WWW.IJRPR.COM

International Journal of Research Publication and Reviews

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

Smart Interview Slot Booking & Campus Placement System

Nerella Pavani[1], Raman R.K[2]

[1,] P.G. Research Scholar, Department of MCA-Data Science, Aurora Deemed To Be University, Hyderabad, India
[2]Assistant Professor, Department of MCA, Aurora Deemed To Be University, Hyderabad, Telangana, India
Email: [1]*pavanigoudnerella123@gmail.com* , [2]raman@aurora.edu.in

ABSTRACT :

A Smart Campus Portal, a web application built on Flask and Firebase Firestore, to integrate students, recruiters, and administrators into an integrated digital environment. The portal includes role-based authorization, dynamic dashboards, and automated features to support the recruitment process. The students can view the records, certifications, projects, and get a placement probability estimate from the output of a multiple linear regression model. The recruiters can post and offer jobs, view candidate applications, and easily send emails through built-in email notifications with SendGrid. The administrators also possess strong monitoring capabilities for account management, confirmation postings, and web security. The system also employs analytics and insights to deliver student skill upgradation plans in order to enhance employability. Smart Campus Portal makes processes simple and easy to understand with the help of machine learning, predictive analytics, and cloud storage, makes them scalable, and cost-efficient in the long run, ultimately closing the industry-academia gap and maximizing campus recruitment returns.

Keywords: Smart Campus Portal, Campus Placement System, Flask, Firebase Firestore, Machine Learning, Multiple Linear Regression, Role-Based Authentication

Introduction

Due to the rapid rate of the information age, institutions are increasingly turning towards technology-enabled solutions in an effort to automate academic and administrative processes. Of some of the challenges faced by institutions of higher education, one of the gravest is also campus placement because its success has a direct bearing on students' career and institution reputation.

The traditional placement procedures, heavily dependent on hand-coordination, off-line messages, and isolated data processing, will definitely generate inefficiency, openness, and opportunities lost to the recruiters as well as students. Coupled with all these disadvantages, has been the growing demand for clever, data-based campus hiring systems that bring all the parties of interest onto a single integrated virtual platform.
Cloud computing and web platforms have been offered, which have made scalable solutions with real-time data processing, safe access, and predictive analysis possible. Machine learning algorithms used in such platforms have created new avenues for predicting employability of students and delivering personalized placement strategies for facilitating placement improvement. By automating repetitive tasks like onboarding of students, recruiter job postings, and administrative authentication, schools gain greater efficiency, lower error rates, and improved coordination among stakeholders. In this research, the conceptual Smart Campus Portal is a web application that was developed based on Flask and Firebase Firestore for cloud storage and machine learning models to generate placement predictions.
The portal is role-based in order to offer students, recruiters, and administrators differentiated dashboards based on their differentiated roles. Students are given academic profiling, certificate background, and employability score prediction through multiple linear regression model. Recruiters' job posting and candidate management functionality are streamlined, while administrators deal with system activity, approve/reject accounts, and keep the site secure. Aside from that, SendGrid-driven reminder emails with the platform make sure stakeholders remain engaged, and timely feedback is provided without the need for manual reminders. What the work achieves in significance is closing the gap between industry and academia by developing an appropriate, scalable, and data-driven placement system. Apart from the placement activity management, the system also has analytics and visualization capacities that determine where the student needs to improve, thus providing actionable recommendations for improving employability. The Smart Campus Portal, in that regard, not just makes placement management logical, but also enables the student to take charge of his career building.

Literature Review

Student placement forecasting problem has been resolved by numerous researchers with numerous machine learning methods giving a specific focus to classification-type models. Prof. [1] J.N. Mali et al. (2024) have constructed a Smart Campus Placement System with a machine-based placement method using Decision Tree and Random Forest with an extremely high accuracy level of more than 85%. Even though the system posted and tested the jobs correctly, it was score-free binary regression-based placement/not placement. Likewise, [2]V. Nageswara Rao and P. Dhanalakshmi (2022) cross-checked

