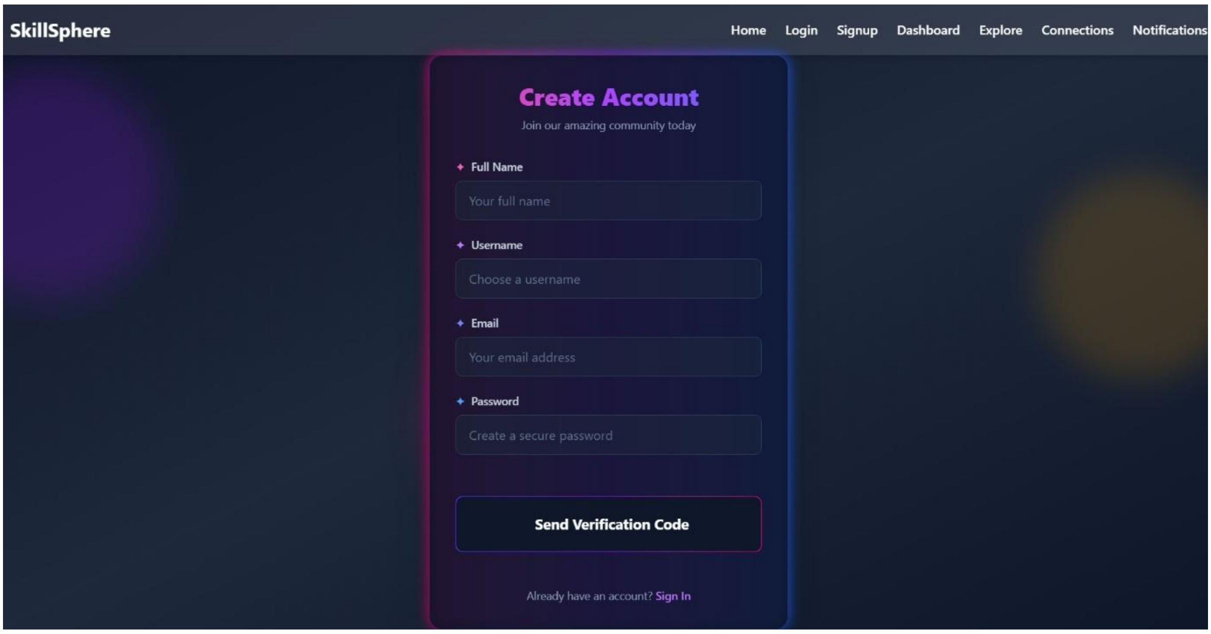
One of the earliest milestones achieved was the **Login and Registration module**. It supports role-based authentication, allowing users to sign in either as a mentor or a mentee. The authentication flow is secured using industry-standard practices such as **bcrypt password hashing** for credential protection and session management. Role-based access control ensures that each user only accesses relevant features, improving both usability and system integrity.



