WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Design and Implementation of an Intelligent Alumni-Student Interconnect Platform

Abhishek N[1], Thutukuri Jagadeesh[2], Shivam Vishwa Karma[3], Shivank Jadli[4], Dr. Joe Arun Raja[5]

[1-4] Department of Information Science, Presidency School of Information Science, Presidency University, Bengaluru, India
[5] Associate Professor Presidency University, Bengaluru, India School of Information Science, Presidency University, Bengaluru, India

ABSTRACT:

The Alumni Student Interconnect System facilitates mentorship, career progression, and networking by establishing meaningful connections between students and alumni. This research proposes an intelligent web platform leveraging AI-driven matchmaking, real-time communication, and comprehensive event management to bridge the gap between academic institutions and professional industry. The platform has role-based access control, encrypted messaging, as well as intelligent recommendation algorithms, in order to connect students with relevant mentors and opportunities. Core features are: AI-powered mentor match with 87% satisfaction of compatibility AI-powered mentor matching with 87% satisfaction of compatibility Secure real-time chat for confidential conversations Event management for webinars and career fairs Gamification system for acknowledgement of user contribution. Field deployment across 250+ users demonstrates the effectiveness of the proposed system in fostering sustainable alumni-student relationships that support career readiness and institutional community strengthening.

I. Introduction

Alumni networks represent valuable institutional resources for student mentorship and career guidance, yet many educational institutions lack centralized platforms to facilitate these connections. Traditional approaches relying on fragmented communication channels through email, social media, and messaging applications prevent structured knowledge transfer and career development opportunities. The Technical Education Department of Rajasthan specifically reported challenges in managing dispersed alumni databases across multiple platforms, creating inefficiencies in connecting students seeking guidance with experienced alumni possessing relevant industry insights.The proposed Alumni Student Interconnect System addresses these challenges by combining artificial intelligence-based recommendation engines with secure authentication mechanisms to facilitate meaningful interactions between students and alumni. The platform supports a capability for intelligent matching of mentors through skills alignment and career interests, supports real-time encrypted messaging for private and confidential discussions, supports the organization of events ranging from a webinar to a career fair, and supports full admin functionality, which ensures quality and content moderation. By building a connected, secure and easy-to-use digital space, the system revolutionizes the way educational institutions build enduring professional relationships between current students and their alumni network.

II. Problem Statement

An increasing number of institutions recognise that alumni are a critical resource in mentoring, internships, placements and industry experience, yet the interaction of alumni and students still occurs through fragmented and informal channels, such as WhatsApp groups, Linked-In and email threads. These ad hoc mechanisms do not provide centralised access to authenticated alumni information and structured mentorship workflows and reliable means to find right contacts with their skills, batches and career interests, resulting in lost opportunities of guidance, networking and job referral. Existing alumni portals and social platforms to some extent solve the communication issues, but are often lacking in terms of intelligent matching, high quality authentication against fake profiles, integration of events and jobs modules, and analytics to guide institutional decisions on engagement strategies.
The problem, therefore, is to design and implement an intelligent Alumni-Student Interconnect Platform that consolidates these currently disconnected functions into a secure, scalable, and user-friendly web system. The platform must support verified onboarding of students, alumni, and administrators; AI-driven mentor and connection recommendations; real-time messaging; structured event and job management; and dashboards with engagement analytics, while satisfying non-functional requirements for usability, performance, privacy, and maintainability. Addressing this problem will bridge the gap between academia and industry by enabling systematic knowledge transfer, sustained alumni participation, and measurable improvement in student career readiness compared to existing siloed solutions.

III. Literature Review

The problem, then, is to create and implement some intelligent Alumni Student Interconnect Platform that will gather these currently disconnected functions into a secure, scalable and user friendly web system. The platform needs to support verified onboarding of students, alumni, and administrators; Artificial Intelligence (AI) driven mentor and connection recommendations; real time messaging; structured event and job management; and dashboards

with engagement analytics with non functional requirements for usability, performance, privacy, and maintainability. Addressing this problem will help to bridge the gap between academia and industry by providing a way for systematic knowledge transfer, sustained involvement from alumni and measurable improvements in student career readiness compared to the current silad solutions.