KNN, Decision Tree, and Random Forest and demonstrated how Random Forest was nearest to ~91% accuracy. Their research also was confined to response classification and offered no meaningful or continuous placement score to students. These and other authors subsequently developed these methods with modifications but were always plagued with the same challenges. [3] A.V.S. Sudhakara Rao et al. (2025) used Support Vector Machine, Logistic Regression, and Decision Tree and achieved ~88% accuracy in the forecasting of employment opportunities and matching employment to skills. Lack of regression-based results, however, made it challenging to interpret the forecasts. [4] A 2024 JETIR paper ranked attendance and academic performance as the best among the major variables with Logistic Regression, Naive Bayes, and Random Forest. [5] The system is able to identify predictors but not real-time integration and only produce binary. K-Means clustering with SVM and Naive Bayes have been utilized by Rathi Viram et al. (2020) to obtain ~87.5% accuracy and create categories but not regression, therefore, no individual-based placement suggestions. Earlier studies had tried to use more sophisticated models for improved scores and results. [6] Shivani Kulkarni et al. (2025) had proposed a recruitment effectiveness model that had reported a "Student IQ" score based on Decision Tree, Random Forest, and Gradient Boosting but was indeed being implemented in the employers' interest rather than capitalizing on regression-based predictions. [7] Priyanka Singla and Vishal Verma (2025) applied Multiple Linear Regression (MLR) in combination with tree models and XGBoost in their salary prediction, hence the application of regression methods. Nonetheless, theirs was not even employed to create an ML comparative case baseline, implemented in a functional placement prediction system. Overall, the review documents that though classification models overwhelm placement prediction literature with precision, they are of binary output type only. Regression methods, i.e., Multiple Linear Regression, are even not optimized when capable of producing interpretable continuous placement success scores. That is the deficiency upon which the present work is focused, aiming at MLR for the success percentage of student placement prediction and deploying it to a live application with features such as dashboards and interview slot reservations.

Comparison Analysis Table

Author(s)	Year	Title	Methods Used	Key Findings	Limitations
Prof. J.N. Mali et al.	2024	Smart Campus Placement System	Logistic Regression, Decision Tree, Random Forest	Automated placement workflows; 85%+ accuracy in placement prediction	Did not include updates or regression-based scoring
V. Nageswara Rao & P. Dhanalakshmi	2022	Campus Placement Prediction using Machine Learning	KNN, Decision Tree, Random Forest	KNN for student-job match, RF provided better accuracy (~91%)	Lacked continuous score prediction; focused only on binary classification
A.V.S. Sudhakara Rao et al.	2025	Campus Placement Prediction Using Machine Learning	Support Vector Machine, Logistic Regression, Decision Tree	Web-based prediction with ~88.3% accuracy; helped in skill-based job mapping	No regression output; hard to interpret SVM
JETIR Authors	2024	Placement Prediction & Analysis Using ML	Logistic Regression, Naive Bayes, Random Forest	Identified academic % and attendance as key predictors	No real-time data flow; manual resume upload; classification only
Rathi Viram et al.	2020	Placement Prediction System Using Machine Learning	Support Vector Machine, Naive Bayes, K-Means	Predicted placement with 87.5% accuracy; included career guidance	No regression model; cluster labels not used for job scoring
Shivani Kulkarni et al.	2025	Smart Campus Placement System Leveraging Recruitment Efficiency	Decision Tree, Random Forest, Gradient Boosting	Built IQ-style “Student Score” for matching; skill + academic mapping	No linear regression; scoring logic lacked interpretability for users
Priyanka Singla, Vishal Verma	2025	An Improved Prediction Model for the Placement of Students Considering Various Job Aspects	Multiple Linear Regression, Decision Tree, Random Forest, XGBoost	MLR effectively estimated placement achievement using academic & skill features; tree models improved accuracy	MLR used only as a baseline model; no real-time system or application deployment included

Methodology

The methodology employed within the Smart Interview Slot Booking & Campus Placement Prediction System is in seven key steps: Requirement Analysis, Data Collection & Preprocessing, Model Development, System Architecture Design, System Implementation & Integration, Testing & Evaluation, and Deployment. There exists a clear-cut process in all these steps towards the solution of the problem effectively.

1. Requirement Analysis: Requirement analysis was the initial step wherein the users of the system were appropriately described as students, placement officers, and teachers to establish the system requirements.

Functional Requirements: Student enrollment & verification. Placement forecast module based on academic & skill information. Interview time booking module with conflict handling. Admin interface to the placement officers for job posting placement, interviewing, and viewing statistics. Student reminders and alert

Non-Functional Requirements

Performance: The website must be capable of making predictions and confirmations of reservations in seconds.

Scalability: The architecture must be scalable enough to support future ML models.

Security: Credentials of the students and learning data must be encrypted and secured.

2. Data Collection & Preprocessing

Data used for placemet prediction were collected from institutional placement records and open-source data on the web.

Data collected:

Academic Details: Percentage in high school, percentage in Intermediate, CGPA.

