

JobQuench: An Intelligent and Automated Placement Management System for Enhanced Campus Recruitment

Mr. Bhuvaneshwaran B

Assistant Professor
Rajalakshmi Engineering College
Chennai, India

bhuvaneshwaran@rajalakshmi.edu.in

Reshma R

Computer Science and Engineering
Rajalakshmi Engineering College
Chennai, India

210701212@rajalakshmi.edu.in

Soniya V

Computer Science and Engineering
Rajalakshmi Engineering College
Chennai, India

210701254@rajalakshmi.edu.in

Abstract - The system focuses on the elimination of the main shortcomings and existing labor-intensive methods of recruitment on campuses by using a centralized and automated platform system. Many placement strategies still exist in the following major issues: putting the employers and students into the wrong positions, actually involves a lot of paperwork, and makes the availability of jobs not well perceived. Using the development tools of the contemporary age, this system empowers the front end in Flutter, the backend data management in Django REST API, and SQLite for light data storage. On ways through which students, placement officers and the employers are created, the system automates functions like resume database, job board as well as scheduling for interviews. There are also complementary services like alumni-mentor and career-profile to help students enhance their employment opportunities. There are positive trends in the first outcomes of the implementation of the Placement System with regard to enhanced satisfaction of users and increased effectiveness of recruitment. The platform also helps minimize time and efforts needed for conducting placement activities as well as enhance partnership between academic environment and industry. This paper defines the system's architectural design and creates an understanding of how the design improves the placement practices, thereby filling the gap between classroom education and employment opportunities for graduates.

Keywords: *Campus Recruitment, Job Matching, Resume Screening, Placement System, Django, Flutter, REST Framework, SQLite, API Development, CRUD Operations, ORM, Database Integration, Notification System, Session Management, API Security.*

I. Introduction

The placement system introduces a unique set of features designed to enable students to elevate their profiles for the highly competitive job lifecycle in a way that redefines industrial expectations from academic institutions. The campus recruitment process in most institutions is still based on outmoded, manual processes that create inefficiencies like tedious paperwork, job matching difficulties, and a lack of accessibility. Most online placement systems available are either very basic, with limited functionality, or highly institution-specific and therefore not adoptable for wider use. Third-party websites tend to raise trust and data security issues, and institutions do not want to adopt these for placing

students.

In order to overcome these issues, Job Quench comes up with a complete and scalable placement management system that can be utilized by any university or college. Contrary to existing systems that are either too restrictive or not reliable, Job Quench offers a central platform which automates essential placement processes, including resume management, job postings, and interview scheduling, ensuring data security and openness.

The platform is constructed on Flutter for cross-platform support, Django REST API for backend operations, and SQLite for minimal data storage, which makes it efficient, scalable, and simple to deploy across several institutions. For instance, a university with various departments can deploy Job Quench to handle department-wise job releases, while a small college can use it to computerize their entire placement process without third-party solutions. Implementing a good student placement system will not just help the students alone but generate a good reputation for all the educational institutions. As highlighted by Patil et al., the online college placement system facilitates the management of student information and allows both the student and the placement officer to establish contacts with just a few clicks. [6] By consolidating all placement-related activities under one roof and reducing the paperwork associated with manual processes, placement management systems streamline the entire placement-driven process, benefiting both students and administrators and enhancing an institution's placement performance and industry interface.

Educational institutions can transform the way they place students and usher in a more customized and favorable career development process with the help of this evolution in technology and placement management. Through the combination of easy-to-use applications built in Flutter, such as Android applications and IOS applications, powerful backends in the Django framework, and database management via SQLite, institutions can offer students tailored placement advice, efficient job matching processes, and greater career prospects. This paper aims to explore the intersection of technology, human-centric design, and data-driven decision-making, with the goal of optimizing student

placement preparation and management, enhancing efficiency, and easing the transition from education to employment.

II. Literature Survey

The literature on placement management systems states that technology, through digital solutions, is automating traditional/reactive processes to make them more secure, accurate, and efficient for human resource and recruitment management systems in managing students' data and details [1][2]. These systems bridge the gap between recruiters, students, and placement coordinators by offering features like auto-document verification, instant placement status updates, and a secure account setup [3][4][5]. Many solutions have been explored, but most are based on web and mobile applications, which have customized features to suit some institution needs [6][7]. These solutions achieve scalability and convenience across devices by utilizing Spring Boot, which includes all its functionalities for backend services, and the React Native framework [4]. They prioritize cross-platform compatibility to ensure a seamless user experience [1][2].