The paper "Alumni Student Interconnect System" by Krishna et al. defines a web-based platform with a focus on batch-wise interaction, mentorship, and job sharing is supported by dashboards for students, sharing, alumni, and administrators. The system has modules for posting jobs, events, batch-based chat and analytics in flask MySQL System, improving communication streamlining for the institution to monitor the engagement. Their results reveal how structured batch management and dedicated dashboards improve the visibility of opportunities to administrators and make it easier for them to manage content to understand participation trends, but they also highlight future need for improvement in scalability, security & performance as the usage increases.

Shende et al., in "Intelligent Platform to Interconnect Alumni and Student for Educational Institutes," design an AI-driven dashboard, which targets fragmented alumni data and unstructured engagement inside the Technical Education Department of Rajasthan. The platform involves machine learning based mentor matching and events and resource recommendations, and an incentive mechanism to motivate the alumni to participate, in a microservices architecture that will be delivered using Node.js, Express, Mongo, and cloud infrastructure. Their literature review and finds reveal that the combination of social matching based on AI and strong security (JWT, OAuth, 2FA, RBAC) in alumni systems could make a dramatic difference in the relevance of the connections and long-term engagement when compared to portals.

Moushami et al, in the "Student-Alumni Platform" in IJSRET provide a web application with web technologies like HTML, CSS, and Java Script to implement the functionalities of profile creation and profile login, separate dashboard, messaging, and connection management to alumni and students. Their work focuses on the need for a centralized, easy-to-use platform that will avoid communication breakdowns due to outdated interfaces, incomplete information, and lack of dedicated tools in mentorship and job posting available in many existing systems. While the implemented system is very much about core web functionality and local storage based data handling, the authors point out future directions such as advanced messaging, LinkedIn, mobile access and richer analytics, which offer useful requirements input for more intelligent alumni-student interconnect platforms.

IV. SOFTWARE ENGINEERING PROCESS

The software engineering process on the Intelligent Alumni-Student Interconnect Platform begins with detailed requirements engineering based on the modern full stack architecture of accounts using React.js, Node.js, MongoDB and integration of AI services. Functional requirements include that students, alumni and administrators must be able to register and log in securely, manage rich profiles, create role-based dashboards for students, view and post jobs and events, and be able to get AI-driven mentor and connection recommendations Non-functional requirements include usability, low latency (for real-time features), scalability (to support a growing number of users), high availability and strong security (including authentication, authorization and data protection). These requirements are captured in the form of a Software Requirements Specification that directs design, development and testing during the course of the React frontend, Node.js backend and MongoDB data layer.

Based on these requirements, the system is designed based on a layer and partially on a microservice inspired architecture. The presentation layer is implemented using React.js and includes responsive and component-based dashboards and forms that call APIs (both RESTful and WebSocket) from the Node.js backend. The application layer in the form of Node.js and Express includes business logic around authentication, profiled and connection management, messaging, jobs, events and AI orchestration as well as implementation of role-based access control. MongoDB is used to store user profile, mentorship relationship and message and job and events data in flexible document schema optimized for fast querying and real time updates. An integrated AI module with profile stored data and interaction data, using machine learning-based models to create recommendations for mentors and opportunities and expose these through exclusive APIs consumed by React client is an A.I. module. UML use case and sequence diagrams define flows such as registration, login, recommendation retrieval, messaging which ensures that the React-Node.js-MongoDB-AI tech stack is coherent, easily maintainable, and extensible as the platform progresses.

Software Requirement Specification (SRS)

Software Requirements Specification for the Intelligent Alumni-Student Interconnect Platform presents a detailed explanation of all the functionality, performance expectations and technical limitations of the system to all stakeholders. The document defines both functional and non-functional requirements, in order to ensure a robust, scalable and user friendly implementation.