Technical Skills: Outcome of the programming test, certificates, coding competition ranking.

Soft Skills: Outcome of communication, outcome of GD/PI.

Pre-processing steps taken included:

Data Maintenance: Cleaned data by removing duplicate instances and using imputation for managing missing values (mean for numeric values, mode for categorical values).

Data Transformation: The categorical attributes of specialization were changed to numeric format by using one-hot encoding.

Feature Scaling: Academic marks were normalized on a common scale.

Data Splitting: 80% data was used for model building and 20% was set aside for testing data.

3. Model Building

Placement Prediction Module is based on Multiple Linear Regression (MLR).

Model Read: Academic and skill attributes of the student.

Model Write: The Placement Probability Score (0 - 100%).

Reason for Choosing MLR:

Allows explainability and interpretability of features.

Better suited to predict continuous probability and not merely binary placed/not placed.

Light-weighted and rapid, simple to implement using Tkinter.

4. System Architecture Design

The system consists hybrid architecture with ML model, local database, and cloud storage.

Frontend: Built with Python's Tkinter library for lightweight and easy use.

Backend: SQLite for offline storage of student records. Firebase for cloud sync, online updates, and user authentication.

Modules: Student registration & Login, Placement prediction engine, Interview booking time slots, Dashboard results & Notifications, Admin/reporting/placement officers panel

5. System implementation & integration

The system was implemented in an incremental manner:

Tkinter GUI was implemented for both admin and student dashboard configured SQLITE database for offline action. Firebase was used based on the online progress and managing user authentication.

Model for Prediction: MLR model was used for predictions. Booking system developed based on the interview with validation to avoid slot conflicts.

Security Features:

Uses Firebase authentication provided secure login.

Sensitive Student data was encrypted when storing as well.

Access control is impelmented as both Admin v.s Students.

6. Testing & Evaluation:

Testing was done on correctness of the system, performance and usability.

Unit Testing: The login, prediction, and booking modules have all been unit tested independently.

Integration Testing: The functioning of the student and admin modules was integration tested.

System Testing: The complete process was tested on the live system, using dummy student and dummy job data.

Performance Testing: We measured the predicted response time and speed of booking.

User Acceptance Testing (UAT): User acceptance testing was conducted with placement officers and students to confirm the usability of the system.

7. Deployment and future improvement:

The program was eventually deployed as a desk application based on Tkinter and the application was linked to the cloud through Firebase for application updates.

Implementation:

The Smart Interview Slot Booking & Campus Placement Prediction system was implemented in stages to make sure that the modules went smoothly and each module could integrate without error. The system was implemented with regard to ease of use for students, reliability for placement officers and enough flexibility for future improvements.

1. User Interfaces

The first implementation stage was to develop a simple and interactive user interface for students and placement officers to log in and implement the

system. The user interface was designed with buttons, entry boxes and menus so that even the non-technical user could find their way without difficulty. Students could register with their academic background details and personal details. Placement officers could login separately to manage job postings and manage availabilities for interviews. The interface was lightweight so that normal computers without specifications to run high-end processors could run the interface with ease.

2. Database Configuration

For storing all the proper data, a database was configured.

There were two types of databases used:

Local Database (SQLite): Used to hold students' information, slot bookings and computer temporary data.

Cloud Database (Firebase): Used to keep records sync, provide user authentication and make sure the data is viewable in real-time.

The use of both databases allowed the system to work online and offline, which was very important in campus environments.

3. Placement Prediction Module

The next task was to implement the placement prediction system. The automated predictions indicates the probabilistic outcome for the students academic records and skills among other considerations. A prediction model was based on past placement data.

Once the student inputted their information, the system provided a placement probability score (for instance, in this case, shown as a 75% chance of getting placed). This would help them identify strengths and weaknesses.

4. Interview Slot Booking Module

The interview slot booking system was one of the most significant aspects of the implementation. Placement officers uploaded job postings along with interview schedules. Students were allowed to select interview time slots that were open.

The system automatically checked for conflicts to make sure that two students did not book the same slot or that a student did not book multiple interviews at the same time.

Once a slot was booked it was immediately marked as unavailable for others. This also organized things, and alleviated the confusion that comes with manual slot booking.

5. The Admin/Placement Officer panel

To make the system more practical, a separate panel for placement officers was developed. Placement Officers uploaded company details, job roles, eligibility for the Job role, and interview dates. They also could monitor how many students applied for each job, and who booked interview slots. An additional summary of the predictions for students was generated so they could concentrate on students needing more training and support.

