Smart Collaboration and Career Guidance System

NAMBURI SRINVASA RAO*[1], MAREPALLI HARYA VARDHAN[2], VASIMALLA SUDHEER[3], NELATURI SURESH REDDY[4]

[1]Assistant Professor, Department of CSE(AIML), Bapatla Engineering College, Bapatla 522101, AP,India
[2]Student, Department of CSE(AIML),Bapatla Engineering College, Bapatla 522101,AP, India
[3]Student, Department of CSE(AIML), Bapatla Engineering College, Bapatla 522101, AP, India
[4]Student, Department of CSE(AIML), Bapatla Engineering College, Bapatla 522101, AP, India

Abstract— Out there among learners, confusion often clouds what comes after school ends. Not every graduate sees where they might fit within today's working world. A new kind of tool steps in here - one that learns who you are through your history, passions, strengths, and goals. Once it knows those pieces, pathways start appearing: jobs that match, abilities worth building, materials to study. Hidden patterns rise into view when data meets purpose. Choices become clearer because the noise fades around them. Besides helping with career choices, the platform connects learners, advisors, and experts using shared spaces for talks and team projects. Ideas flow freely here, where questions meet real-world answers through active participation. Learning resources appear alongside tailored course suggestions, building abilities that matter in today's work settings. With smart tools guiding personal growth, advice from seasoned mentors, and group-based skill sharing, theory meets practice more easily. Outcomes show clearer paths forward - stronger readiness, sharper skills, better chances when applying for jobs.

Key Words— **Career Guidance, Skill Recommendation, Student Collaboration, Artificial Intelligence, Mentorship Platform.**

II. INTRODUCTION

Figuring out what comes next after school matters a lot for students and those just starting work. Some finish classes but still feel unsure about where they fit best. Without someone pointing the way, picking a job can miss the mark when it does not line up with what they enjoy or how they think.

Out here in the real world, tech shifts fast - companies need workers who know exactly what to do. Yet most learners have no clear idea what those jobs actually ask for. Because of that, school lessons often miss the mark when it comes to workplace demands.

Without support from seasoned individuals, learners might struggle to see clear paths forward. It begins when there is little access to teamwork or trusted advisors. Guidance shapes how they explore professions, notice shifts in fields, or build abilities step by step. Someone who has walked the road before can make choices feel less uncertain.

Starting off, the tool helps learners tackle tough choices through smart support. A unique space opens up when pupils explore jobs that fit their strengths. One way it works is by checking what subjects someone studied. Another part looks at hobbies plus talents to suggest next steps. Pupils talk with classmates while building links to experienced guides. This setup shifts how advice feels - closer, clearer. Paths appear based on real data about each person. Sometimes a match clicks right away. Behind scenes, patterns shape suggestions without extra effort. Getting help becomes natural, almost like chatting.

Another benefit is access to tools like study guides, suggested classes, together with chances to work on team projects. Because of this setup, learners build abilities while getting ready for jobs down the road.

One way to look at it is how the platform mixes advice on jobs with shared workspaces online. What stands out is its smart suggestions shaped by artificial intelligence. Students start seeing clearer paths forward when these pieces come together. Instead of guessing, they build real abilities needed today. The outcome often means choices that fit their goals more closely.

III. LITERATURE SURVEY

Career guidance tools use educational history, personal preference and skills to recommend careers. Most of these systems recommend careers only which does not also offer collaborative learning environment and mentorship support for students. [1].

The researchers proposed the web-based portals which analyze students' academic records and recommend possible career options. The examination results and qualification are assessed by these system to make students get into a job. Even though such systems might be helpful for the students to identify possible career option but it does not include interaction features. For example, interaction features like communicating with mentor or collaborating with other students. Consequently, students are unable to converse with experts about careers. [2].

Machine learning algorithms have been deployed by researchers to develop intelligent career recommendation systems that analyze user profiles, technical skills and interests. Based on the analyzed data, these systems offer personalized recommendations for job roles. Using machine learning techniques has the potential to improve the accuracy of choice of career suggestion and help students to explore their choice of career. Though, such systems usually do not contain real-time industry insights or collaborative platform users can share knowledge and experience. [3].

