FEDERAL UNIVERSITY OF TECHNOLOGY, AKURE
JOURNAL OF ENGINEERING AND ENGINEERING TECHNOLOGY

INTERNLY: A MOBILE PLATFORM FOR CONNECTING NIGERIAN STUDENTS TO INTERNSHIP OPPORTUNITIES

[1]Akinlabi, A. A.[*], [2]Faluyi, A. P., [3]Iberedem, J. M. and [4]Dahunsi, F. M.

[1,2,3,4]Department of Computer Engineering, School of Electrical Systems Engineering,

Federal University of Technology, Akure, Ondo State, Nigeria

**Corresponding author: aaakinlabi@futa.edu.ng*

Akinlabi, A. A., Faluyi, A. P., Iberedem, J. M. and Dahunsi, F. M (2026): Internly: A Mobile Platform for Connecting Nigerian Students to Internship Opportunities. FUTA Journal of Engineering and Engineering Technology 20(special), 113-123

Received Date: 15.01.2026 **Accepted Date: 10.03.2026**

Abstract
Finding the right internships remains a significant challenge for undergraduate students in Nigeria due to the lack of effective filtering and matching mechanisms on existing platforms like job boards, LinkedIn, and Indeed. These platforms do not take into consideration factors such as skill competency, experience or alignment with professional goals because they rely on simple keyword searches or manual filters including location and job title. INTERNLY is a mobile platform that bridges the gap between students seeking internships and organizations offering these opportunities. The platform's front-end component was built using React Native, while Node.js, Express, and MongoDB served as the back-end technologies. Cloudinary, a cloud media storage managed image and video uploads. INTERNLY utilizes rule-based matching algorithms and centralized management tools to streamline the placement process. Prospective Interns are matched to internship opportunities using a scoring system that prioritizes their skills and interests and employer requirements like minimum Cumulative Grade Point Average (CGPA), skills or experience. Students can create profiles, browse tailored internship listings and apply for relevant roles, while organizations can register, post openings and manage applications efficiently. A structured database supports user information storage, and filtering algorithms generate personalized recommendations. Security measures that were put in place include encrypted authentication and data protection. Initial testing of the INTERNLY prototype demonstrates improved efficiency in internship placement. The platform reduces the time and effort required by all stakeholders, increasing accessibility and engagement.

Keywords*: Internship opportunity, mobile app, tertiary education, career development, rule based matching algorithm.*

Introduction

Internships are essential for students in tertiary institutions in Nigeria, as they serve as a bridge between academic learning and industry experience. Students who become interns during their mandatory 3 to 6 months Student Industrial Work Experience Program (SIWES) get on-the-job training in organizations relevant to their respective majors, thereby preparing them for future employment (Ibrahim *et al.*, 2021). Industrial training is aimed at enhancing the employability of the students and making them ready for future careers by instilling in them the necessary competencies and confidence to cope with work-related demands in a professional manner (Hendrawan *et al*., 2024). But the process of pairing the students with the appropriate industry or organization has proved to be increasingly challenging with the high number of applicants and the absence of the tailor-made systems which are critical in matching the students to the best organization. Currently, organizations rely on individual placement letters issued to students by their tertiary institutions as well as referrals, since they do not have direct access to students' academic profiles. The prospective

interns on the other hand face difficulties in identifying organizations that align with their field of study (Ibrahim *et al.*, 2021). Quite a number of these students waste valuable time that should have been spent in acquiring skills in search of internship placements. The aforementioned challenges pose the need to develop an efficient digital platform that can seamlessly bridge the gap between Nigerian students seeking internships and organizations offering these opportunities. This digital solution streamlines the matching process and ensures that students are placed in organizations where they can gain relevant experience. INTERNLY is a mobile application that utilizes rule-based matching algorithms and centralized management tools to streamline and speed up the placement process. Prospective Interns are matched to internship opportunities using a scoring system that prioritizes their skills and interests and employer requirements like minimum Cumulative Grade Point Average (CGPA), skills, or experience. Students can create profiles, browse tailored internship listings, and apply for relevant roles, while organizations can register, post openings, and manage applications efficiently. A structured database supports user information storage, and filtering algorithms generate personalized automated recommendations. Additionally, more features like application tracking and direct communication channels between students and employers are incorporated into INTERNLY application. Besides, the widespread use of smartphones amongst students presents a unique opportunity to enhance internship accessibility through mobile applications (Tyagi *et al.*, 2023). INTERNLY will allow Nigerian students to stay informed and updated on internship opportunities, reducing the likelihood of missing critical and appropriate placements. Because the UI is simple and the search filters are efficient, students can rapidly find and apply for internships that fit their academic and professional goals. When stakeholders start to use this digital solution for the internship placement process, both institutions and students can improve career readiness, making the transition from education to employment seamless and more efficient. This platform will be good for students and will also strengthen links among academic institutions and organizations, so interns get the right training and experience.