These systems typically integrate role-based modules for recruiters, administrators, and students that provide focused data access and administration [8][9]. Specific innovations, such as the Excel-to-PDF converter for data processing, image preprocessing for document verification, customized tools for skill development, and so on, have pushed the limits of the performance of these systems [3]. This course incorporates mentorship, real-time exams, and expert support to equip students for the forthcoming challenges in this profession [5] [8]. These platforms offer a comprehensive strategy that transforms students' employability by assuming administrative tasks, while simultaneously preparing Our new batch of students by successfully bridging the knowledge gap between academia and industry [11]. On balance, shifting to web-based and cross-platform placement systems has dramatically enhanced data accuracy, reduced overhead, and increased institutional successes [6] [12].

The systems under evaluation underscore the crucial role of availability, user-centered design, access control, and data security in managing placement activities [4]. The majority of

these systems boast robust backends that securely store and retrieve data, facilitating the scalable and efficient management of vast amounts of student data through frameworks like Django, Laravel, and MySQL [8] [14]. There are more advanced features that enable role-based access for administrators, placement coordinators, and businesses, facilitating faster access to private data on a need-to-know basis without violating users' privacy rights [9] [13]. Administrative management ensures that only approved people can modify or view critical data, and they play an important role in protecting the integrity of the system [7] [15].

These tools aid in efficiently monitoring placement activities and student progress, simplify data administration, and enhance communication between the student and the placement coordinator [16]. Some systems also let employers filter applicants based on specific criteria that enable better alignment of student profiles with business requirements [5] [17]. This automated approach helps educational institutes manage placement procedures efficiently by eradicating administrative pressure and human error [10]. In sum, these are positive signs, which indicate a huge move towards optimizing the placement management process, aligning student readiness with the shifting market needs, as well as bridging the gap between academics and corporations [12] [14].

Recent advancements in machine learning and data analytics are making placement management systems more functional to provide predictive analytics [10] [18]. Now, systems can analyze the student profile, academic record, and skill set and suggest a career, which further increases the chances of successful placement [19] [20]. For example, by analyzing past placement data, the system can predict which companies are likely to recruit a particular set of students based on their profiles and past trends. Targeted matching thus helps placement coordinators streamline recruitment.

The Placement Management System brings a new-age method of managing student placements through the incorporation of automation, AI-based insights, and increased security. In contrast to conventional systems, it focuses on real-time updates, predictive analytics, and

III. METHODOLOGY

The methodology adopted for the project was a systematic approach directed at designing and implementing a placement management system suited to institutional needs. The paper illustrates the various stages and techniques that were applied from gathering the requirement to the actual system design, followed by the development, testing, and evaluation. User-

centric design should be focused, accounting for various stakeholders such as student needs, recruiter needs, coordinator needs, and administrator needs while keeping it efficient, secure, and scalable. The whole project employs agile practices to deliver iterative feedback that is better and enables cross-platform compatibility while utilizing both front-end React Native along with Spring Boot back end. Additionally, features such as document verification, predictive analytics, and role-based access control were developed systematically to optimize the placement process and enhance user experience.

Ensuring that the project keeps moving smoothly allows continuous improvement to be incorporated in the system so that it eventually leads to improving the placement outcome and bridges the gap that exists between academia and the industry.

3.1 Modules

3.1.1. Authentication and Authorization Module

User Access Control and Roles

The system has various roles, including students, recruiters, placement officers, and administrators, with different permissions set for each. This enables users to access only relevant features, such as posting jobs by recruiters and only viewing job listings by the students.

Login, Registration, and Security

This involves secure login and registration using encrypted credentials. The authentication in the backend will be taken care of by Django Rest Framework (DRF) using JWT or JSON Web Tokens. This will also provide a layer of secure session for the user. The credential of users will be sent over HTTPS, which cannot be intercepted.

Role-Based Access Control (RBAC)

RBAC allows the system to grant the ability to exert exact control over the performance of system functions. In detail, all roles are managed and only authorized users can access certain features. For instance, it can allow placement officers to approve applications and recruiters to manage job postings.

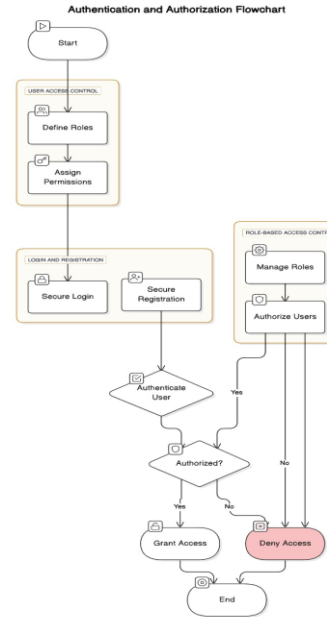


Fig.1 Authentication and Authorization Module

3.1.2. User Profile Management Module

The User Profile Management module depicted in Figure 1 allows students, recruiters and placement officers to manage their profiles. Students can fill in their academic details, skills, work experience and upload their resume and create a comprehensive profile that recruiters can view. Recruiters and companies can also create a profile with their company information, job roles, job description and placement details and statistics. Placement officers can also have a profile with access to administrative tools to manage both student and recruiter profiles. This module uses Django Rest Framework to store and retrieve data in a secure and structured way. Flutter's UI toolkit handles form and data validation smoothly. Data updates trigger Firebase notifications to students or recruiters at Figure 2. when their profile status changes.