Login to SkillSphere

Don't have an account? Sign Up

Fig 5.1 Login page



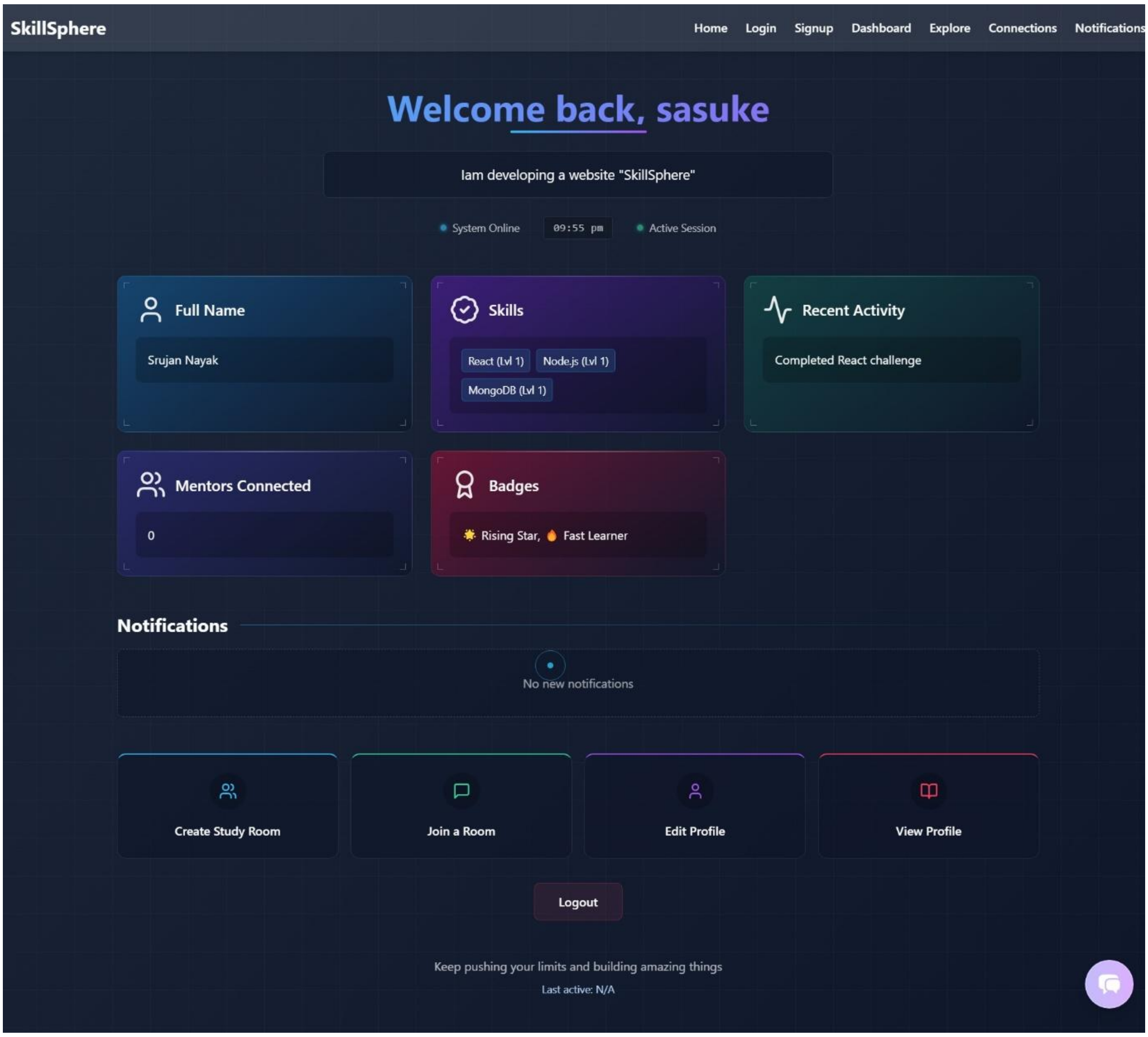
Create Account

Join our amazing community today

Already have an account? Sign In

Fig 5.2 Sign-up page

The **Dashboard** page in SkillSphere has been fully implemented to provide users with a personalized overview of their activities and connections. It displays key statistics such as the number of connections made, sessions attended, and skill endorsements received. The interface supports real-time updates, showing notifications for new connection requests, messages, and other relevant events.



Welcome back, sasuke

Iam developing a website "SkillSphere"

09:55 pm

Full Name: Srujan Nayak

Skills: React (Lvl 1) Node.js (Lvl 1) MongoDB (Lvl 1)