Numerous platforms have been created where students can interact and network with the industry to offer mentorship opportunities. Together with a panel of mentors and leaders from the industry, students get to learn. These platforms may provide useful professional advice, but they mainly consist of manual interactions and don't offer you automated career recommendation through data analysis. [4].

To encourage skill development in students, various online learning platforms recommend courses and training based on user interests. These platforms allow learners to enhance their knowledge and strengthen their technical skills which are in demand in industries these days. Yet, such mechanisms focus primarily on learning and training and do not offer personalized career planning or integrated guidance mechanisms linking learning and career opportunities. [5].

In order to improve the career guidance systems effective integration of recommendation systems based on Artificial Intelligence to the collaborative platforms is found useful. Systems that use artificial intelligence can analyze huge amounts of user data and give career recommendations. Such systems can recognize the best opportunities according to user profiles for taking decisions. Nonetheless, a unified system that integrates career recommendation, mentorship, skill evaluation, and collaboration features is still absent in many other existing models. [6].

IV. EXISTING SYSTEM

At present, the majority of existing platforms cater to just one aspect of student or career development, whether they be job recommendation systems, online learning portals, mentorship platforms or skill assessment tools. In one place, students don't get the chance to receive comprehensive career guidance, as generally, the systems don't work in tandem. Several platforms reportedly provide career suggestions based on only academic marks or some questionnaire answer. They do not offer suggestions based on student's interest, year strength, learning behaviour, and live performance. Also, many existing systems do not support collaborative learning features like team formation, peer interaction, and project-based guidance which are now in skill-based days. Consequently, students tend to be confused about choosing appropriate career options, lack mentorship and miss exposure opportunities. The necessity of one smart platform that converges career guidance, skill assessment, collaboration and personalized recommendation so that decisions made by students are not only informed but also well thought of.

V. PROPOSED SYSTEM

A contemporary all-inclusive platform was created to encompass a career guidance and collaborative learning app for students. The system starts with the registration of users and the creation of the profile in which a learner will provide the necessary details related to their personal information, educational background, skills, interest, career choice, technical knowledge, etc. The platform performs student profiling as per this data which helps to understand the strength of the child. The technical and problem-solving skills assessment module will help us understand where the candidate's strengths lie and where they are lacking.

The system will study their data collected and assessment results using various Artificial Intelligence techniques to recommend them the right career option like Software Developer, Data Scientist, Artificial Intelligence, Cyber Security, and Cloud Computing, etc. The platform also provides online courses, tutorials, certifications, and project ideas to further improve your skills in each suggested career option. In addition, the inclusion of collaborative elements like discussion forums and group projects alongside interaction with mentors promotes peer-to-peer learning and real-world engagement. The platform goes above and beyond to ensure that there are mechanisms for tracking the career progress of the learners by monitoring the improvement over the years and redirecting guidance dynamically.

VI. SYSTEM ARCHITECTURE

1. Client Layer (Browser / User Interface)

Users such as students and mentors access the system through a web browser running a Single Page Application (SPA) built with React and Vite. The platform provides both student (those who consume mentoring services and participate in collaborative study rooms), and mentor (individuals who deliver mentored sessions, where they have the opportunity to be paid for their service). It is interactive and responsive, to ease the user experience in navigating through various

2. Authentication and Session Management

Users log in or register which provides the system with credentials for authenticating users. The frontend makes authentication requests via REST endpoints to the Flask backend (/api/auth/login, /api/auth/register etc). User credentials are validated on the backend, before they are securely hashed with bcrypt and a JSON Web Token (JWT) is generated with a validity of 24 hours. The token gets stored in react context api (AuthContext) and this is then appended to all future API calls as a Bearer token to provide security.

3. Frontend Routing and Dashboard

After authenticating, users are redirected to a protected dashboard depending on their role. The dashboard follows a modular component-based structure that implements every function tab as an independent react component. These components consist of an overview dashboard with statistics, discovery and management of mentors that users can connect to, creating and joining study rooms, communicating with AI-based career bot, discovering courses and managing profiles. With this modular architecture, scalability and maintainability of the frontend application is improved.