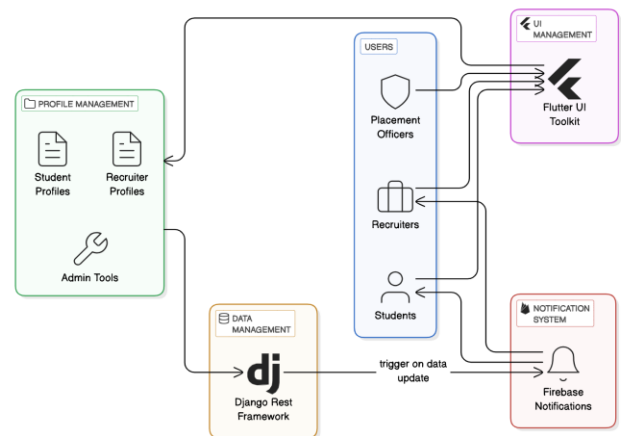


Fig.2 User Profile Management Module

3.1.3. Job Posting and Management Module

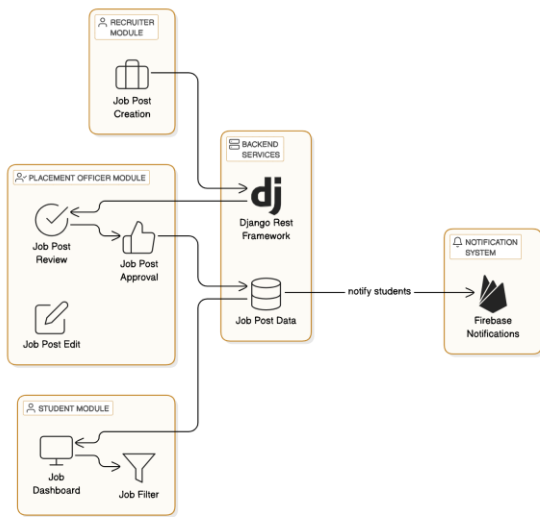


Fig.3 Job Posting and Management Module

This module Figure.3 makes job posting easier. Recruiters can create job posts with role, job description, eligibility criteria, application deadline and other details. Once posted, job openings will appear in the student's dashboard based on eligibility and preferences. Placement officers can review, approve or edit these job posts before they are published to maintain quality. Django Rest Framework stores job post data securely and communicates between backend and frontend. With Flutter, students can filter and browse job posts and Firebase notifications will alert students when new job openings match their profile.

3.1.4. Application Management Module

In the Application Management module students can apply to job postings and track their application status. Students can apply by submitting their resume or other required documents and recruiters get these applications through a dashboard where they can shortlist or reject candidates. Status updates like 'Applied', 'Shortlisted' or 'Rejected' are sent back to students so they are informed throughout the process. This module is powered by Django Rest Framework to handle real-time status changes and to manage application data securely. Flutter provides a responsive UI for students to view and manage their applications and Firebase sends notifications to students about their application status.

3.3.5. Placement Drives and Scheduling Module

This module helps organize placement drives, schedule interviews, and manage other placement events. Placement officers can create schedules for assessments, interviews, and placement drives, while students and recruiters can register for these events. The system sends reminders and updates through Firebase notifications to all participants. Django Rest Framework provides real-time updates and syncs schedule data across all users, while Flutter displays an intuitive calendar interface on the frontend, showing all scheduled events and allowing students to sign up or recruiters to Figure.4 modify schedules as needed.