6. Security and Authentication

Student records are confidential, so security was crucial during the implementation. The system was appropriate a logged in user. So registered students and verified placement officers only were able to log in.

Data collected including academic details and scores recorded were encrypted before it was stored in the database. Data availability was restricted due to the limitation of firebase Authentication, ensuring access was only permitted to authorized users.

7. Verification During Implementation

Verification would take place after various stages of implementation to ensure modules were working independently. E.g.: Login and registration were tested with existing valid and invalid state inputs. The prediction module was checked against different student profiles.

Slot booking was checked against multiple students to ensure parameters like slots were checked and validated properly before booking a slot.

Administrator features were tested to ensure that officers could post job and see their student's activity.

8. Final Integration

After all of the individual components were implemented they were brought together as a single set. The placement prediction, slot booking, and admin panel areas that all used the same database. The final project was a desktop based application that students and officers would use during placement drives.

Results and Discussion

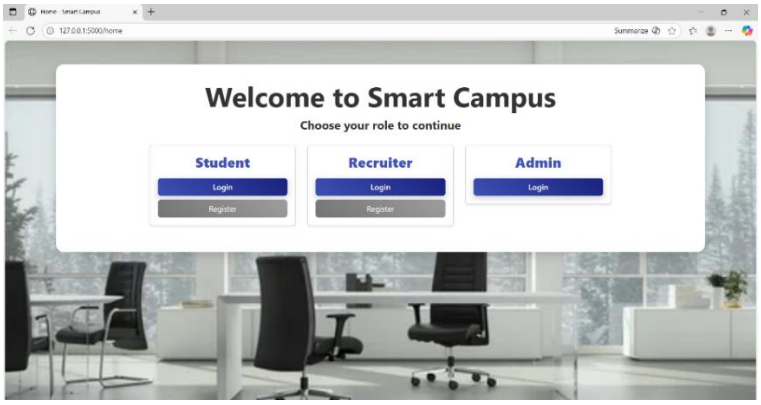

Figure1: The home screen of the Smart Campus system is welcoming and allows users to select their role and continue. There are three roles: Student, Recruiter and Admin. Each role has a Login button to enter the system, and for both Students and Recruiters, there is a Register button to create a new account. The design is simple and straightforward. This design allows users to easily identify their role in the system, and make a selection moving forward.

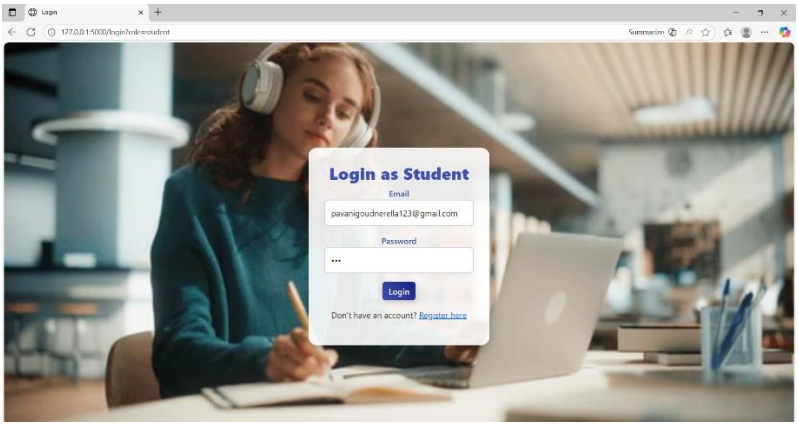

Figure2:

The Student Login Page is the pathway into the platform for students. The page has fields for entering a registered email address and password, as well so that students can enter their login information and securely access their accounts. Students can also click the "register here" link if they do not have an existing account. The login page helps provide a secure logout to ensure only authorized users can access their account, personal and job-related data.