Recent Activity: Completed React challenge

Mentors Connected: 0

Badges: 🌟 Rising Star, 🔥 Fast Learner

Notifications

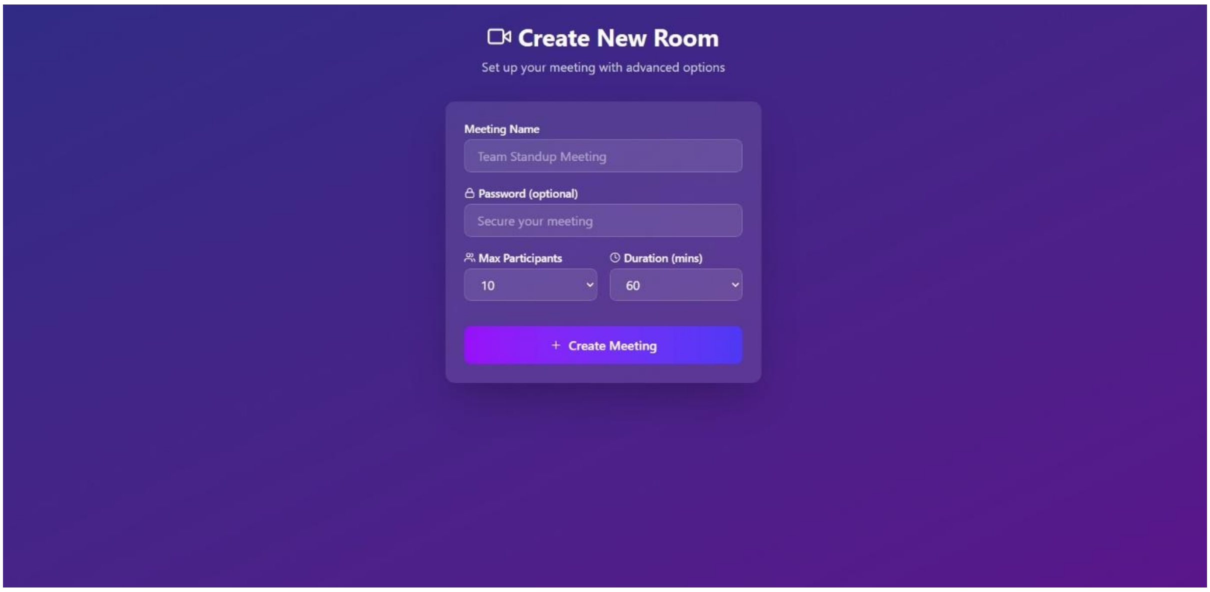
No new notifications

Keep pushing your limits and building amazing things

Last active: N/A

Fig 5.3 Dashboard page

Another key feature, the Explore page, has been developed to enable users to discover and connect with other members of the SkillSphere community. The Explore interface is simple, intuitive, and designed to help users search and filter profiles based on skills, roles, and keywords.