Literature Review

Recent studies have revealed applications and systems developed to assist students in finding internship opportunities through web and mobile technologies, intelligent matching algorithms, and increased focus on student-centered design. Hartley and MacAllum (2018) discussed an implementation that proposed a conceptual model for managing student-sponsor expectations. There was considerable attention on the complexities of the matching process of skills, goals, and the social connectedness that a visual map of expectations offers a student. However, no practical or system-based option was provided as a technical implementation in their research. Chand and Deshmukh (2019) highlighted the importance of internships in making students more employable, particularly in underdeveloped countries where the likelihood of receiving high-quality education, as well as employment opportunities, may be scarce. According to these authors, a properly designed information system would assist students in traversing the. strenuous task of searching internships that suit their academic backgrounds and professional. goals. Saxena *et al*., (2020) on their part emphasized the need for an internship finder application to streamline the hectic search process. Ibrahim *et al*., (2021) proposed a fuzzy matching algorithm to establish a mapping from academic credentials to an internship position profile. While the fuzzy matching algorithm has been effective, researchers cited a need for employers to have a much more inclusive database of industry partners to support long-term engagement experiences and also use it to develop institutional strategy. Tyagi *et al*., (2023) introduced an Android-based recruitment assistance application specifically designed for engineering students, providing a user-friendly environment for internship placement-related activities. Abubakar *et al*., (2023) described “Li2U”, a mobile app that connects students with universities with suitable internship placement options. While the app provides user-friendly search and placement processes, it has weak privacy responses that raise questions of sensitive user data exposure. Navghare (2024) described a college department's internship portal, emphasizing the importance of student engagement and satisfaction. Hendrawan *et al*., (2024) presented a web-based information system that centralizes internship listings, offering features like personalized profiles and customizable search filters. Overall, the system utilized intuitive search, filtering, and easier management of applications and

placements but does not provide for post-internship evaluation or feedback loops for institutions. Faridiansyah (2024) developed a web-based monitoring system using Rapid Application Development Framework (RAD). It showed improvements in placement efficiencies and data business processes, as well as the required data; the study highlighted the use of hybrid platforms and the need to integrate web-based and mobile applications to meet student demands for real-time needs. Research into intelligent matching algorithms has also emerged to match student profile requirements to employer profiles. Mahendru *et al*. (2024) launched "Campus Ease," a mobile app that provides job alerts and application tracking. While being simple and user-friendly, it still does not offer any alumni mentoring or bring resources together to provide for all individuals seeking career development in one place. Larger platforms also strive to connect various parties.

Summary of Gaps in Existing Studies

Previous studies have shown that current internship management systems are inadequate, especially when it comes to balancing technical functionality with designs that truly meet users' needs. However, these earlier studies place greater emphasis on technical features while overlooking the value of continuous feedback from key stakeholders, which is essential for improving usability for students and supervisors. In addition, the absence of multi-functional platforms that bring together resources, alumni mentorship, and career support limits students access to well-coordinated assistance. Privacy also remains a concern, as existing approaches often do not adequately safeguard sensitive user information. There are currently no hybrid solutions that can successfully combine the effectiveness of a web-based administration and the convenience of mobile access to increase the interest of users, and research on deployment methods is not extensive. Moreover, limited post-internship support has also been noted with institutions in most instances lacking centralized records of previous internship placements and employer partnerships which makes it hard to derive trends or make the most out of upcoming placements. The lack of feedback mechanisms between firms, alumni, and students hinders iterative adjustments to curricula or matching procedures, even though alumni insights are still underutilized. Lastly, there is the tendency of overlooking post-internship support since it lacks official frameworks to evaluate the internship performance and guidance once it is over. Collectively, these highlighted deficiencies point to a clear need of developing secure, flexible and integrated systems that address the whole internship management lifecycle, from placement and preparation to post-internship evaluation and institutional learning.

Methodology

Requirement gathering

To properly find out the desired features for the proposed internship matching platform (INTERNLY), a survey was conducted among potential beneficiaries of the platform, which includes students, lecturers, and business owners/organizations. The survey was designed to gain insight into the following questions:

i. Is a dedicated internship matching platform really necessary?
ii. Why would anyone want to use it?
iii. Does it solve the problem statement or add to it?
iv. What difference does it make to a user?

The responses from the survey indicated discontent in the traditional job posting sites and a growing demand for a dedicated internship placement platform.

Development Approach

The INTERNLY platform was developed using the Agile software development methodology. Agile was chosen to ensure iterative progress, continuous feedback, and quick adaptability to changes during the software development lifecycle (Pressman and Maxim, 2015). The project was divided into multiple sprints, each focusing on incremental features, bug fixes, and performance enhancements. Git was used for sprint planning, issue tracking, and version control.

System design and architecture

The functional requirements in this stage of the project are converted into detailed technical designs that support scalability, efficiency, and long-term maintainability. To guide this process, the use-case diagram in Figure 1 was employed to illustrate how different users interact with the system. The main actors identified are the student (primary actor 1), the organization (primary actor 2), and the administrator (secondary actor).

