



An-Najah National University
Faculty of Engineering & Information Technology
Computer Engineering Department

InternSphere

Prepared By:
Sohaib Arafat
Osama Dweikat

Supervised By:
Dr. Raed AlQadi

Presented in partial fulfillment of the requirements
for Bachelor degree in Computer Engineering

June 10, 2025

Acknowledgment

First of all, we would like to take this opportunity to thank our supervisor, Dr. Raed AlQadi, for his time, effort, and valuable guidance that he has given to us during the course of this project. We would also like to thank the academic staff of the Computer Engineering Department for their knowledge and support over the years. We appreciate their guidance and encouragement in the completion of this project.

Disclaimer

The following report has been authored by students Sohaib Arafat and Osama Dweikat from the Computer Engineering Department, Faculty of Engineering, An-Najah National University. The report has been through minimal modifications, limited to editorial corrections, and may still contain errors in language and content. It is important to mention that the opinions expressed within the report, including any conclusions and recommendations, solely belong to the students. An-Najah National University bears no responsibility or liability for any consequences arising from the utilization of this report for purposes other than its intended commission

Contents

Acknowledgment	i
Disclaimer	ii
Abstract	viii
1 Introduction	1
1.1 Background and Motivation	1
1.2 Objectives	2
1.3 Significance	3
1.4 Report Organization	3
2 Literature Review	4
2.1 Internship Management Systems	4
2.2 Artificial Intelligence in Educational Platforms	4
2.3 Communication and Collaboration in Educational Technology	4
3 Methodology	5
3.1 Tools, Methods and Programming Languages	5
3.1.1 Client Side	5
3.1.2 Server Side	6
3.1.3 Database	7
3.2 System Architecture	7
3.2.1 Architectural Pattern	7
3.2.2 Database Design	7
3.2.3 Security Framework	8
3.3 System Implementation	9
3.3.1 Student Interface	10
3.3.2 Company Personnel Interface - Training Lead/HR	15
3.3.3 Company Personnel Interface - Trainer/Mentor	19
3.3.4 Practical Training Center Interface	22
3.3.5 Mobile Interfaces	30
3.3.6 System Notifications	46
4 Discussion and Results	55
4.1 Results	55
4.1.1 System Implementation	55
4.1.2 Feature Development	55
4.1.3 AI Integration	55
4.1.4 Technical Achievements	56

4.2	Discussion	56
4.2.1	Development Challenges and Solutions	56
5	Conclusions and Recommendations	58
5.1	Conclusions	58
5.2	Recommendations and Future Work	58
	References	59

List of Figures

3.1	Login Page	9
3.2	Request Access for Students	9
3.3	Student Dashboard	10
3.4	Internship Browser	10
3.5	Internship Application Form	11
3.6	Internship Application Form (Cont'd)	11
3.7	Applications Tracker	11
3.8	Tasks	12
3.9	Task Submission	12
3.10	Task Submission (Cont'd)	13
3.11	Weekly Reports - Main	13
3.12	Weekly Report (Cont'd)	14
3.13	Weekly Reports Form	14
3.14	Profile Page	14
3.15	Resume Submission	14
3.16	Meetings Calender (Training Lead)	15
3.17	Internships Creation Page	15
3.18	Internships & Application Forms Creator	16
3.19	Internships & Application Forms Creator (Cont'd)	16
3.20	Mentors Management	17
3.21	Mentors Addition Form	17
3.22	Student Applications Tracker	18
3.23	Student Applications Forms AI Powered Review	18
3.24	Student Applications Forms Answers	18
3.25	Final Decision & Interview Scheduling	18
3.26	Final Application Decision	18
3.27	Meetings Calender (Mentors)	19
3.28	Tasks Manager	19
3.29	Task Creation Form	20
3.30	Task Submissions	20
3.31	Meetings Scheduler	21
3.32	Meeting Scheduler Form	21
3.33	Admin Dashboard 1-3	22
3.34	Admin Dashboard 2-3	22
3.35	Admin Dashboard 3-3	23
3.36	Meetings Scheduler	23
3.37	Pending Students Accounts	24
3.38	Student Accept/Reject	24
3.39	Pending Internships Application Approvals	25
3.40	Meetings Scheduler	26