Create New Room

Set up your meeting with advanced options

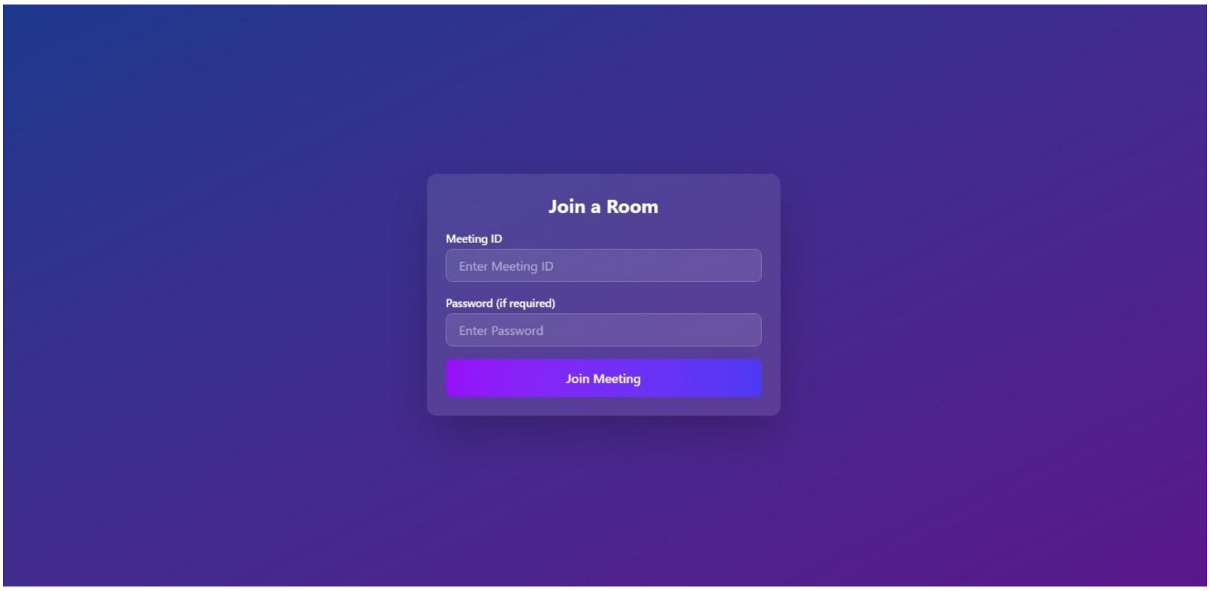
Meeting Name: Team Standup Meeting

Password (optional): Secure your meeting

Max Participants: 10

Duration (mins): 60

Fig 5.4 Create Room page – Creating a new meeting



Join a Room

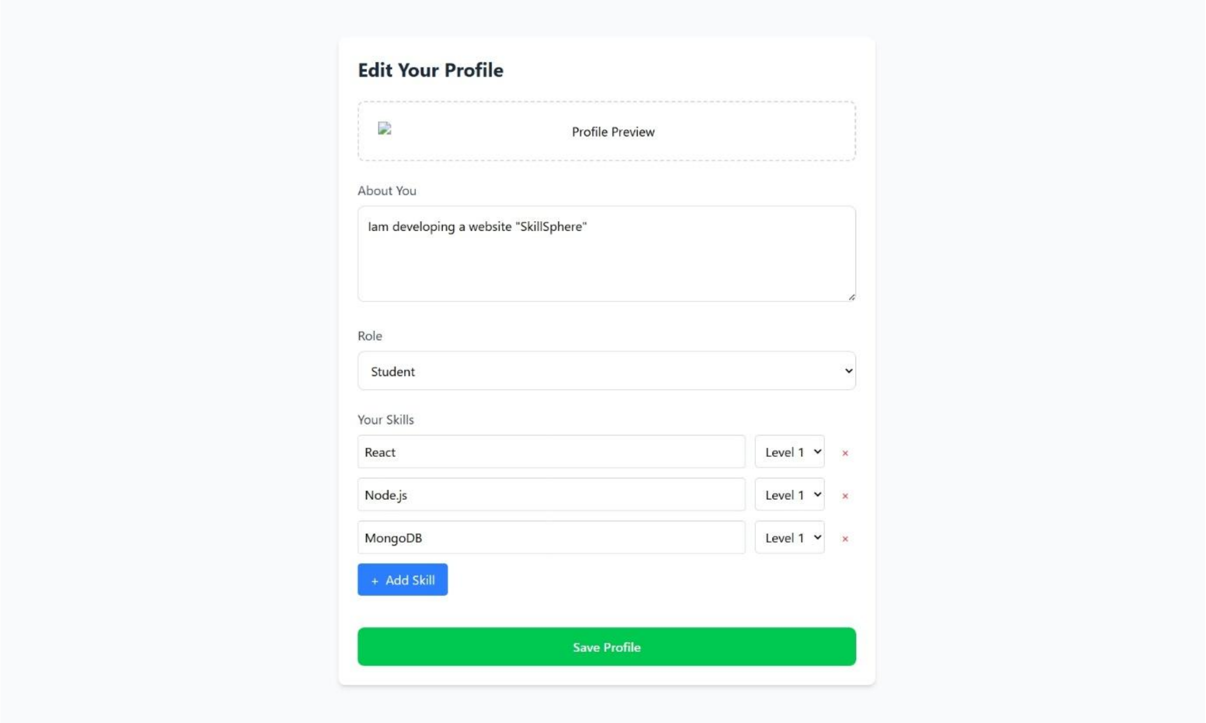
Meeting ID: Enter Meeting ID

Password (if required): Enter Password

Fig 5.5 Join Room page – joining an existing room

The **User Profile section** has been split into two major components:

- An Overview Page, where users can view their basic profile details, skills, connections, and recent activity summaries. This page provides a snapshot of their engagement and progress within the platform.



Edit Your Profile

Profile Preview

About You

Iam developing a website "SkillSphere"