Technology Stack

INTERNLY was built using the popular MongoDB, Express, React and Node (MERN) architecture (Kadam *et al*., 2023) shown in Figure 2, prioritizing high code quality, scalability, performance and clear separation of concerns. The front-end (React Native) and back-end (Node/Express) are decoupled via a RESTful API. This aligns with REST principles which provide a lightweight way to build web APIs that connect components in a scalable, microservices-like architecture. In particular,

statelessness in the API was enforced as each client request includes all needed information, and no session state is stored on the server. This decoupling improves maintainability. The major components of the INTERNLY system are as follows:

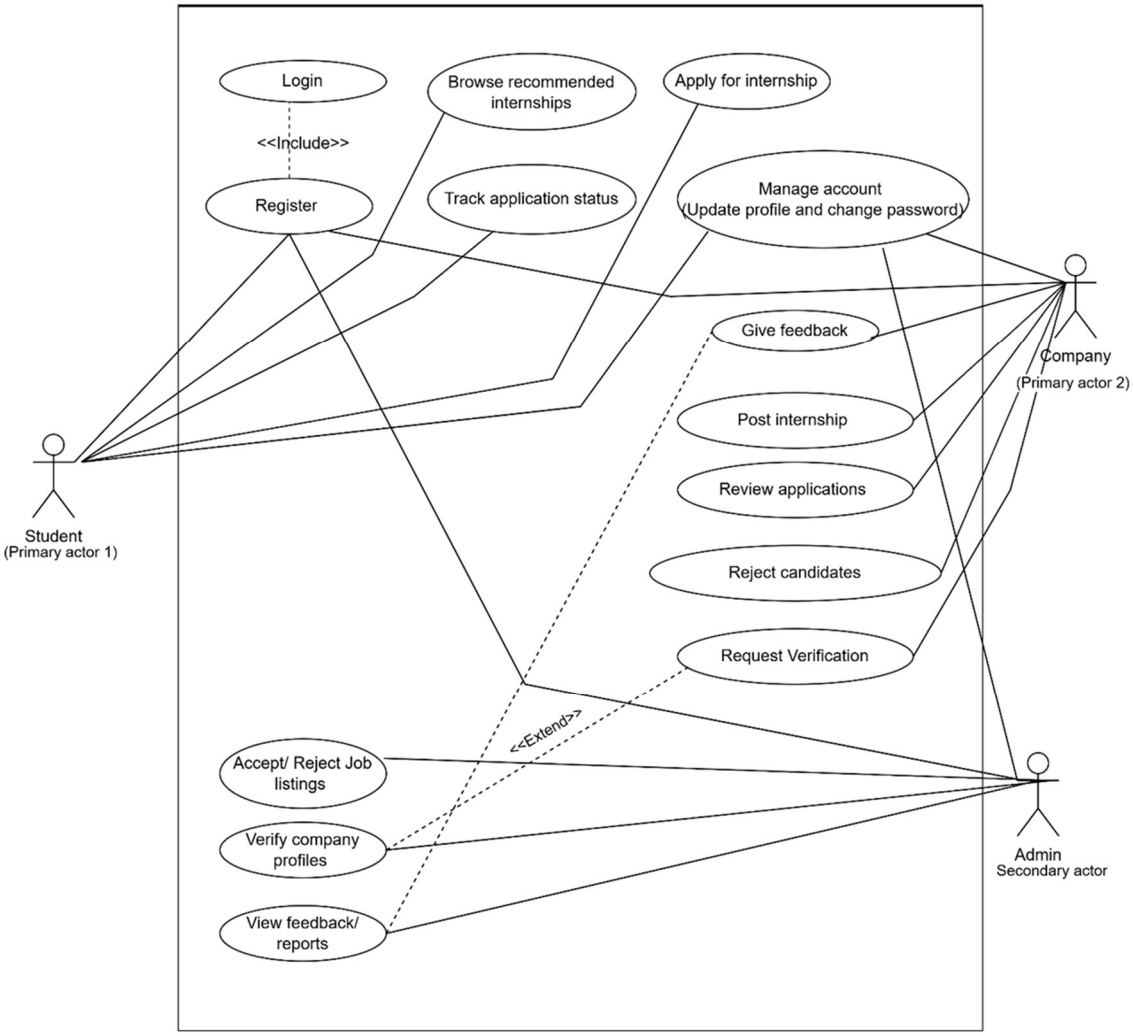


Figure 1: Use-case diagram of INTERNLY app

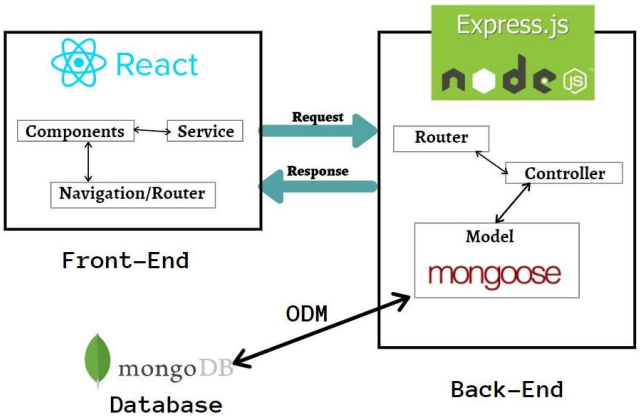


Figure 2: INTERNLY MERN stack architecture diagram

Frontend

The user interface of the system was developed using React Native, a cross-platform framework that enables the use of a shared codebase for both Android and iOS applications. React Native adopts a component-based architecture, whereby elements are built as independent, reusable components (Kadam *et al*., 2023). Because of the component-based design, different parts of the application are easy to maintain and reuse.