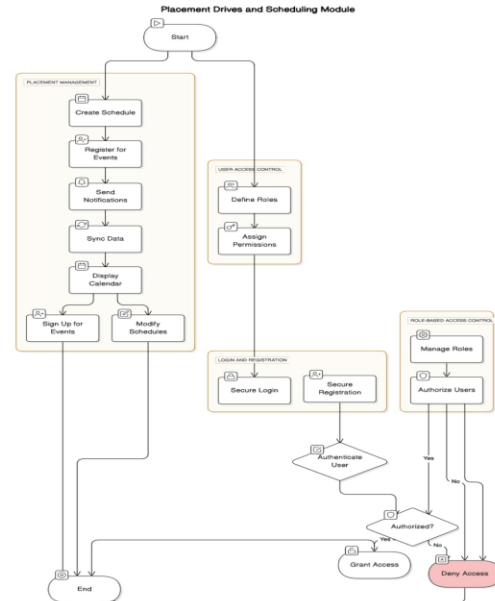


Fig.4 Placement Drives and Scheduling Module

3.3.6. Reports and Analytics Module

The Reports and Analytics module generates detailed reports on placements, job applications, student performance, and recruitment trends. Placement officers and administrators can access insights, such as placement rates, in-demand skills, and company engagement levels. Django Rest Framework processes and compiles data for these reports, which are then visualized using graphs, charts, or tables in the Flutter frontend. This module also allows for data filtering and trend analysis, giving decision-makers actionable insights into the placement processes. Firebase notifications can alert users when new reports are available or when specific metrics reach thresholds that require attention.

3.3.7. Administration and Configuration Module

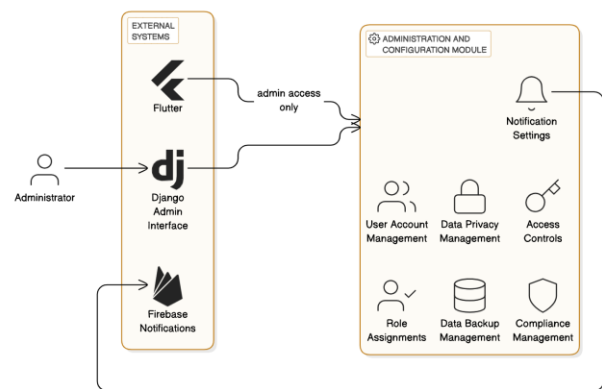


Fig.5 Administration and Configuration Module

The Administration and Configuration module depicted in Figure 3 allows administrators to manage the entire system, including user accounts, data privacy, and notification settings. Admins can configure access controls, manage role assignments, oversee data backup, and ensure system

compliance with relevant regulations. Using Django's built-in admin interface, administrators can efficiently manage the backend without needing extensive custom tools. In Flutter, only users with admin privileges can access configuration settings, and Firebase notifications can be used to alert admins of any critical system updates or user account issues.

3.3.8. Notifications and Alerts Module

This module is critical for keeping all users updated in real-

time. Using Firebase notifications, the system can send custom alerts to users about important events such as application status changes, new job postings, upcoming interviews, and application deadlines. Django Rest Framework handles message queues and event triggers, enabling notifications to be sent based on specific triggers, enabling notifications to be sent based on specific actions or schedules.

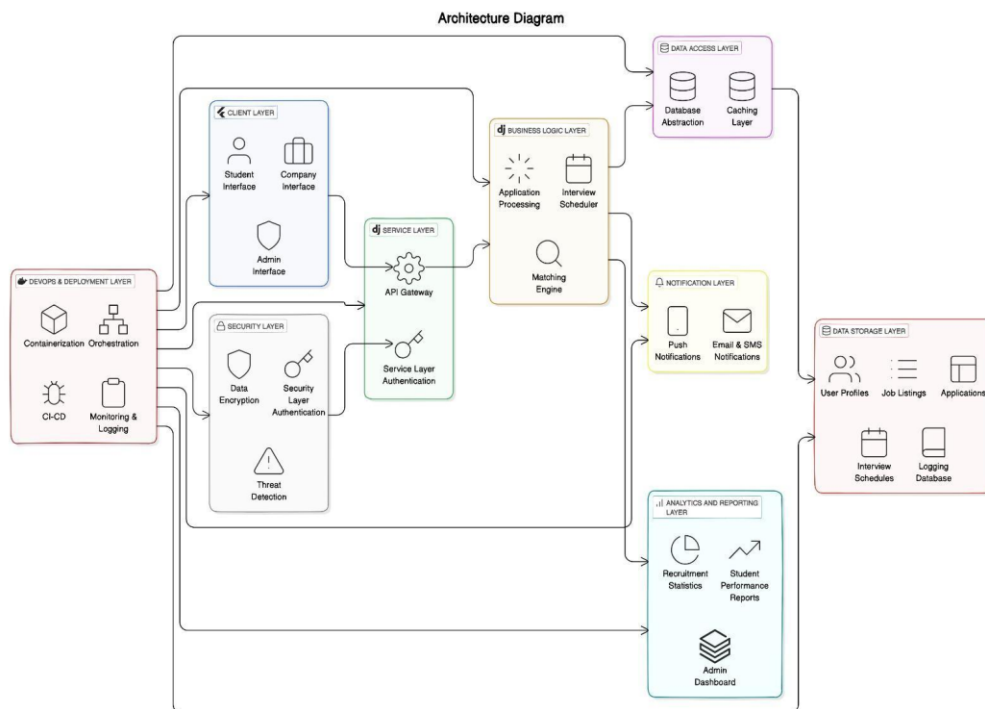


Fig 6 Architecture diagram

In Flutter, users receive these notifications directly on their devices, enhancing engagement and ensuring they stay informed about their placement journey.