4. API Layer (Flask Blueprints)

This separated the back-end into a series of RESTful API structures organized around Flask Blueprints. Types of responsibility are divided into different blueprints, for example user authentication, management of profiles, operations within a room, mentor connections, chatbot services, and payment processing itself. Also, for the React front end and Flask back end, there needs to be cross-domain access to resources at each end connected by AJAX calls. This is achieved using Flask-CORS. Use of Flask-JWT-Extended enforces secure access to protected endpoints, and verifies the JWT included in each API request.

5. Control Layer (Controlling flow layer)

Any routing adapter spins off the business logic handling to a separate controller-style module. One carries the user controller be accustomed to addendum knowledgeable consultants, profile collects and connects to Razorpay Account. Another-one of these,'room controller' is in charge for creating study rooms or approving their members to collaborating which tasks/messages are held within it. The connection controller looks after mentor requests, including sending, agreeing with or declining requests from one state into another. The payment controller is responsible for creating Razorpay purchase orders, verifying HMAC signatures, and routing money to mentors' accounts through a URL-based payout system.

6. Database Layer (SQLAlchemy ORM)

The system uses SQLAlchemy ORM in order to map Python classes relational databases. Through this method, data management is made much more efficient. The core data model is made up of user information, study room meta data, friends, collaborative tasks, chat messages, mentor connection requests and payment transactions. PostgreSQL is used in production for reasons of reliability and scalability. Meanwhile, SQLite can also be used as a fallback development database. The configuration of databases is dynamically controlled through such environment variables as DATABASE_URL.

7. Automated Career Assistant Service.

A chatbot providing career assistance powered by AI is available on the dashboard. When the user sends a query, it is sent to the backend endpoint /api/chat/message. This calls the service layer module that generates a response. This module takes knowledge from a local career guidance document repository and combines it with output stream responses from the Grok AI API of xAI. The merger makes students get personalized and contextual career recommendations.

8. Razorpay Integration Payment Flow

Students can choose mentors and start scheduling payment through a secure payment workflow. The frontend will make a request to the backend to create an order. The backend, in turn, will send a request to Razorpay to create a payment order. The payment widget is displayed on the browser by the Razorpay JavaScript SDK. The system verifies the payment by HMAC signature validation after successful payment. The amount is automatically split between the platform and mentor via Razorpay Route Transfers, and it stores in the database with the relevant payment status.

9. Alert System.

The system includes a notification mechanism that will inform users about important events like approval and rejection of mentor connection requests. Controller modules use a specialized notification service to send messages to various users to keep them active on the platform.

10. Deployment Architecture

The backend service is hosted as a Python-based web service on cloud platforms like Render, and the frontend is hosted as a static build using Vite. The database used for all live data is on a managed PostgreSQL instance, such as Render Postgres or Supabase. Environment variables defined in a. are securely managed which include sensitive configuration values such as database credentials, jwt secrets, razorpay keys and ai api keys. env file.