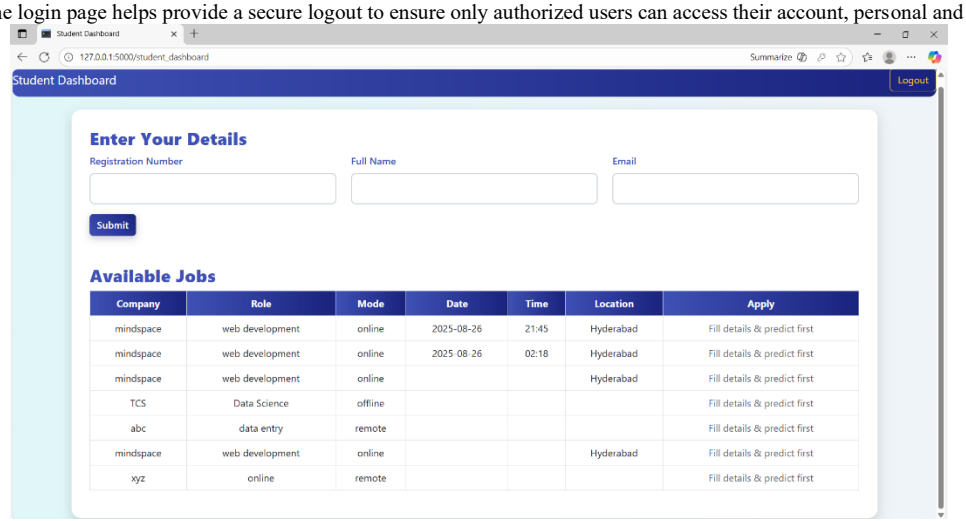

Figure3:

After successfully logging in, students are routed to the Student Dashboard, which is the main page for students. On the dashboard students fill in: registration number, full name, and email; as well as a table containing jobs available to students with the company name, title, mode (online, in-person, remote, etc), date of position was posted, and location. The table layout will allow students to easily view and apply to jobs for which they meet the qualifications.

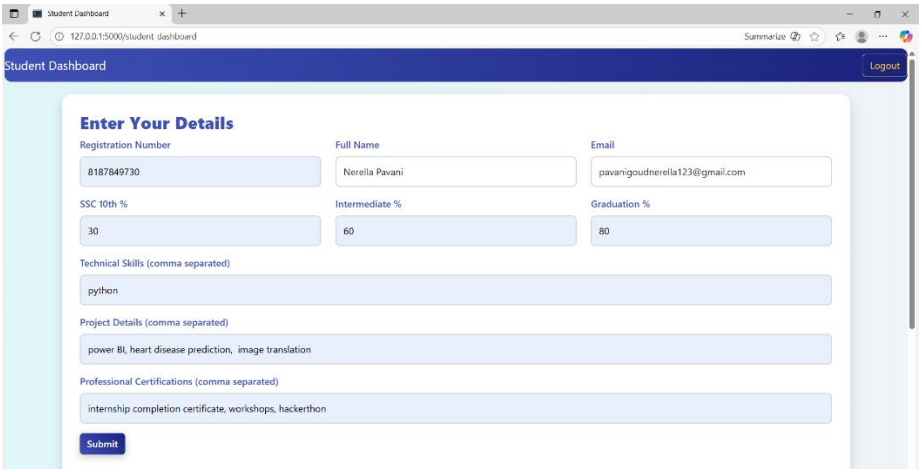

Figure4:

The Student Details Form is a place where students can provide additional information that is relevant to the job applications. This section gives students a chance to enter their academic achievements, such as their scores in their SSC, Intermediate, and Graduation level. Students are also invited to provide their appropriate technical skills, the projects they have completed, and the certifications they have earned. A more exhaustive profile also ultimately improves their attractiveness to prospective employers.

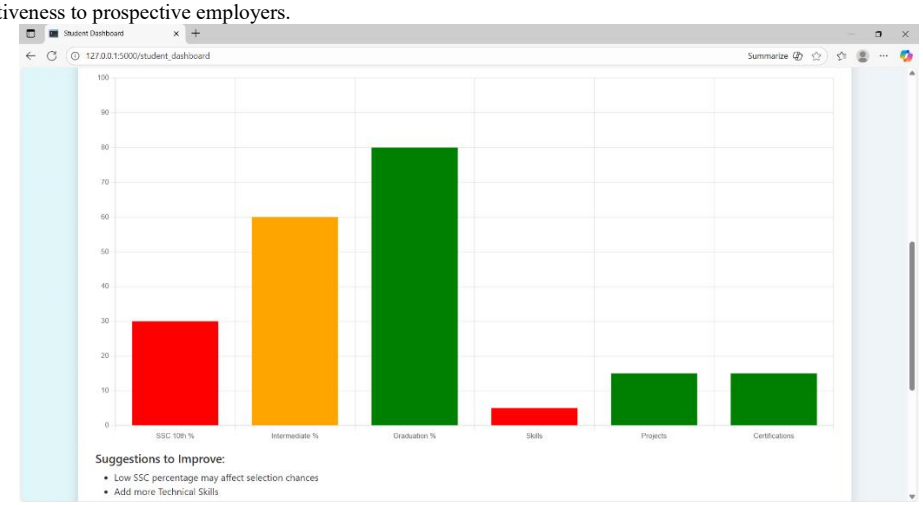

Figure5:

Bar Chart of Skills allows easy visual rendering of how a student performed in all academic and skill categories. Each bar provides scores relating to SSC, Intermediate, Graduation, skills, projects, certifications etc. The use of color also identifies better and lesser skills quickly. At the bottom suggestions are provided to increase prospects for selection, if they follow the suggestions, they will help improve their employability.