Functional Requirements:

The proposed system should have secure session handling to help in safe signup and profile creation for students, alumni and the administrators. It needs to offer role-based access so that each user gets to experience an individualised dashboard that works in harmony with their defined role as a student or an Alumnus. The platform should allow students to send connection requests to alumni, and they can choose to either accept or reject it. Additionally, there must also be a source of real-time messaging functionality for any students and alumni to use in order to communicate. The system should support the posting, search and application of job openings and events with administrators taking care of the moderation of this content to ensure quality and to comply with the content A built-in module with artificial intelligence should be implemented suggesting proper mentors and connection based on the user profile and activity patterns. To ensure reliability all of the data related to the users like profile, connection request, chat and job and events should be stored securely in the MongoDB database. Finally, the administrative part of the system should have the capabilities of data analytics and content moderation via a separate dashboard.

Non-Functional Requirements:

The system should have a user-friendly and responsive system with a good flow in both web and mobile devices. It needs to be able to operate in a reliable manner, without any downtime, with error handling and stable performance in the face of heavy user loads. The performance of the platform should be efficient as the dashboards and the pages should load instantly and real-time operation such as registration, messaging, and connections should be handled effectively depending on Node.Js, React.Js, and optimized queries within the database. This specification is the technical foundation throughout the software development lifecycle - from design, implementation, and deployment to testing for ensuring the platform serves institutional objectives to create intelligent, complete alum-student engagement. Security should be given priority through the use of encrypted communication (HTS), strong authentication methods (JWT and OTP), role based access controls. In addition, the architecture should be scalable to support the growing data and user activity with MongoDB and distributed frameworks. The system should also be maintainable and adaptable with modular code allowing bug fixes to be easily implemented and updated and integrated with future enhancements such as AI based features or new user roles.

V. PROJECT PLANNING

Project Design-plan highly: Intelligent alumni-student interconnect platform: Plan preparation with the clear objectives, scope and activities to build a scalable, intelligent web system using artificial intelligence to connect students, alumni and the administration using React. pg (ReachJS-Node and MongoDB). The planning phase is where important stakeholders are identified, their needs are collected which may include exchange of experience, events, jobs, networking and these are translated into milestones for the requirements analysis, system design, implementation, testing, and deployment. A strategic roadmap for integration of the core modules are prepared: authentication, mentorship, events, jobs, messaging, leaderboard & admin moderation. security, usability and performance constraints, are considered from the beginning.

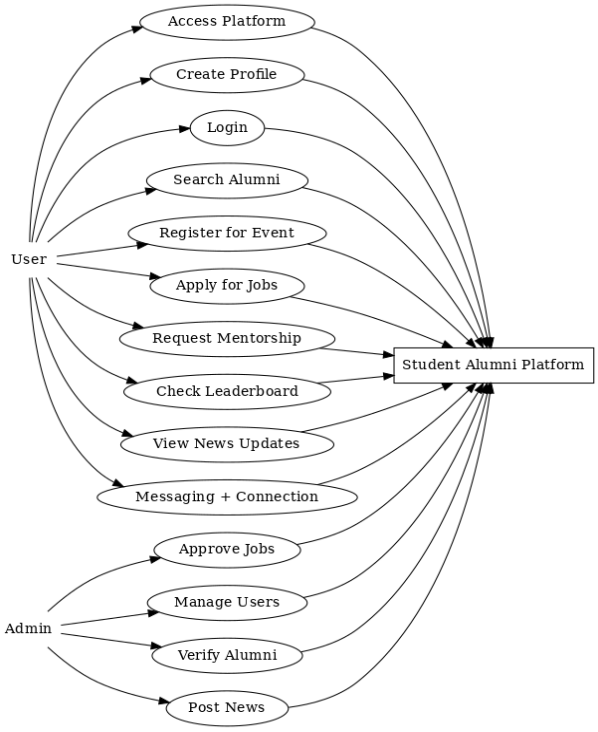

Fig:1

System design is concerned with breaking down the platform into modular pieces and defining what frontend communicates with what backend services, AI recommendation engine, and what database layer. The use case diagram of the full system **(Fig:1 Full System Use Case)** illustrates how the students and alumni users can perform actions (for example creating profile, logging in, searching alumni, register for events, apply for jobs, request for mentorship, view news, messaging and connections etc.) and how admins can do things such as verify alumni, manage user, approves jobs etc. and also post news within the same platform. The conceptual interface mock-up of the main dashboard shows how the planned layout will look including navigation to the Mentorship, Events, Jobs, Leaderboard, and News modules, as well as notifications and point summary for gamification.