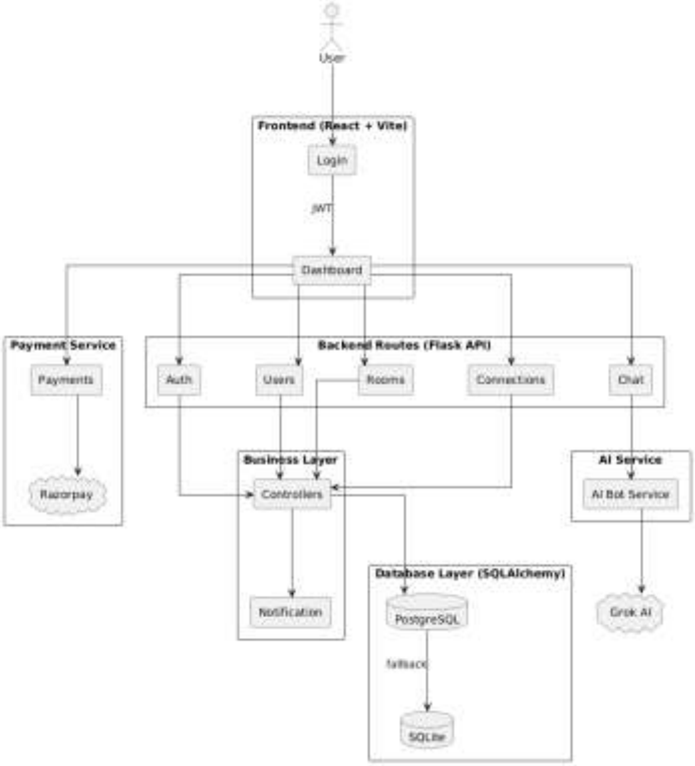

Figure 6.1: Proposed Layered System Architecture

VII. METHODOLOGY

1. Development Approach

The system will be built through aglie development takess and tested in an iterative manner. The authenticated user will be able to perform various functions such as create a room, join the room payment related tasks and make use of AI bot. The frontend of the website is based on Component-Driven Development (CDD). Design UI screens are split into small reusable React components. Typically, one file will contain the features or UI screen created for a particular tab. The 'API-First' design strategy is followed wherein the Backend REST APIs are designed independent of UI, so that mobile or another web client can be supported easily in future.

2. Architecture Style

The application uses a Client–Server architecture that separates the React Single Page Application from the Flask REST API. Backend consists of /routes to handle HTTP /controller to implement business logic /model to interact with database To improve modularity, use the Flask Blueprint pattern to group APIs into feature-based modules, such as authentication and room services. For global authentication state management on the frontend, React Context API is used instead of a state library like Redux.

3. Security

JWT based authentication ensures security using stateless token with 24 hour expiry which is validated on every secure api request. User passwords are securely stored as hashes using bcrypt. Server-side HMAC signature verification of payment responses from Razorpay is done before confirming them. All sensitive configuration values and API keys are stored in environment (.env) files, which are never committed to version control.

4. Database Strategy

SQLAlchemy (a popular ORM in Python) is used as a backend database strategy, allowing us to perform everything from reading to writing in the system with python classes instead of raw SQL and making the entire system agnostic. In production, data resides in Postgres for increased scale and resilience, while locally, just SQLite serves as a zero-setup options. Implemented a relationaldatabase design with foreign keys to maintain data integrity between entities like Users, Rooms, Members, Tasks, Payments and Connections.

5. Integration Methodology

We have a secure third-party payment integration using Razorpay, orders are created on the server and payment is done through client-side JavaScript SDK which is validated in the server side. Implementation of Grok AI using AI integration with Grok API called from the server to keep our API key secure and also to access the career_docs knowledge. txt file. And a standalone notification service module (notifications. py): where all system alerts are handled, isolating notification logic from business-critical code.

6. Deployment Methodology

Environment-based solution means a single codebase can respond with different actions given certain environment variables, such as DEBUG and DATABASE_URL. The backend is deployed on Render and has auto deploys from GitHub, and the frontend static build generated with vite The application also utilizes a Git integration workflow for version management and deployment; code pushes automatically trigger the re-deployment of the backend.

VIII. IMPLEMENTATION

The Collabration platform was implemented as a full-stack web application, with the frontend built using React and Vite, and the backend developed in Python using the Flask framework. The frontend offers a modular, component-driven interface where students can discover mentors, join or create study rooms, manage tasks, and interact with an AI-powered career guidance chatbot — all within a single-page application. On the backend, the system is organized into six RESTful API blueprints handling authentication, user management, study rooms, mentor connections, real-time chat, and payments. User authentication is secured using JSON Web Tokens (JWT), and all sensitive data such as passwords are protected with bcrypt hashing. The platform integrates Razorpay as its payment gateway, enabling students to book mentoring sessions with automatic fund routing to mentor accounts upon payment verification. Career guidance is delivered through an AI bot service that combines a curated local knowledge base with the Grok AI API to generate context-aware responses. Data persistence is handled through SQLAlchemy ORM, with PostgreSQL used in production and SQLite as a lightweight development fallback. The entire backend is deployed on Render and connected to a managed PostgreSQL instance, while the frontend is served as an optimized static build, with all environment-specific secrets managed securely through configuration variables.