3.3.9. Alumni Connect and Tracking Module

The Alumni Connect and Tracking module allows current students to connect with alumni who have completed the placement process and moved into the workforce. Alumni can share their career journeys, provide mentorship, and even recommend job opportunities to students. This module includes an alumni directory, search functionality, and messaging options that let students reach out to specific alumni or filter by industry, company and role. Django Rest Framework manages alumni profile data and interaction tracking, while Flutter provides an easy-to-navigate interface for students to connect with alumni. Firebase notifications can inform alumni of new connection requests or messages, making the experience seamless for all users.

3.3.10. Feedback and Surveys Module

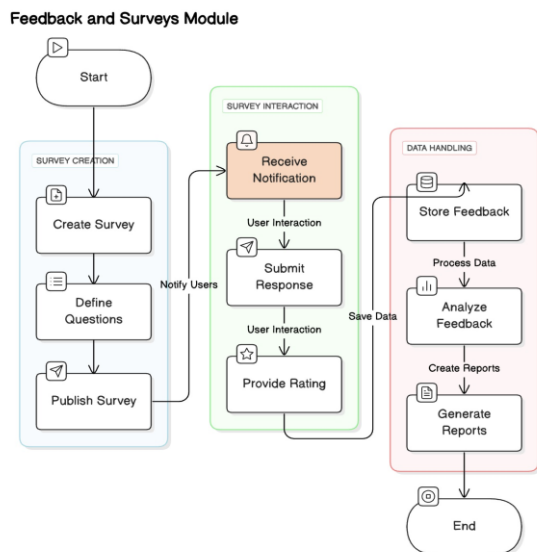


Fig.7 Feedback and Surveys Module

The Feedback and Surveys module is designed to collect feedback from students, recruiters, and alumni on various aspects of the placement process. Placement officers can create surveys to gather insights on system usability, placement experience, and career support services. Students and recruiters can submit responses and provide ratings for the process, helping administrators make informed improvements. Using Django Rest Framework, surveys and feedback are stored and analyzed for reporting purposes. Flutter enables a responsive UI for quick survey interactions, and Firebase notifications can be used to remind users to complete feedback forms or notify them of new surveys.

IV. RESULTS & DISCUSSION

The JobQuench Placement Management System effectively deploys core backend functionality with Django and the Django REST Framework, forming the backbone of an all-encompassing placement platform. Advanced functionality such as secure user authentication, role-based access control, and real-time notifications is provided through Firebase, and Flutter provides a smooth cross-platform experience for students, recruiters, and placement coordinators. With modules for job postings, application tracking, alumni mentorship, and skill development, JobQuench simplifies campus recruitment by eliminating manual processes, decreasing paperwork, and increasing user satisfaction. Early outcomes show increased efficiency, scalability, and user engagement, making the platform a one-stop solution for filling the gap between academia and industry, with future releases expected to include AI-powered resume screening, machine learning-powered job recommendations, and blockchain-enabled credential validation.

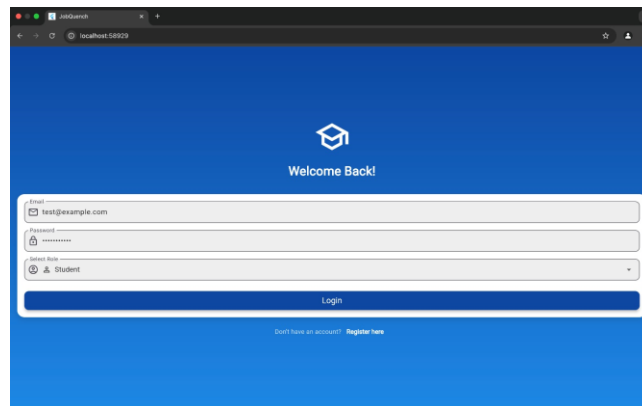


Fig.8 Login Page

This screen serves as the login interface for users of the JobQuench system. It allows users to authenticate using their email and password. The inclusion of a role selection dropdown ensures that different user types (e.g., Student, Recruiter, Placement Officer) can access the appropriate features based on their roles. The registration link facilitates new user sign-ups, ensuring accessibility for all stakeholders.