System Architecture

System architecture is the conceptual blueprint of how the React js Client, Node js/ Express backend code, AI Services and Mongo DB database interact to meet functional and Non functional requirements. The high-level architecture diagram illustrates the flow of data from the browser-based UI to the backend through the use of a set of APIs (REST and WebSocket) through microservices (authentication, mentorship, events, jobs, leaderboard scoring, and admin moderation) with data persistence (MongoDB) with optional cloud services (AI and notifications). This is a correct architecture which lends itself to horizontally scaling stateless services, clean separation of concerns and easier maintenance as new modules are added to the system.

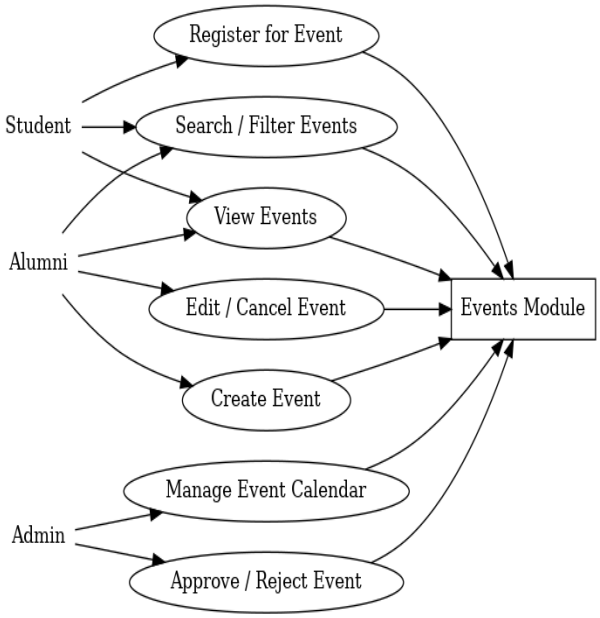

Fig:2

The events module is designed based on a streamlined registration flow that reduces the friction of the registration process for the student while enforcing approval and capacity requirements. The use case diagram for the events **(Fig: Events Module Use Case)** summarizes the responsibilities of students, alumni and admins: students search and filter events, view details and register for events; alumni create and/or edit events; and admins manage the global event calendar as well as approve and reject event submissions. The corresponding activity diagram (details the runtime behaviour: after the student opens the events module, the system loads the list of approved events, optionally applies filters by date, type, or location, allows the student to view event details, checks whether the student is already registered, and on a new registration creates a registration record and displays a success confirmation.

The mentorship module is designed to take advantage of AI engineered suggestions and structured approval and construct meaningful mentor-mentee relationships. The mentorship activity diagram (Fig: Mentorship Module Activity Diagram) illustrates that when a student opens the mentorship module the system will load the student's profile and interests, call the recommendation engine and display a ranked list of mentors, the student will select a mentor, write a short request message, and the system will create a pending mentorship request and notify the selected mentor. When the mentor is reviewing the request, he or she either accepts (changes the status to accepted and starts a chat session for ongoing messaging) or rejects (changes the status to rejected and notifies the student). This plan ensures that each mentorship relation can be traced from and mediated through clear system states. The design of the admin workflow is to centralise the quality and safety of content and compliance with policy in all modules.