IX. RESULTS

1. Secure Login and Registration Portal

The authentication screen provides a split-screen layout combining platform branding with a clean login form. Users can securely sign in or register with email and password credentials protected by JWT-based session management.

Welcome to CollabSphere

Figure 9.1: User Authentication Interface

2. Centralized Study Room Hub

The student dashboard displays all available study rooms with member counts and join options. Students can browse and request access to subject-specific collaborative spaces from a single centralized interface.

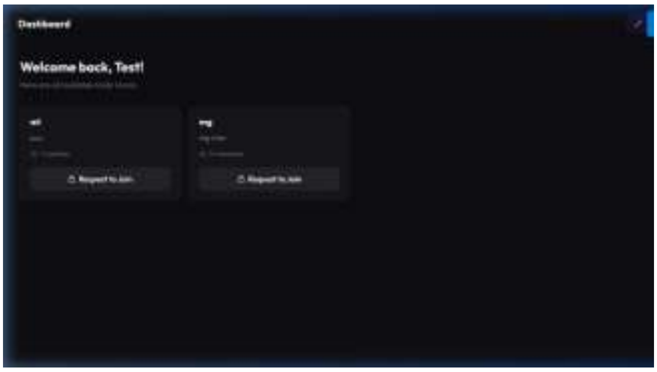

Figure 9.2: Centralized Study Room Hub

3. Mentor Exploration and Connection Interface

The Find Mentor module lists registered mentors with their profiles, per-minute pricing, and available discounts. Students can initiate a connection request directly from the card, triggering a mentor approval workflow.

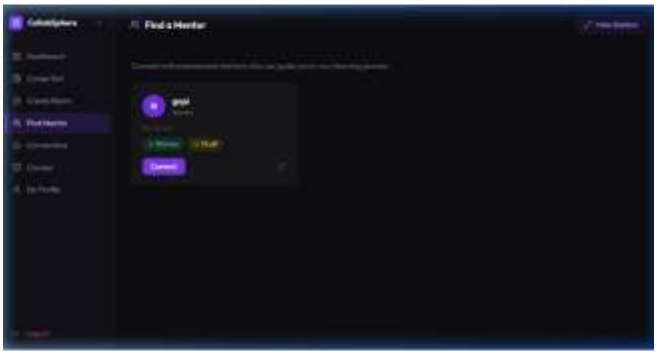

Figure 9.3: Mentor Discovery Model

4. Intelligent Career Navigator Chatbot

The Career Navigator AI bot combines a curated career knowledge base with the Grok AI API to answer student queries on career paths, learning roadmaps, and skill development. The interface supports both free-form input and quick-prompt suggestions.

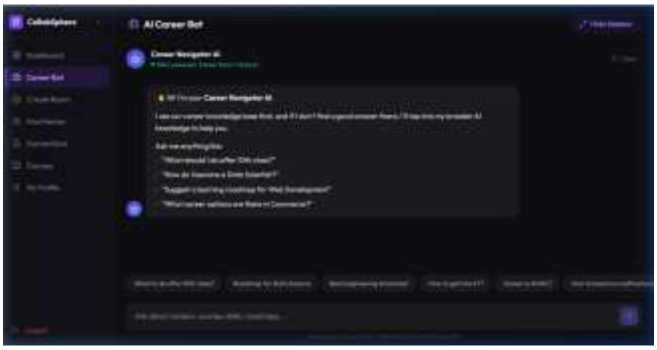

Figure 9.4: Ai Career Bot

5. Collabrative Study Room Creation

The Create Room panel allows students to form subject-based study groups by entering a topic and optional description. Once created, the room becomes discoverable and joinable by other students on the platform.