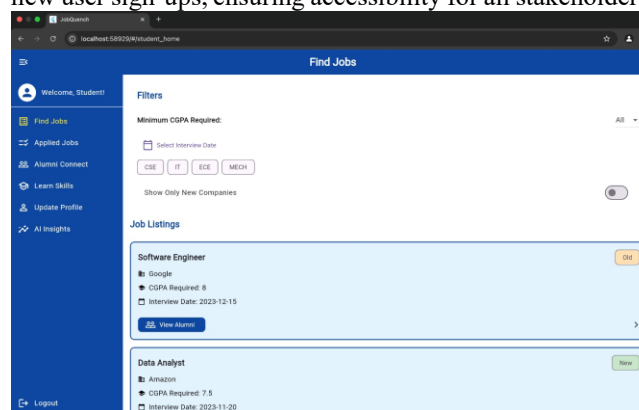


Fig.9 Jobs Page

The Jobs Page provides students with a comprehensive view of available job opportunities. It includes advanced filtering options such as CGPA requirements, department-specific jobs, and interview dates. Each job listing is accompanied by details like company name, required qualifications, and application deadlines. This page also integrates alumni insights, enabling students to connect with alumni who have worked in similar roles or companies.

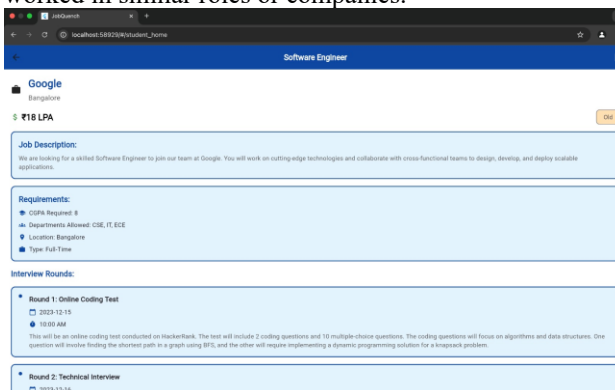


Fig.10 Job Details Page

The Job Details Page offers an in-depth overview of a

specific job posting. It includes information such as the job role, responsibilities, eligibility criteria, and application process. Students can also view related alumni profiles and reviews from previous applicants. This detailed view helps students make informed decisions about applying for jobs.

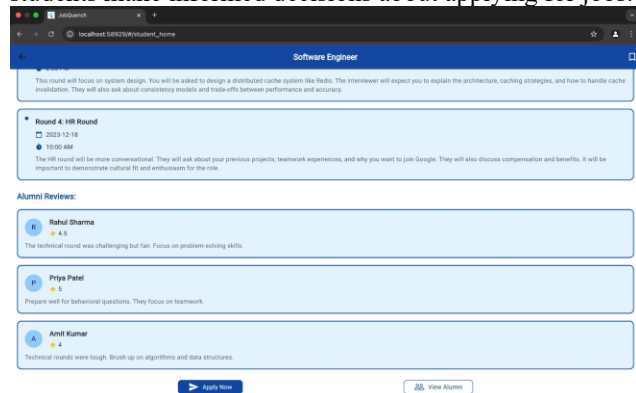


Fig.11 Reviews Page

The Reviews Page displays feedback and ratings provided by students and alumni about specific companies or job roles. It includes star ratings, written reviews, and insights into interview experiences. This feature helps students gauge the reputation of companies and prepare for potential challenges during the recruitment process.

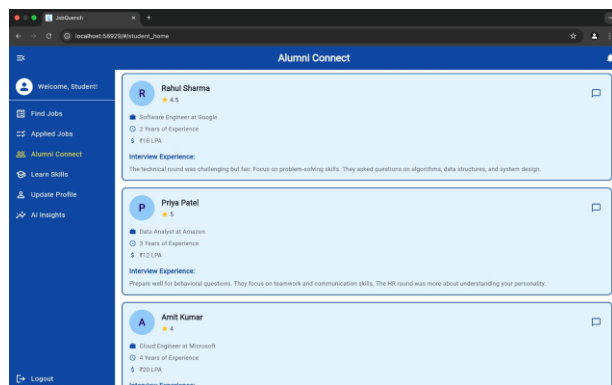


Fig.12 Alumni Connect Page

The Alumni Connect Page enables students to interact with alumni who have successfully navigated the placement process. Each alumni profile includes details such as their current job, experience, salary, and interview tips. Students can filter alumni by industry, company, or role and initiate direct communication for mentorship or career advice.

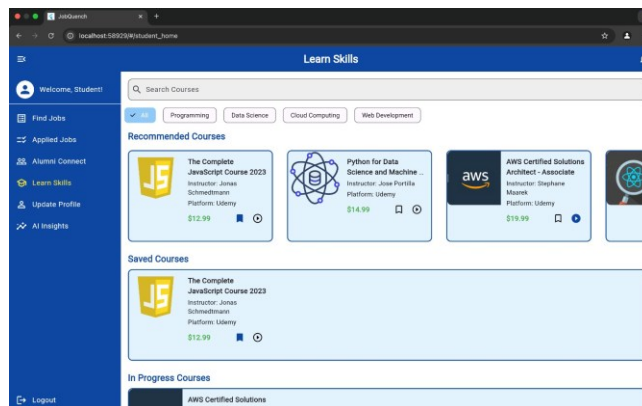


Fig.13 Skills Training Page

The Skills Training Page is designed to help students enhance their technical skills through curated online courses. It categorizes courses into areas like programming, data science, cloud computing, and web development. Students can save courses for future reference, track their progress, and receive recommendations based on their career goals.