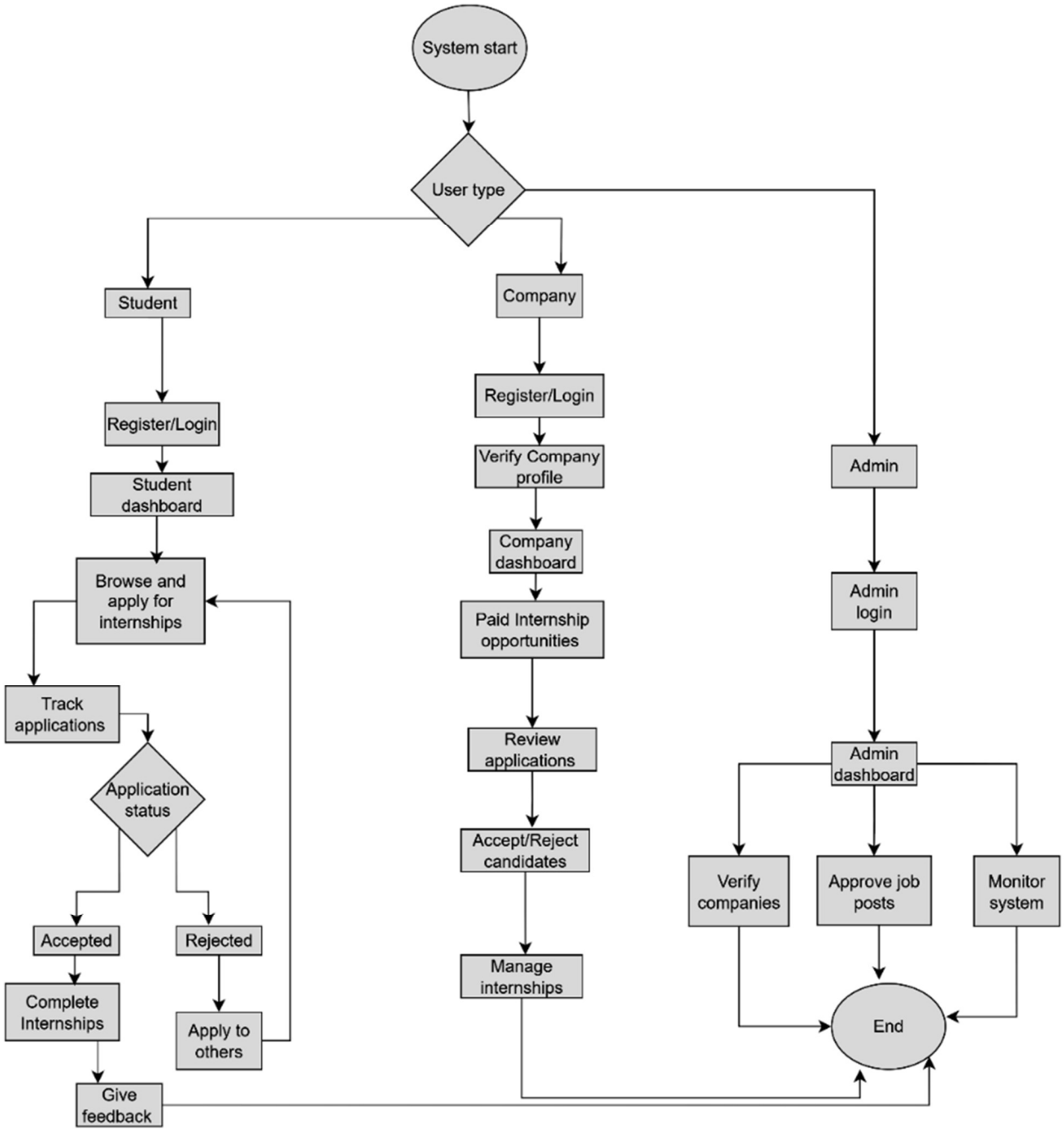


Figure 3: INTERNLY detailed flow chart

Backend

The backend was coded using Node.js together with Express.js as the web framework. The reason for this stack is that it is lightweight and effective at supporting asynchronous event-driven applications. The backend was designed using the Model-View-Controller (MVC) architecture (Necula, 2024),

which guarantees that the code base remains organized and is not too complicated.

Database

The system stores data in MongoDB, a document-oriented NoSQL database. MongoDB was selected mainly because it makes it easy to store all types of data with a flexible schema (Kadam *et al*., 2023). It is important to have this flexibility because users' details, application records, and organizational metadata are dynamic. The schemas were made in a way that it optimizes computing resources, the documents used less space, referred to data sets appropriately, and accurately indexed the important data fields.