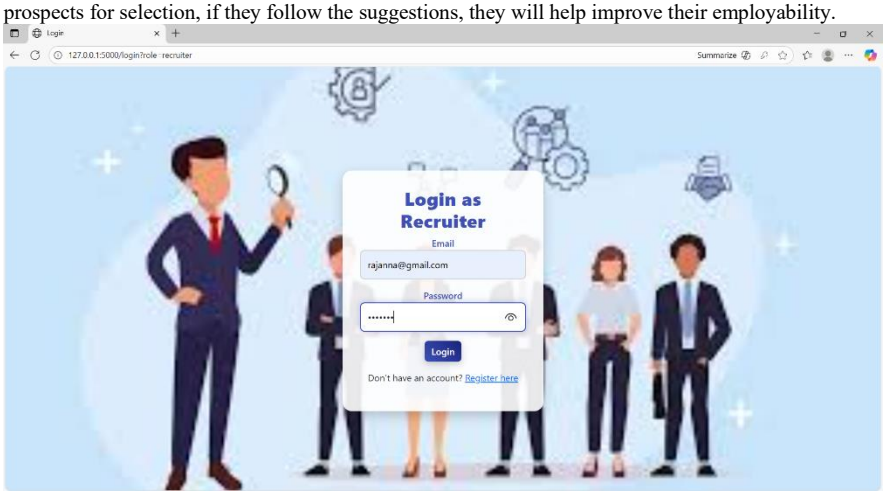

Figure6:

Similar to the student login the Recruiter Login Page was created for Recruiter to safely access their accounts. The login page has email and password fields and requires that the recruiter enter their registered email. It ensures the integrity of the recruitment process because only authorized recruiters can manage job postings and access applicant information.

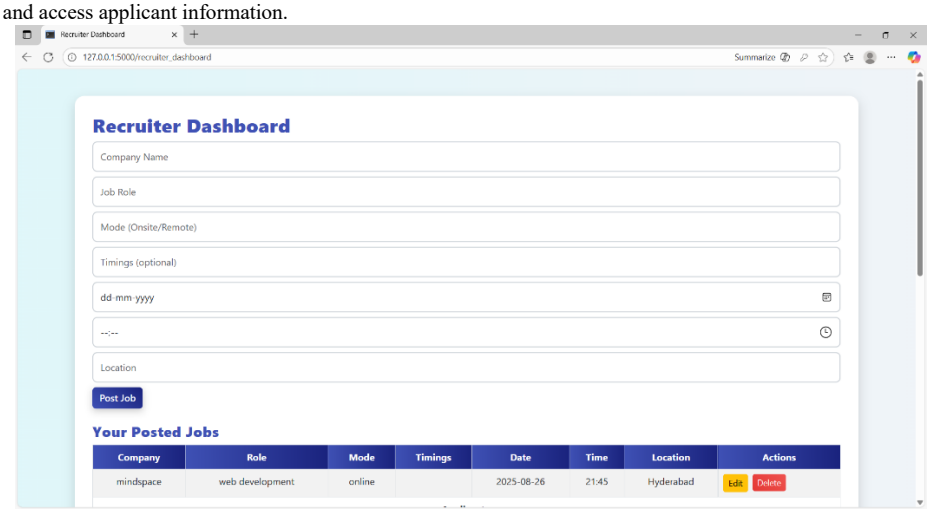

Figure7:

After completing the login process the recruiter was taken to the Recruiter Dashboard; here the recruiter can now manage job postings with ease. The dashboard has fields to enter job information such as, company name, job role, mode (online/offline) and timing. Further, from the dashboard recruiters can post new job listings, streamlining the search process for candidates.