Role

Student

Your Skills

React — Level 1

Node.js — Level 1

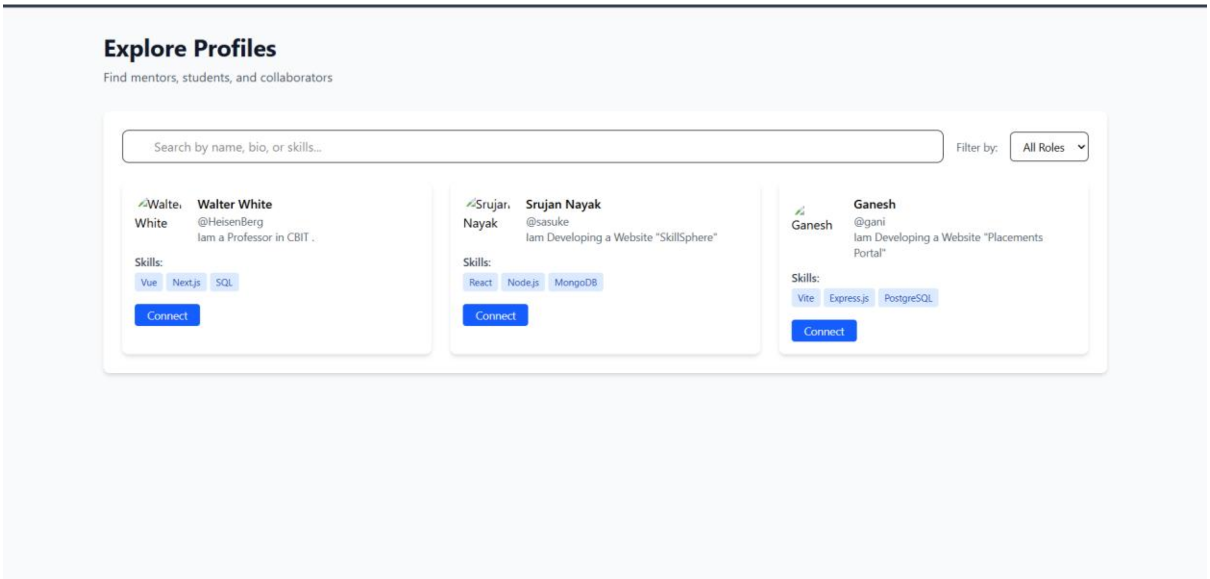
MongoDB — Level 1

Fig 5.6 Profile Page

One of the most significant and innovative parts of the system is the **Generative AI Integration**. Using Google's **Gemini API**, we've enabled the system to analyze user behavior, past performance, feedback, and task data to produce personalized weekly insights. This process involves:

- Compiling the prompt and users data.
- Structuring and sending this data along with a tailored prompt to Gemini.
- Receiving and parsing JSON-formatted output from the API.
- Displaying meaningful feedback and suggestions directly in the UI.

This module was particularly complex but highly rewarding. It transforms ChronoFlow from a standard task tracker into an intelligent productivity assistant.



Explore Profiles

Find mentors, students, and collaborators

Search by name, bio, or skills...

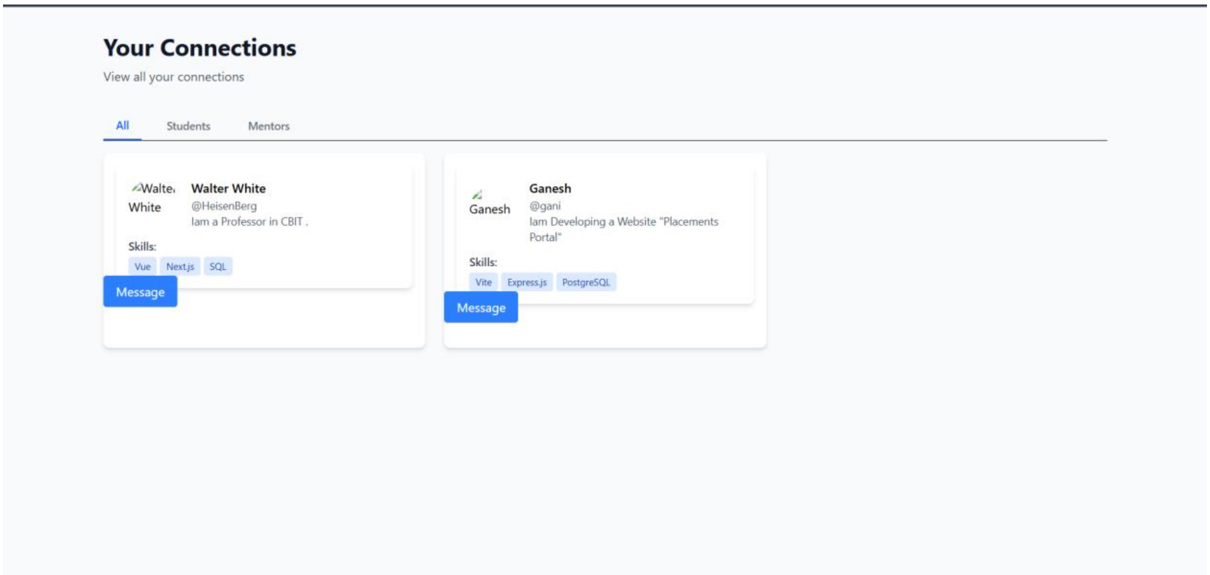
Filter by: All Roles

Walter White
@HeisenBerg
Iam a Professor in CBIT .
Skills: Vue Next.js SQL

Srujan Nayak
@sasuke
Iam Developing a Website "SkillSphere"
Skills: React Node.js MongoDB