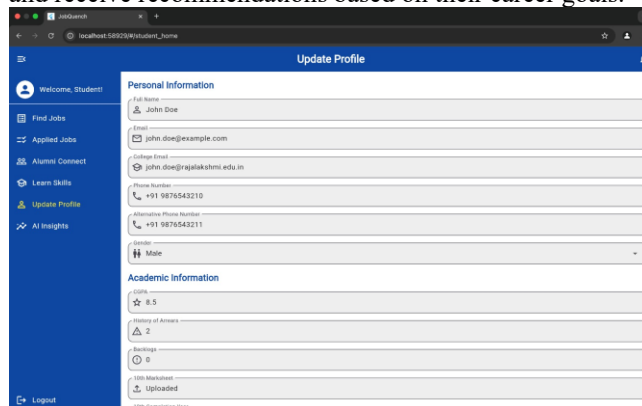


Fig.14 Student Profile Page

The Student Profile Page allows students to manage their personal and professional details. It includes fields for academic qualifications, skills, work experience, and resume uploads. This page ensures that students present a comprehensive profile to recruiters, increasing their chances of being shortlisted for job opportunities.

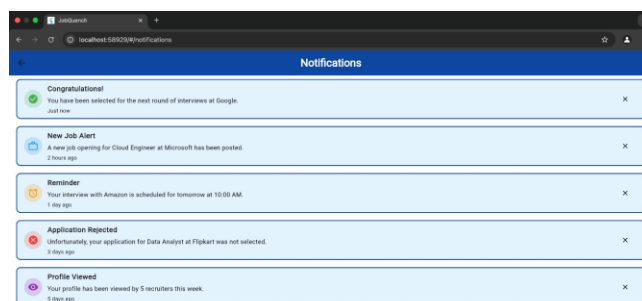


Fig.15 Notifications Page

The Notifications Page keeps users informed about important updates related to the placement process. It includes alerts

about application status changes, new job postings, upcoming interviews, and deadlines. Real-time notifications ensure that students and other stakeholders stay engaged and up-to-date throughout the recruitment journey.

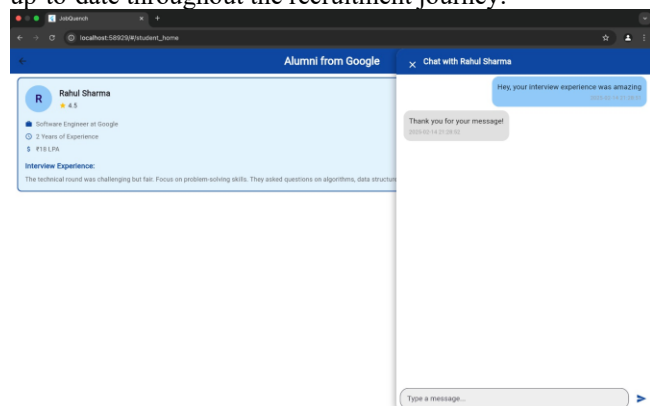


Fig.16 Alumni Chat Page

The Alumni Chat Page facilitates direct communication between students and alumni. It provides a real-time messaging interface where students can seek guidance, ask questions, and gain insights into career paths. The chat history is timestamped, ensuring a seamless and organized interaction experience.

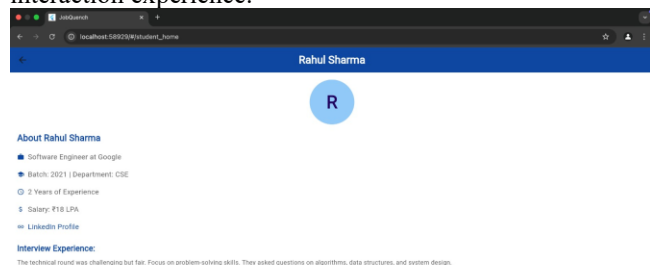


Fig.17 Admin Student Details Page

The Admin Student Details Page is a specialized interface for administrators to manage student data. It includes tools to view, edit, or deactivate student accounts, track application statuses, and oversee placement activities. This module ensures that administrators can efficiently maintain the integrity and functionality of the placement management system.