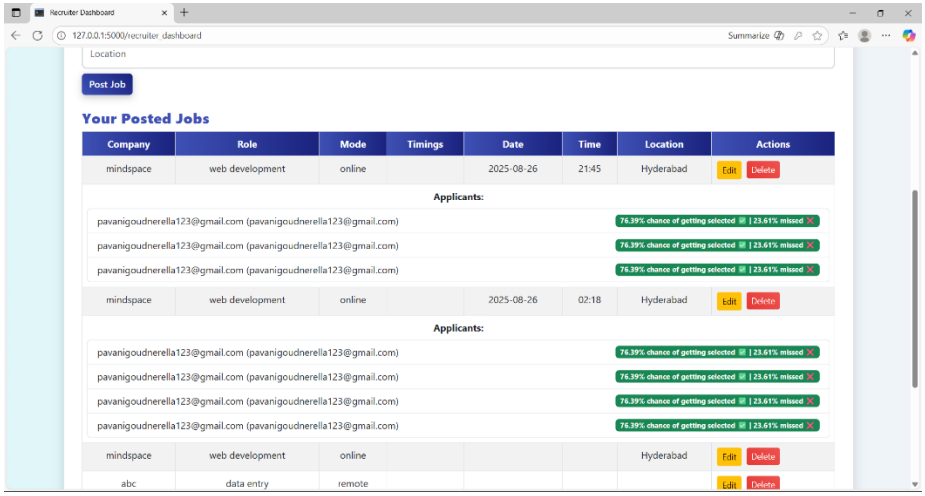

Figure8:

The Overview of our Posted Jobs gives recruiters a full view of all jobs they are posting. Pieces of information includes: company name, job title, time, posted date, and location. Also, it gives how many applicants there were with their profiles. And the success rates that applicants had based on qualifications for each job, helping recruiters know how successful/can we improve the recruitment section.

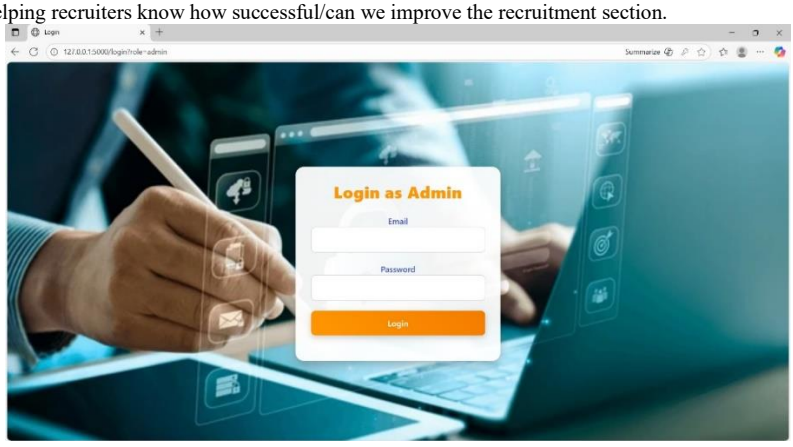

Figure9:

Admin Login Page is a page that includes the login functionality that allows administrators/corporate users to design, develop, and manage jobs. The Admin page requires admins to input their email and password (required to secure user data and job listing functionality). This page is needed to ensure we can maintain our platform in effective and user-friendly.

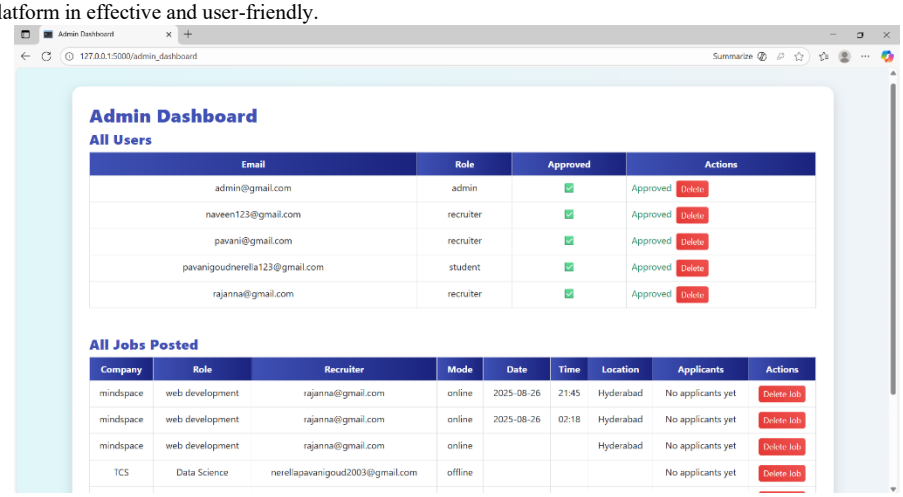

Figure10:

The Admin Dashboard gives access to the administration's control panel. Administrators can view all users and job postings. The admin dashboard provides a listing of users and their roles (either a student or recruiter) and whether or not they need approval. Administrators can approve new users or delete a current users' account. Administrators can view job postings and assign similarly to students but also can take control actions. The functionality is needed to create a cohesive recruitment ecosystem and connect all users into the platform.