Ganesh
@gani
Iam Developing a Website "Placements Portal"
Skills: Vite Express.js PostgreSQL

Fig 5.7 Explore Page



Your Connections

View all your connections

All Students Mentors

Walter White
@HeisenBerg
Iam a Professor in CBIT .
Skills: Vue Next.js SQL

Ganesh
@gani
Iam Developing a Website "Placements Portal"
Skills: Vite Express.js PostgreSQL

Fig 5.8 Connections Page

Future Scope

SkillSphere lays a strong foundation for future growth and innovation. While the current web platform is fully functional and feature-rich, several enhancements could further elevate the user experience and expand its capabilities:

- **Mobile Application**: A React Native-based version would allow users to manage tasks and view insights on the go, increasing accessibility and engagement.
- **Local AI Integration**: By using tools like Ollama, insights could be generated offline, reducing API latency, cost, and dependency.
- **Advanced AI Models**: Shifting to premium models like GPT-4 Turbo or Claude could offer deeper, more nuanced insights and improved language generation.
- **Organization-wide dashboards**: A scalable admin panel for schools or companies to monitor multiple mentees, view collective productivity trends, and assign mentors at scale.
- **Interactive Visualization**: Use of D3.js or Chart.js to provide better insight into time management trends through animated, filterable visual analytics.
- **Multilingual Support**: Supporting local and international languages to make the platform inclusive for non-English speakers.

These directions ensure that ChronoFlow evolves into a comprehensive productivity and mentorship ecosystem.

Conclusion

SkillSphere was more than just a technical project; it was a journey to create a meaningful platform that connects students, mentors, and professionals to foster skill development and collaboration. In building SkillSphere, we aimed to explore how real-time communication, AI-driven insights, and secure mentorship workflows could be combined to empower users in their personal and professional growth.

What sets SkillSphere apart is its integration of real-time virtual rooms, AI-powered chatbots, and personalized recommendations within a unified platform. This approach goes beyond traditional networking tools by offering users actionable insights, seamless collaboration, and structured mentorship opportunities.

On a personal level, this project was a major milestone in our development as software engineers. We got hands-on experience with the MERN stack, secure authentication, prompt engineering, and AI integration through Google’s Gemini API. More importantly, we learned how to think from a user’s perspective and how to turn technical concepts into meaningful, user-friendly features.

SkillSphere has a solid foundation, but its journey is just beginning. With plans for mobile app development, enhanced AI capabilities, and broader institutional adoption, this project has the potential to evolve into a comprehensive skill development and mentorship ecosystem.

Key Takeaways

1. **Full-Stack Development with MERN Stack**: Working on ChronoFlow gave me a comprehensive understanding of the MERN stack. We built the frontend using React.js, styling the interface with Tailwind CSS and Material UI for a clean, responsive design. We developed RESTful APIs using Node.js and Express.js, which allowed seamless communication between the frontend and backend. Data modeling and database interactions were handled using MongoDB with Mongoose ORM, enabling effective schema design and data management. We also implemented secure authentication mechanisms using JWT and bcrypt, which helped me understand how real-world applications handle user verification and protect sensitive information.

2. **Prompt Engineering**: This was my first real experience working with Generative AI in a practical project. Initially, getting structured responses from Gemini was a challenge, free-form outputs often caused frontend failures. Through trial and error, I learned how to design clear, consistent, and role-specific prompts that returned predictable, machine-readable JSON output.
   I realized how small changes in the prompt (like tone, instructions, or data format hints) could drastically affect the quality and structure of the AI's response. This process helped me deeply understand the concept of prompt engineering, one of the most crucial skills in today's AI-driven development space.

3. **Professional Networking:** Documented our development journey on LinkedIn, sharing milestone updates and lessons learned. This improved technical communication skills and helped build a public portfolio.