Application Workflow

Figure 3 shows the flow chart of INTERNLY application. INTERNLY defines two main user roles: student and organization. Each has distinct capabilities under role-based access control (RBAC) (Singh, *et al*., 2024). For instance, students can browse internship listings and submit applications, but they cannot create postings. Conversely, organization accounts can create and manage internship adverts and review applications, but they cannot apply for internships. This RBAC approach is a common practice in that each user or group is assigned a specific role that determines the permissions granted on API resources. The typical workflow is explained as follows: an organization user creates an internship advert by filling in the title, description, requirements, deadlines, etc. This advert is stored in the database and becomes visible to student users. A student user (once authenticated) can view available internships and apply to ones of interest. To apply, the student uploads required documents, including a Curriculum Vitae (CV) and a cover letter, and optionally adds a comment to be sent alongside the documents to the prospective organization. This creates an application record linking the student and the internship. The organization user can then view all applications for their postings, examine the attached materials, and update each application's status (e.g., accepted, rejected, or pending). Status changes and new application submissions trigger notifications in the form of email or in-app alerts to the relevant users. Throughout, clear APIs enforce that students cannot see other applicants' data, and organizations cannot access data outside their own postings.

Media Handling

User-uploaded media including resumes, cover letters, company logos are handled by Cloudinary (Feinberg, 2020). When a user submits a document or image, the mobile app sends the file to the backend, which in turn uploads it to Cloudinary using Cloudinary's API. Cloudinary stores the file in cloud storage and returns a Uniform Resource Locator (URL). The URL is saved alongside the metadata on MongoDB. Later, when the app needs to display a PDF or image, it loads it from the Cloudinary URL which is served via Cloudinary's CDN. This approach simplifies file management as binary files are not stored on the database or server filesystem. Cloudinary also provides on-the-fly transformations like automatic PDF preview, image resizing which can be leveraged for better performance and user experience.

Deployment

The components of INTERNLY system are deployed in a cloud environment. The Node.js/Express backend is hosted on a cloud server with the URL render.com. The MongoDB database runs on a managed service, MongoDB Atlas. Environment variables store configuration like database URLs and API keys separately from code. The React Native app is built and distributed via APK and Expo-go for rapid testing. Continuous deployment pipelines automate the process by pushing code to the repository, thereby triggering a rebuild and redeployment of the backend and a test release of the mobile app. This ensures the live system is always up-to-date with the latest code and database.

Security and Privacy

Security best practices were followed throughout INTERNLY app development. All client–server communications use Hypertext Transfer Protocol Secure HTTPS (Transport Layer Security-TLS) to encrypt data in transit. For authentication, JSON Web Tokens (JWTs) were used. After a successful login, the server issues a signed JWT to the client, which the client includes in the authorization header of future requests. The server verifies this token on each request. This stateless authentication model avoids storing session data on the server and keeps the communication secure. User passwords are never stored in plaintext. Before saving the passwords, bcrypt to hash passwords are applied. Hashing is a one-way transformation, so stolen hashes cannot be reversed to recover the original password. We also implement role-based access control, where each account is assigned a role – student or organization, and each API endpoint checks this role to allow only permitted actions. Lastly, only the minimum personal data needed is stored, and Cloudinary's secure hosting is relied on for media storage. Together, these measures help ensure that user data remains confidential and secure.