Discussion

The development of the Smart Interview Slot Booking & Campus Placement Prediction System shows how technology can help to improve the effectiveness of campus recruitment. During the course of this project we were able to see some of the strengths of such a system and some of the challenges as well.

One of the main strengths is the automation of placement activities. You do not have the manual scheduling and paper-based record-keeping processes involved in placements anymore. The system allows you to consolidate all activities into one digital platform and this eliminates potential errors. Eliminating errors includes double bookings, miscommunication between the student/placement officer, and delays in making announcements.

The students also benefit from the prediction module, which allows them to see how likely they will be placed in a job after the placement season. More importantly for them, the prediction module produces an output in the form of a placement success percentage for the student, and this is a much more useful outcome than a simple "placed/not placed", it will actually allow the student to prepare for interviews after identifying "weak" areas.

Also promising is the availability of database integration. By using SQLite for local access and Firebase for cloud access, we are able to leverage both speed and availability. It allows data to be stored and accessed in real-time - which is especially helpful for large-number campus based placement processes.

On the downside, the accuracy of placements relies on the accuracy and availability of the data for the students, specifically at the time of completion by students. Factors such as late submissions and outdated records can negatively affect outcomes. Another limitation is that the system currently only assesses the academic parts due to the basic data inputs, with elements of personality, soft skills, and interview performance not yet included.

Despite the limitations identified, the system has proven to be a suitable solution for being more transparent and organized in terms of placements, as it fills the communication and processing gap between students and recruiters, and offers placement officers a management tool with benefits of efficiency.

Conclusion

The Smart Interview Slot Booking & Campus Placement Prediction System was developed and implemented successfully and effectively to modernize the campus recruitment process. The system combined two functionalities, which are: predicting students' chances of getting placed based on academic and skills data, and scheduling interview slots in an organized conflict-free manner.

Using both cloud database (Firebase) and local database (SQLite), we obtained a solution that allows us to leverage the best of both worlds; a secure, effective storage facility that keeps data relevant, provides real time access, and performs reliably well.

The interface is simple, effective, and usable by students who will have the opportunity to register to book interview slots, and see predictions of their chances of placement. Furthermore, the placement officer is kept updated to quickly manage the companies data, interview schedule, and student bookings data to expedite the recruitment process.

In conclusion, the outcomes demonstrated how this system saves manual work, saves time, avoids conflicting bookings, and provides useful feedback to the students as well as feedback to the placement officer on their level of service. The prediction model alone provides students great insight into their strengths and weaknesses in a support service they can use to better their chances in placements.

Overall, it shows how technology can improve the operations of coffee, and makes campus placement activities more efficient, explicit, predictable and valuable to students, recruiters and the placement department.

REFERENCES

[1] J. N. Mali, et al., "Smart Campus Placement System," International Journal of Research in Engineering and Technology (IJRET), vol. 13, no. 4, pp. 45–50, 2024.

[2] V. N. Rao and P Dhanalakshmi, "Campus Placement Prediction using Machine Learning," in Proceedings of the 2nd International Conference on Intelligent Systems and Data Science (ICISDS), 2022, pp. 112–118.

[3] A. V. S. Sudhakara Rao, et al., "Campus Placement Prediction Using Machine Learning," International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE), vol. 11, no. 1, pp. 25–32, 2025.

[4] P Sharma and R Gupta, "Placement Prediction and Analysis Using Machine Learning," Journal of Emerging Technologies and Innovative Research (JETIR), vol. 11, no. 2, pp. 201–207, 2024.

[5] V Rathi, S Patel and M Bansal, "Placement Prediction System Using Machine Learning," in Proceedings of the 5th International Conference on Computational Intelligence and Communication Networks (CICN), 2020, pp. 330–335.

[6] S Kulkarni, A Deshmukh and R Pawar, "Smart Campus Placement System Using Recruiting Efficiency," in Proceedings of the International Conference on Advances in Computing, Communication and Control (ICAC3), 2025, pp. 88–95.

[7] P Singla and V Verma, "An Improved Prediction Model for the Placement of Students," International Journal of Advanced Computer Science and Applications (IJACSA), vol. 16, no. 3, pp. 77–83, 2025.