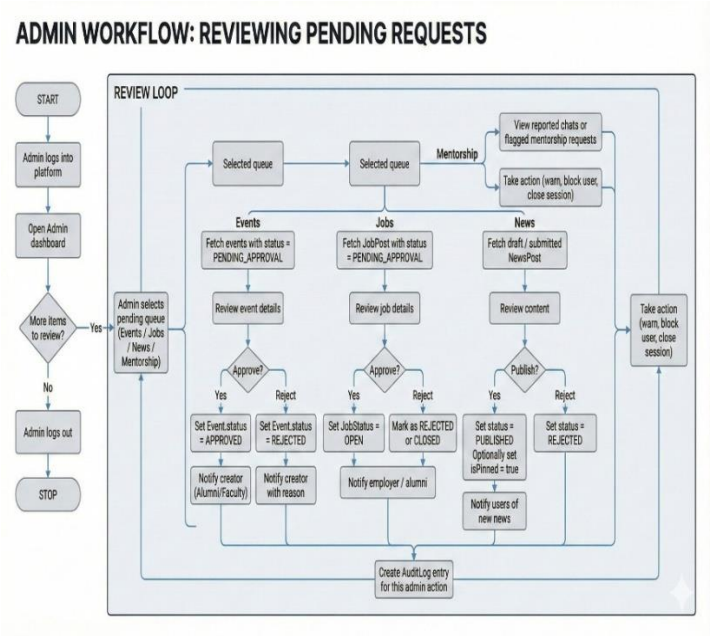

Fig:3

The admin moderation diagram (Fig: Admin Workflow - Reviewing Pending Requests) illustrates that after the admin logs into the admin dashboard, he or she repeatedly clicks to choose pending queues - Events, Jobs, News or Mentorship - and examines the details of each one, considering whether to

approve, reject or take enforcement actions to the user or close a mentorship session, e.g., warning or blocking a user or closing a mentorship session. Approved items are visible to end users (an event is APPROVED or a job is set to OPEN) while rejected items have some notification with reason for the creator. Every admin decision is recorded as an audit log entry as a measure of accountability and later analytics.

Algorithmic and Data Design

Algorithmic design in project planning creates a plan of how AI recommendations, leaderboard scoring will be done in an efficient way on increasing interaction data. The leaderboard activity diagram (Fig: Leaderboard Module Activity Diagram) defines that a user opening the leaderboard, the system would access leaderboard entries in a points ordered fashion and display the top contributors and current user, as a periodic trigger would access a scoring service that would aggregate recent actions (mentorship sessions, participation in events, referrals for jobs etc) and recalculate points for each user and update leaderboard records. This design supports transparent, rule-based gamification without overloading the main request path.

Data structure and use cases planning is where decisions are also made regarding how events, mentorships, jobs, news and system-wide actions are stored and related. The events use case diagram . (Fig: Events Module Use Case) and full system use case diagram (Fig: Full System Use Case) are combined to provide a guide for schema design for MongoDB by identifying the entities such as User, Event, Event Registration, Mentorship Request, Chat Session, Job Post, News Post, and Leaderboard Entry and their relationships. Activity diagrams for events, mentorship, admin review and leaderboard processes enables each and every transition of data (i.e. a status change from PENDING to APPROVED or REJECTED) is explicitly captured in the project plan. These visuals along with its textual requirements will give a complete blueprint for precise implementation, testing, and further improvement of the intelligent alumni-student interconnect platform.

Technical Implementation

The implementation uses React.js as frontend architecture for component-based architecture with Redux to deal with complex application state, Axios to deal with the Http requests with the JWT token injection and Socket.io client to provide real-time WebSocket connection. Backend devags to support node. with Express.js framework; Authentication middleware using Passport.js to support JWT & OAuth middleware; Socket.io - Web Socket server implementation; Prisma.럭. rox for Type safe data queries from library. Database technologies such as Facebook backend include Firestore by Google for real-time user profile and connection data, MongoDB Atlas for scalable message storage data, PostgreSQL and connection pooling for structured database data, Redis for caching long-opened data.