V. CONCLUSION

JobQuench: An Intelligent Placement Management System efficiently solves the problems of campus hiring by bringing in automation and centralization. Utilizing Flutter for the frontend, Django REST API for backend processing, and SQLite for light data storage, the system increases the efficiency of placement operations. It automates the recruitment process by offering tools such as resume database, job board, and interview scheduling, along with value-added services such as alumni mentorship and career profiling. The initial results of implementation show enhanced user

satisfaction and operational efficiency, saving manual effort and filling the gap between industry and academia. In all, JobQuench enormously streamlines campus placements, improving the recruitment process for students, placement officers, and employers by making it easier, clearer, and more effective.

REFERENCES

- [1] Vandal, K., Kumbar, M., Ullegaddi, P., Angadi, S. and Deshpande, P.K., 2024. Placement Management Android Application. *International Journal of Research in Engineering, Science and Management*, 7(5), pp.101-106.
- [2] Kale, A., Joshi, S., Rajyaguru, T., Borgaonkar, I., Jadhav, C. and Sakunde, P., Enhancing Student Placement with Cross-Platform Application.
- [3] Shanmugam, P., Lenka, J. and Suhail, M.Y., 2024. Enhancing Student Placement Preparation Through Web Application. Available at SSRN 4834698.
- [4] Patila, M.R., Salavia, S.P., Bhisea, S.S. and Duma, A.A., A Development of Web Based Placement Management System.
- [5] Patil, M.S., Kapse, P.R., Mahure, A.Y., Rathod, O.R., Khan, A. and Thakare, V.R., 2023. Implementation of Online College Placement System. *Int. J. of Aquatic Science*, 14(1), pp.242-248.
- [6] Chandra, S.K., Raju, K.S., Raju, P.S., Varma, M.K.S. and Raju, K.L., 2024. Promotion of Graduate Placement Through Academics by Improving Performance Using Artificial Neural Networks. In *Algorithms in Advanced Artificial Intelligence* (pp. 112-117). CRC Press.
- [7] Sharma, N.K., Singh, A.K., Salvi, S. and More, S.S., 2021, August. College kart and k-NN algorithm based placement prediction. In *2021 Second International Conference on Electronics and Sustainable Communication Systems (ICESC)* (pp. 1271-1276). IEEE.
- [8] Banu, A. and SK, D.M.B., 2022. A concise study on Placement Management System. *International Journal of Computer Technique*, 9(2), pp.187-192.
- [9] Vishwasrao, S., Birari, S., Gudelli, T., Sharma, H. and Chavan, S., 2022. A Review on Training and Placement Management Systems.
- [10] Sawai, P.P., Chambhare, P.V., Jaysingpure, A.N., Karhe, A.G., Rathod, D. and Gulhane, V.S., 2024. Alumni Connect Hub: A Comprehensive Alumni Management System. *Journal Impact Factor*, 3(1).
- [11] Yumen, N.M., Alumni Network Platform Leveraging Regression Models for Data Analysis.
- [12] Shanmugam, P., Lenka, J. and Suhail, M.Y., 2024. Enhancing Student Placement Preparation Through Web Application. Available at SSRN 4834698.
- [13] Heck, L., Heck, S., and Sundar, A.S., 2024. mForms: Multimodal Form Filling with Question Answering.
- [14] Stefanova, M., Stefanov, T., and Varbanova, S., The E-Student Profile – Status, Challenges and Perspectives.
- [15] Дуйсебекова, Кулянда Сейтбековна, Роман Хабиров, and Азамат Жолжан. "DJANGO AS SECURE WEB-FRAMEWORK IN PRACTICE." *Вестник КазАТК* 116, no. 1 (2021): 275-281.
- [16] Braathen, Christian. "Interview Scheduling: An Integer Programming Approach" (April 21, 2022). NHH Dept. of Business and Management Science Discussion Paper No. 10, 2022.
- [17] Talluri, Sreelalitha, G. Vaishnavi, G. Naidu, D. Shiva. "Messaging App Using Flutter" (2023). Vats, A., Azim, S., and Chauhan, A.S. "Chat Messenger App Using Flutter," 2023 5th International Conference on Advances in Computing, Communication Control and Networking (ICAC3N), Greater Noida, India, 2023, pp. 1531-1535. doi: 10.1109/ICAC3N60023.2023.10541433.
- [18] Allain, Hugo. "Improving Productivity and Reducing Costs of Mobile App Development with Flutter and Backend-as-a-Service," Aalto University, 2020.

- [19] Olsson, Matilda. "A Comparison of Performance and Looks Between Flutter and Native Applications: When to Prefer Flutter Over Native in Mobile Application Development" (2020). Blekinge Institute of Technology, Faculty of Computing, Department of Software Engineering.
- [20] Stender, Simon, and Åkesson, Hampus. "Cross-platform Framework Comparison: Flutter & React Native" (2020). Blekinge Institute of Technology.