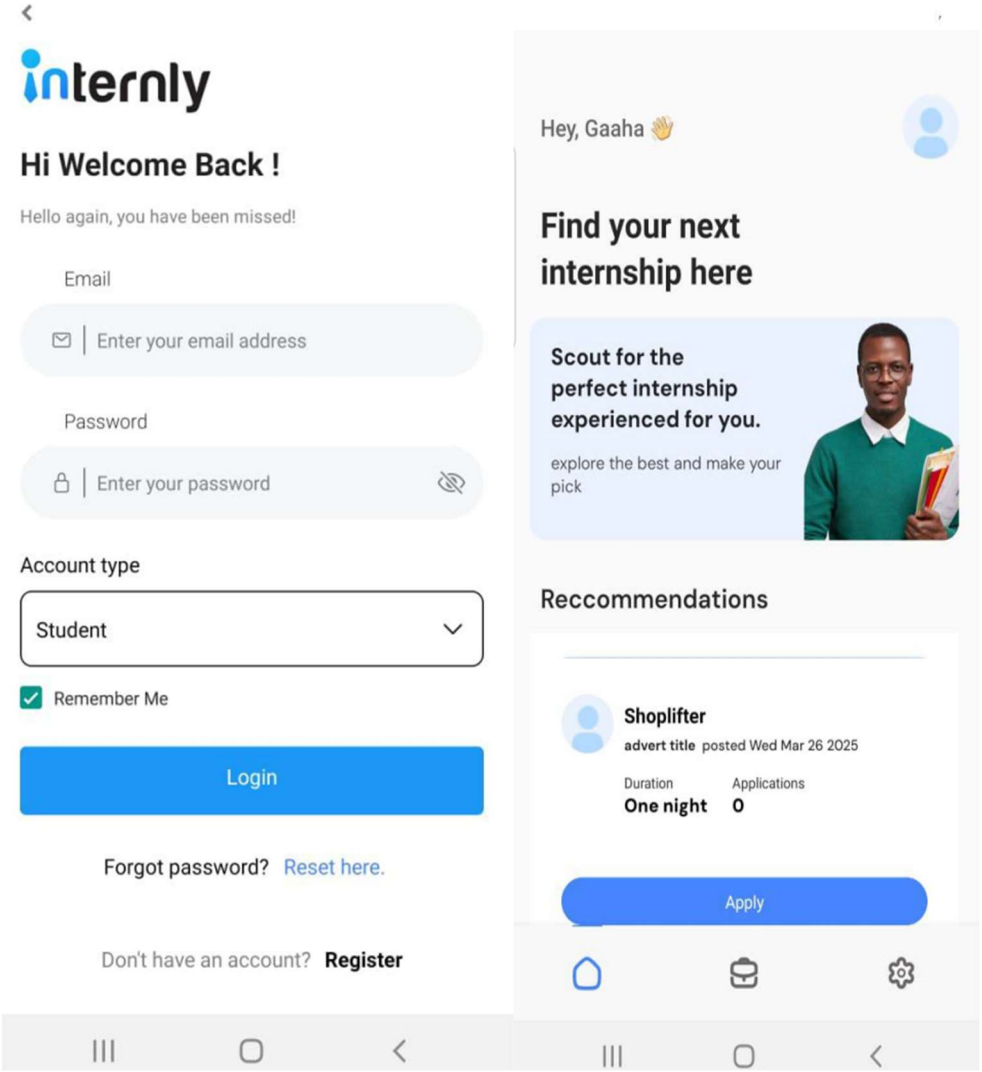


Figure 4: INTERNLY mobile application interface

Results and Discussion

Testing and Validation

A multi-layered testing approach was implemented to evaluate the quality of INTERNLY. Unit testing was used to verify individual components, including API endpoints and utility functions. For example, backend controllers were tested with mock requests to ensure they return the expected data or errors. Integration testing covered end-to-end scenarios across modules by testing the full flow of creating a user, logging in, posting an internship, and applying to it. These tests helped identify issues relating to how modules interact. In addition to automated tests, early user testing with a small group of volunteers (including some who had taken the initial survey was conducted. Feedback from these pilot users was collected through follow-up interviews and bug reports. This user feedback led to refinements: for example, improving the application form User Interface (UI) and clarifying error messages. Throughout development, iterations were carried out on the design, for example, adjusting the recommendation algorithm parameters based on this validation. This combination of automated testing and real-user validation follows agile best practices to ensure reliability and usability.

INTERNLY User Interfaces

In addition to the functional testing explained above, typical user interface flows were analyzed for each user role. Figures 4 through 8 showcase the key screens experienced by company (employer) and student users during routine clicks. These visuals highlight key processes such as internship advert creation by the employer, the employer's advert dashboard, the prospective intern's advert view, and the employer's application review panel. A detailed breakdown of each interface follows in the subsequent discussion.