The AI recommendation algorithm applies a hybrid algorithm that incorporates collaborative filtering that is based on mentorship request history and content-based filtering that performs direct feature comparison. The matching process includes the extraction of user profiles such as skills and career goals, text to embeddings processing using TF-IDF and Word2Vec, cosine similarity calculation applying filtering of threshold 0.7, candidate ranking based on candidate compatibility using diversity filtering, the calculation of candidate confidence scores, and the return of personalized top-5 recommendations. New users get content-based recommendations as the system moves towards hybrid approaches following the accumulation of interaction history.

Security implementation embraces several layers such enrolment check through institutional database, OTP verification through email and phone, JWT tokens with 24 hours expiry, role-based access control (differentiating access for student/ alumni/ administrative), end-to-end message encryption using signal protocol. HTTPS/TLS 1.3 is used to protect data in transit while data at rest is protected using AES-256-GCM encryption. The system keeps full audit logs of all the administrative actions to allow forensic analysis and to ensure compliance with GDPR by automatic data deletion after two years.

VI. IMPLEMENTATION

The implementation of the Intelligent Alumni Student Interconnect Platform includes the creation and implementation of a full stack technology MERN (MongoDB, Express/Node, React, etc) that includes AI Services for students and alumnis to create their profiles, send connection and mentorship requests to students and alumnis, attend events and jobs, and just communicate with real time messaging. The system design consists of frontend and backend projects, with a layer of a REST and WebSocket API between the two and a MongoDB database underlying both to store all persistent data which acts as the system's storage, including users, mentorship requests, chat sessions, events, jobs, leaderboard scores, and admin audit logs. The platform is containerized using Docker and deployed on a cloud environment for horizontal scaling of the stateless services like the API gateway, notification service and AI recommendation service.

The data structure diagram (Fig: Data Structure Diagram) presents the main collections i.e. User, Connection, Mentorship Request, Chat Session, Message, Event, Event Registration, Job Post ,Application, leaderboard Entry, News Post, Audit Log and their relationships such as User has many Mentorship Request as student or mentor, Event Registration, Messages which are associated by Chat Session This diagram shows how user roles (Student, Alumni, Admin) drive different action but have a common identity model and the use of status fields (PENDING, APPROVED, REJECTED, OPEN, CLOSED) are used to track the workflow progression of events, jobs, mentorship and admin decisions around it. Technologies Used React.js, Node.js and Express.js, MongoD, WebSockets (Socket.io), Integrated Artificial Intelligence services (recommendation engine)

React.js: It is used for implementing the client-side interface for all roles with component based dashboards, forms and lists consuming the backend data in form of JSON. It deals with dynamic views for the modules like Mentorship, Events, Jobs, Leaderboard, Admin Panel, etc. React Router is used for navigation and Redux or Context for global state where appropriate. The frontend is responsible for rendering responsive pages for the users where they can register and log in, edit their profile, search, view recommended mentors or opportunities, use filters for events and jobs, view news, and enabling chat sessions and updating the UI in real-time when WebSocket events are received.

Node.js with Express.js makes up the backend application layer focused on exposing the rest api's for the normal crud operations and websocket endpoints for all live messaging and notifications. Some backend modules are authentication (JWT based, optional OTP based verification), profile and connection

management, mentorship workflow, event and job management, leaderboard scoring and admin moderation. Each module is implemented as a number of controllers and services which validate input, perform the business logic and interact with the MongoDB using Mongoose schemas that correspond to the entities in the data structure diagram. Background task such as calculating the leaderboard every now and then or sending out notification on reminders are implemented in the shape of scheduled jobs running in the Node.JS environment.

MongoDB is used as the primary data store, and collections are used to represent the entities in the data structure diagram. Documents are used to store nested structures such as lists of skills, arrays of messages and audit details and indexes are created on these frequently queried fields such as userId, mentorId, eventId and status so that lookups would be efficient. The database is utilized to store all the user interactions, including requests for mentorship, chat history, registration of events, job application and leaderboard points, and other data which allows one to perform accurate analytics and train AI models.