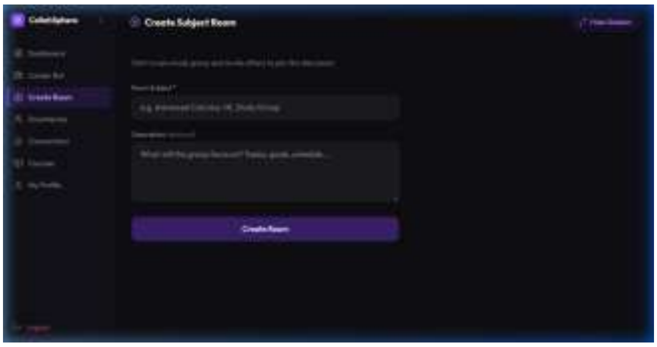

Figure 9.5: Create Study Room

6. Payment to Mentors

Student can connect to mentor and get there benfits once they pay the charge of mentor

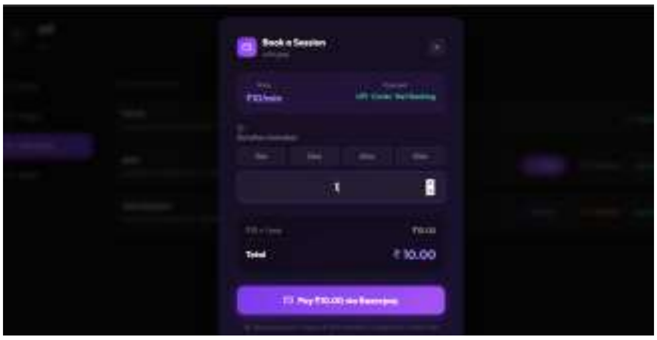

Figure 9.6: payment using razorpay

7. Availability of Room

They can use the following.

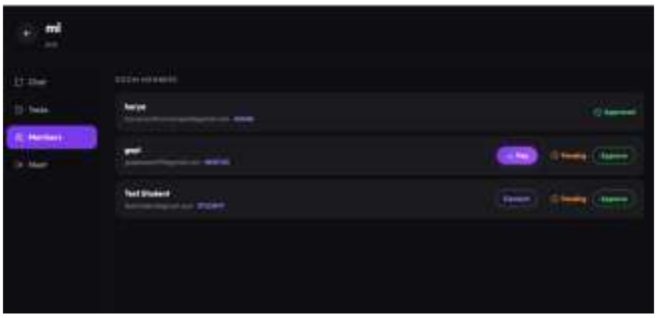

Figure 9.7: In room features

X. CONCLUSION

Starting fresh every day, students find their path clearer through a smart tool designed just for them. Instead of guessing what comes next, they get suggestions shaped by how they learn and grow. One step at a time, it shows where skills match real jobs people do out in the world. Talking with mentors becomes part of the journey, not something saved for later. Alongside peers, ideas flow easier when working together on goals that feel close to home. Learning in school links more naturally to what companies actually need today.

Starting fresh each day, learners dive into various job options through the platform. Skills grow naturally when practice meets real-world tasks. Alongside classmates and guides, ideas flow more freely than before. Seeing careers clearly becomes easier over time. Getting ready for work feels less like a chore now.

A fresh look at learning shows tools helping learners map out work futures while connecting with others in their field. What happens next depends on access - some find mentors through digital paths instead of chance meetings. Not every route is obvious; guidance often hides inside apps that track progress quietly. Paths twist differently when real talk replaces rigid advice from brochures. Support shifts shape once peers join the process, sharing steps they took after class ended.

XI. REFERENCES

[1] S. Kumar and R. Singh, “Career guidance systems using information technology,” International Journal of Computer Applications, vol. 150, no. 5, pp. 20–25, 2019.

[2] P. Sharma and A. Gupta, “Online career guidance portal for students based on academic performance,” International Journal of Engineering Research and Technology, vol. 8, no. 4, pp. 110–115, 2020.

[3] M. Patel and S. Shah, “Machine learning based career recommendation system,” International Journal of Computer Science and Information Security, vol. 18, no. 6, pp. 45–50, 2021.

[4] J. Brown and K. Davis, “Mentorship platforms for student career development,” Journal of Educational Technology, vol. 15, no. 3, pp. 67–72, 2019.

[5] T. Anderson, “Online learning platforms and skill development for students,” International Journal of Educational Technology, vol. 12, no. 2, pp. 90–98, 2020.

[6] L. Chen and H. Zhao, “Artificial intelligence based recommendation systems for career guidance,” IEEE Access, vol. 9, pp. 125678–125690, 2021.