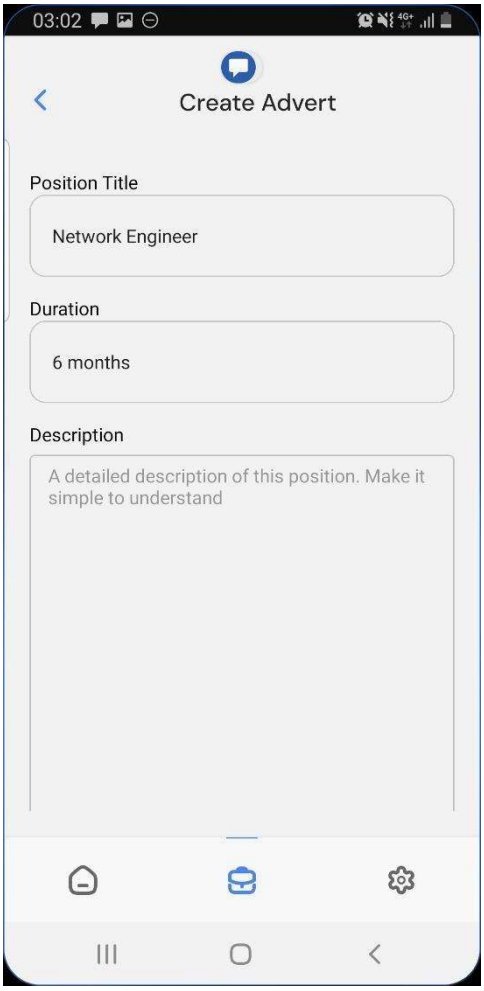


Figure 5: INTERNLY create advert interface

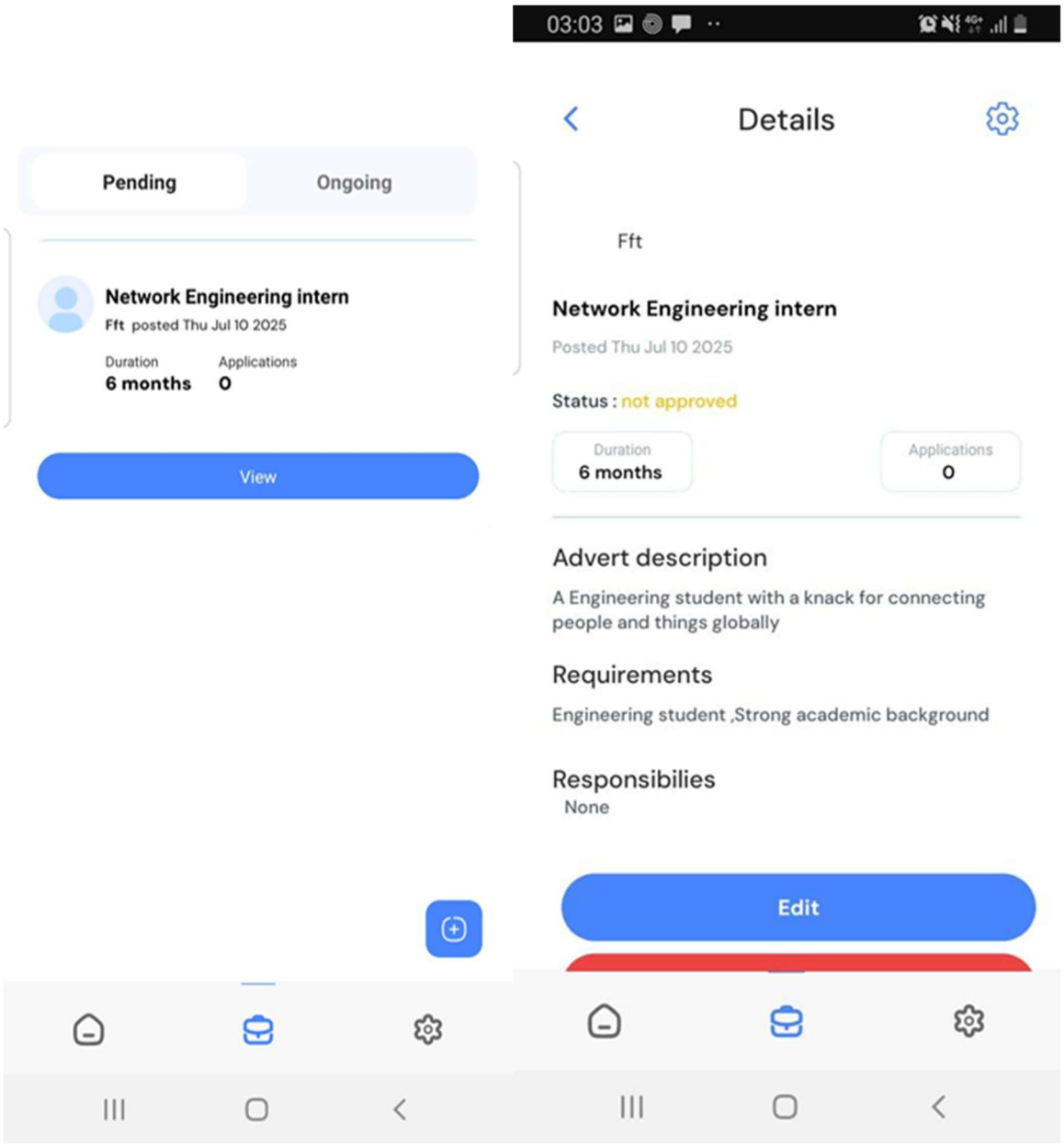
A sample of INTERNLY's mobile application interface is presented in Figure 4 where a user gets to register before being permitted to use the app either as a student or as an employer.

Figure 5 shows the company interface screen for creating a new internship advert, with input fields for position title, duration, and a text area for the advert description. The "Create Advert" form allows an employer to enter the key details of a new internship listing. For instance, as revealed in Figure 5, the company inputs "Network Engineer" as the position title and "6 months" as the duration. Additional fields for description and requirements, enable the employer to specify the internship qualifications and responsibilities. Once all fields are completed, the employer can submit the form (via a "Submit" button) to post the advert on the platform.

The company dashboard (Figure 6a) presents a list of "Pending" internship adverts. Each advert card includes key details such as the internship title ("Network Engineering Intern"), company name ("Fft"), posting date, duration ("6 months"), and the current application count (0). A prominent blue "View" button allows users to access the advert's full details, while a "+" button at the bottom right corner

lets employers initiate the posting of a new advert. Advert specifics are depicted in Figure 6b. Through this dashboard interfaces (Figures 6a and 6b), employers can efficiently oversee and manage all internship postings. Tabs like "Pending" and "Ongoing" categorize adverts by status for streamlined navigation. The advert cards are arranged chronologically and offer essential information at a glance. Employers can tap the "View" button to open or modify an advert with the detailed view shown in Figure 6b. Selecting the "+" icon launches the advert creation form, depicted in Figure 5.

The student interface view of an internship advert presented in Figure 7 features a structured layout displaying the position title ("Network Engineering Intern"), company name ("Brand name"), posting date, and comprehensive advert details, including job description, requirements, and responsibilities.