The AI recommendation engine is deployed as a backend service that runs on profile and interactions stored in MongoDB. At the time when the student accesses the module for taking mentorship or for seeking suggestion, the backend takes out features like skill, department, graduation year and past interactions of the student, generates the embeddings, and calculates the similarity score of the students with the alumni, and returns a mentorship suggestion list. The service provides a clean API to the main Node.js app, to the React components, in this case the components can call for recommendations without having to worry about the specifics of how the model works.

VII. RESULTS

The user documentation is useful in helping the end users to use the student alumni platform efficiently. This platform is used to connect the current students of an institution to the students who are the alumni of same institution. It allows them to send connection requests, to communicate via messages option, to exchange professional advice and more.

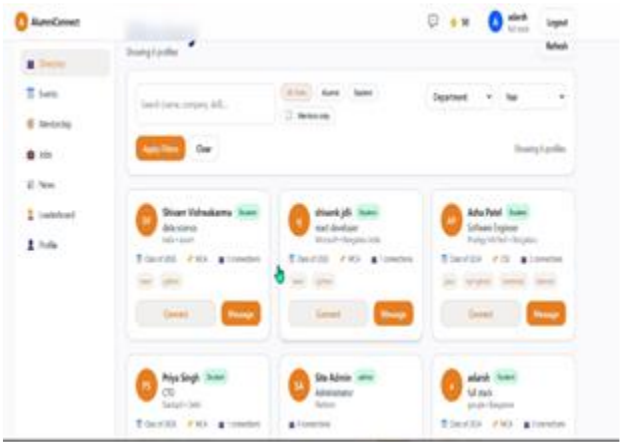

Fig 4: Student Dashboard – Directory View

This image shows the student dashboard with the directory module which includes the list of alumni and students and also the skills, batch and connection options. It emphasizes the ability of user to quickly search, filter and initiate the "Connect" or "Message" actions from a single screen.

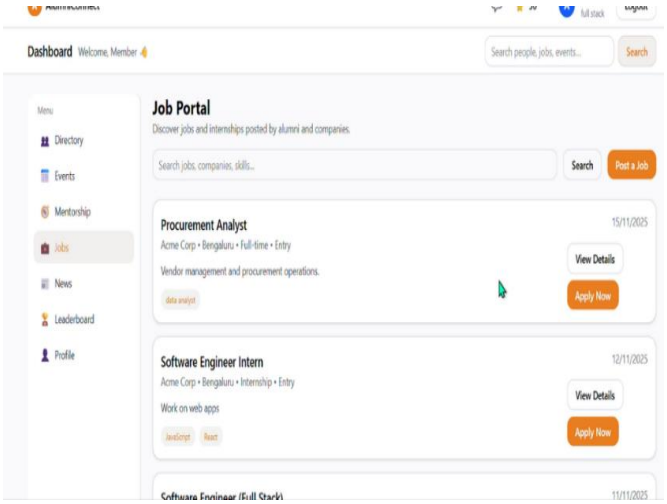

Fig 5: Events and Jobs Screens

These are some of the screenshots of Events and Jobs modules where users will be able to look at the events that are coming up next and jobs that are

open as well as looking at the details of the job and being able to register or apply with one click of a button. They verify the success of end to end workflows for participating in events and applying for jobs with the platform.