Figures 6a and 6b: INTERNLY company dashboard

At the bottom, the advert shows the internship duration ("6 months") and the number of received applications. (0), accompanied by a clear and accessible "Send Application" button. From the student's perspective, this interface delivers all essential information in structured format, enabling informed decision-making before applying for a role. The transparent presentation of role expectations promotes clarity, while the consistent layout and prominent call-to-action help streamline the application process. The application review screen for a specific applicant is presented in Figure 8. This screen displays key information such as the application title, submission date, and current status ("submitted"). The applicant's cover letter is shown in full beneath these details, offering the employer a comprehensive view of the candidate's interest and qualifications. At the bottom, two clearly labeled action buttons – "Hire Student" (blue) and "Reject" (red) enable quick decision-making by the employer. Once a student submits an application, the employer gains access to all relevant details for evaluation. The visible cover letter provides insight into the applicant's motivation, while the status label (e.g. "Interested") shows the applicant's expressed interest. Employers can either proceed with hiring by tapping "Hire Student," which triggers a confirmation step, or decline the application using the "Reject" option.

Conclusion

This paper presents an Android application purposely built to help Nigerian students in tertiary institutions in the country to seamlessly access internship opportunities based on their academic background, interests, and employer-defined criteria. INTERNLY is a highly secured system that utilizes rule-based matching algorithms and centralized management tools to streamline the placement process. It also features an intelligent recommender system powered by the Term Frequency-Inverse Document Frequency (TF-IDF) algorithm that analyzes keywords in student profiles and internship descriptions to suggest highly relevant opportunities that may not surface through direct matching alone. More tailor-made features offered by INTERNLY include student and organization profile creation, personalized internship listings, application tracking, real-time email and push notifications of important updates that ensure users never miss deadlines, new opportunities, or messages. Its secure messaging feature allows students and organizations to communicate directly, while the UI is user-friendly. All posted internship opportunities are reviewed and approved by platform administrators before becoming visible to students. This process ensures quality control and protects users from misleading or inappropriate content.

References

Mohd Zamri Abu Bakar, Ariff Azly Muhamed, Mohd Faizul Hassan, Mazuin Mat Halif and Nur Najjah Abd Rahim (2023), Mobile Application Development for Internship: A Case Study of Li2U. *International Journal of Academic Research in Business and Social Sciences*, 13(3), 319 – 324.

Chand, V., and Deshmukh, K. (2019). Addressing the undergraduate internship challenge in developing countries. Education + Training, 61(9), 1064-1077. https://doi.org/10.1108/et-12- 2018-0254

Faridiansyah, M., Irawan D., Purnamassari S. D. and Suyanto (2024). Intern Placement and Monitoring System at SMK YP Gajah Mada Palembang, *Brilliance: Research of Artificial Intelligence*, 4(1), 463 – 469.

Feinberg, M. (2020) 'Uploading, Managing and Delivering PDFs with Cloudinary',
Cloudinary Blog, 21 May. Available at:
https://cloudinary.com/blog/uploading_managing_and_delivering_pdfs (Accessed: 9 June 2025).

Hartley, T., and MacCallum, K. (2018). Managing expectations during internship matching, *NZACE International Conference*, Waiheke Island, Auckland.

Hendrawan, B., Novianto, R. A. and Sahara, S. (2024) 'Development of Website-Based Information System to Facilitate Students in Finding Internship Opportunities', *International Journal of Software Engineering and Computer Science*, 4(2), pp. 595–603. Available at: https://doi.org/10.35870/ijsecs.v4i2.256.

Ibrahim, N., Hanum, H. M., Bakar, Z. A., and Abdullah, N. A. S. (2021) 'Student-Industry Matching for Internship Placement', in *2021 Fifth International Conference on Information Retrieval and Knowledge Management (CAMP)*, Kuala Lumpur, Malaysia: IEEE, pp. 122–126. Available at: https://doi.org/10.1109/CAMP51653.2021.9498088.

Kadam Y., Goplani A., Mattoo S, Gupta S. K., Amrutkar D., Dhanke J. and Kadam Y. V. (2023) 'Introduction to MERN Stack and Comparison with Previous Technologies', *European Chemical Bulletin*, 12 (Special Issue 4), 14382–14386. Available at: https://doi.org/10.48047/ecb/2023.12.si4.1300.

Mahendru, M., Nigam, S., Bajpai, R., and Ahmad, Z. (2024). *Campus ease: Elevating student placements and career success.* 2024 International Conference on Electrical Electronics and Computing Technologies (ICEECT), Sharda University, Greater Noida, [Online], 1–6.
https://doi.org/10.1109/ICEECT61758.2024.10739082

Necula, S. (2024) 'Exploring the Model-View-Controller (MVC) Architecture: A Broad Analysis of Market and Technological Applications'. Available at: https://doi.org/10.20944/preprints202404.1860.v1.

Pressman, R.S. and Maxim, B.R. (2015) *Software Engineering: A Practitioner's Approach*. 8th edn. New York: Mc Graw Hill Education.

Singh, J., Rani, S. and Kumar, V. (2024) 'Role-Based Access Control (RBAC) Enabled Secure and Efficient Data Processing Framework for IoT Networks', *International Journal of Communication Networks and Information Security*, (16), pp. 91–103.

Tyagi D.; Kazim D., S. Bhadra, Gupta A., Kumar P., and Sharma A., Job and Internship Assistance Application (2023),

International Conference on Disruptive Technologies (ICDT), Greater Noida, India, 2023, 571-575, doi: 10.1109/ICDT57929.2023.10150490.