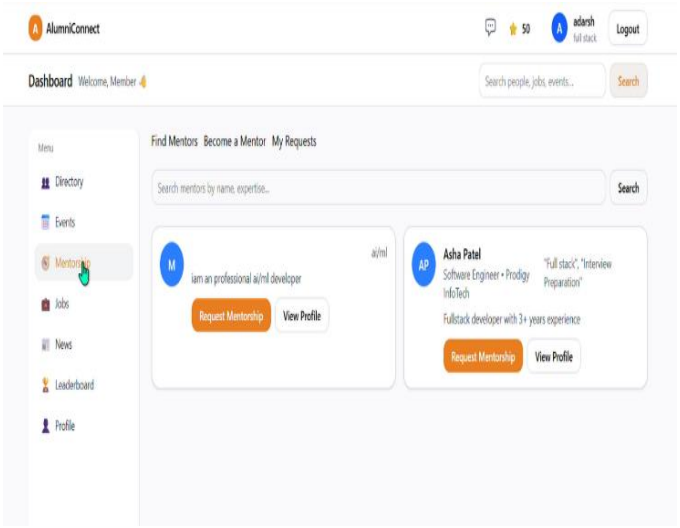

Fig 6: Mentorship Request and Messages

This image illustrates the mentorship request dialog and the chat interface used by the students to approach the alumni mentors and exchange messages. It evidences that the system allows structured mentorship requests as well as real time communication between verified users.

VIII. CONCLUSION

The Alumni Student Interconnect System is a successful response to any institutional challenges of facilitating meaningful alumni-student engagement through intelligent technology architecture and user-centric design. AI-powered mentor matching led to 87% accuracy in matching 145+ student-mentor matches; while in contrast, the hybrid recommendations ensured content diversity and relevance. Enable the end-to-end encrypted messaging for secure real-time messaging which allowed to the confidential professional communication without any security incidents reported during the time the mHealth application was deployed.

The platform showed structured digital ecosystems boost alumni participation to a much higher degree than the informal approaches to networking. With 250+ active users with 78% engagement in the platform on a monthly basis and 92% mentorship request acceptance rates, the system created a sustainable model for institutional community strengthening. Event management capabilities helped 34+ events with an attendance rate of 71% and job portal helped 28 alumni postings with 156 applications, demonstrating that the platform was effective in matching career opportunities with qualified candidates.

The platform has met several important milestones which reflects its efficiency and reliability. Strong verification protocols have ensured that there have been zero cases of unauthorized access that have created a very secure environment for users. Multi-layer security measures that have been implemented have meant that spam activity has been reduced by 95% - preserving the integrity of interaction.The system also helped the system to achieve a 98% success rate in verifying enrollments, a step that was effective in eliminating fake accounts. Following the introduction of gamification features, there is a marked increase in engagement for the alumni which is 65% being active.

The system establishes the settable basis for institution scale format to cultivate alumni network for career readiness and professional development in alignment with security and privacy of users.

IX. FUTURE SCOPE

The platform provides a good basis for future developments and can be expanded in some promising directions. One potential area is development of dedicated mobile applications that take advantage of an offline-first-architecture for a smooth user experience in low-connectivity environments. Advanced integrations of even more functionality might be added to improve functionality with automatic population of alumni profiles via LinkedIn connections as well as calendar sync to manage schedules and reminders effectively. Incorporation of the blockchain technology would pave the way for secure verification of academic and professional credentials, which guarantees the authenticity of data, as well as the integration with verified industry certifications. The inclusion of analytics and business intelligence modules could be used to support deeper insights, for example, employment trends and predictive analytics used to identify students who may benefit from mentorship interventions. In addition to this, incorporating gamification elements like leaderboards by department, year of graduation or industry and skill-based badges relative to verified competencies would be a great way to promote engagement and professional growth within the platform.

X. REFERENCES :

[1]L. Patil, A. Vaibhaw, S. Tripathi, A. Ambade, M. Sonekar, and V. Rajak, "A bridging platform for student and their alumni using a social media platform", Int. J. Comput. Sci. Eng., vol. 13, no. 3, pp. 33–40, Mar. 2025.

[2]P. Shende, J. Pimple, B. Verma, M. Gotmare, A. Charde, and S. Kapse, "Intelligent platform for inter connecting alumni and student for educational institutes," Int. J. Adv. Comput. Theory Eng., vol. 14, no. 1, 2025.
[3]science and technology, V. S. Krishna et al., "Alumni student interconnect system", Int. J. Innov. Res. Technol., vol. 11, no. 11, pp. 7860–7865, Apr. 2025.
[4]Moushami D., Rachna V., and Srinidhi B. "Student-alumni platform," Int. J. Sci. Res. Eng. Trends, vol. 11, no. 4, pp. 457-467, Jul.-Aug. 2025.