3.41	Meetings Scheduler	26
3.42	Weekly Reports and Students Info	27
3.43	Weekly Reports and Students Info - Expanded	27
3.44	Companies Management Portal	28
3.45	Company Addition Form	28
3.46	Training Manager Account Creator	29
3.47	Internships Approval	29
3.48	Login Page - Mobile	30
3.49	Request Access for Students - Mobile	30
3.50	Student Dashboard - Mobile	31
3.51	Internship Browser - Mobile	31
3.52	Internship Application Form - Mobile	32
3.53	Applications Tracker - Mobile	32
3.54	Tasks - Mobile	33
3.55	Task Submission - Mobile	33
3.56	Weekly Reports Main - Mobile	34
3.57	Weekly Reports Form - Mobile	34
3.58	Weekly Report (Cont'd) - Mobile	35
3.59	Profile Page - Mobile	35
3.60	Resume Submission - Mobile	35
3.61	Meetings Calender (Mentors) - Mobile	35
3.62	Tasks Manager - Mobile	36
3.63	Task Creation Form - Mobile	36
3.64	Task Submissions - Mobile	37
3.65	Meetings Scheduler - Mobile	37
3.66	Meeting Scheduler Form - Mobile	38
3.67	Meetings Calender (Training Lead) - Mobile	38
3.68	Internships Creation Page - Mobile	39
3.69	Internships & Application Forms Creator - Mobile	39
3.70	Internships & Application Forms Creator (Cont'd) - Mobile	40
3.71	Student Applications Tracker - Mobile	40
3.72	Mentors Management - Mobile	40
3.73	Mentors Addition Form - Mobile	40
3.74	Pending Students Accounts - Mobile	41
3.75	Pending Internships Application Approvals - Mobile	41
3.76	Meetings Scheduler - Mobile	42
3.77	Meetings Scheduler Form - Mobile	42
3.78	Weekly Reports and Students Info - Expanded - Mobile	43
3.79	Weekly Reports and Students Info - Mobile	43
3.80	Companies Management Portal - Mobile	44
3.81	Company Addition Form - Mobile	44
3.82	Training Manager Account Creator - Mobile	45
3.83	Account Creation Email with Credentials	46
3.84	Meeting Invite	46
3.85	Interview Invite	47
3.86	Meeting Starting Soon	47
3.87	Meeting Starting Now	48
3.88	Task Overdue	48

3.89	Weekly Report Overdue	49
3.90	Application Deadline Approaching	49
3.91	Mentor Internship Assignment	50
3.92	Application Review	50
3.93	Application Interview Stage	51
3.94	Application Requires Admin Approval	51
3.95	Application Initial Acceptance	52
3.96	Application Final Acceptance - Pending Student Approval	52
3.97	Application Rejected	53
3.98	Full Completion of Mandatory Training Hours	53
3.99	Notifications Tray	54

Abstract

It is a big challenge for IT students to find the right internships, apply for them, and get feedback from them. Companies also face challenges in managing application numbers, intern performance, and systematic training. This results in waste and hampers the university's time and effort in overseeing practical training and tracking progress. Therefore, time is lost, and potential is wasted.

Our platform is a one-stop solution for these issues for students, companies, and universities. It assists students in discovering and applying for internships and receiving feedback on their work from our AI code review tool in real-time. In this way, they can enhance their skills and be workplace-ready. Companies can use this platform to post internship opportunities, sort applications, assign tasks, and monitor intern performance. This saves time for companies, and they can focus more on the training process. Universities can also use this to monitor and oversee the activities of the students and track their progress through detailed reports on their activities to check if the training goals are being met.

This solution is unique because the platform integrates AI feedback to offer students continuous and individualized feedback. The platform also has task management tools, deadline notifications, and video call integration to enable students to meet their trainers or supervisors. These features offer a complete internship experience because feedback is provided, and students have proper communication and goal-based systematic training.

Thus, with the help of these tools, we seek to improve the internship experience for students, companies, and universities. The primary goal is to improve the internship process's overall performance and ensure that students have the right tools and support to succeed in their internships and future careers.

1 Introduction

1.1 Background and Motivation

The rapidly changing Information Technology environment requires internships to connect academic knowledge with professional skills for computer science and IT students who want to enter demanding tech industries. Traditional internship management systems depend on manual fragmented processes which do not fulfill the advanced requirements of technology-focused educational programs. IT departments and tech companies and computer science students face problems with inefficient communication systems and paper-based tracking methods and inconsistent evaluation standards which do not match the technical standards needed for programming and development positions. The COVID-19 pandemic exposed the immediate necessity

need to accept and dismiss for complete digital solutions which must support remote collaboration and code review processes and real-time progress tracking and data-driven decision making for IT internship requirements.

The development of InternSphere emerged from recognizing the significant difference between current internship management systems and the complex needs of Information Technology educational programs. The current systems generate administrative workloads that block effective communication and prevent students from obtaining essential technical learning experiences needed for IT career advancement. Technical supervisors encounter difficulties in monitoring coding progress and project deliverables and programming language and framework skill development because computer science students face complex and unclear processes. The reporting systems of tech companies remain inconsistent while their development workflows and project management systems lack proper integration. The educational institutions of IT need complete analytical capabilities to assess program success and monitor technical skill development and maintain standards that match industry requirements. InternSphere uses modern web technologies and artificial intelligence along with user-centered design principles adapted for technical environments to create an integrated platform which converts IT internship management from administrative work into a strategic educational instrument for developing future technology professionals.

1.2 Objectives

The main purpose of this project involves creating InternSphere as a complete internship management system to transform Information Technology programs' student internship management methods. The following objectives describe the specific goals of this project:

1. **Application Processing Efficiency:** The platform reduces application processing time through automated workflows which eliminate manual paperwork and enable real-time status tracking for students and companies and academic supervisors.
2. **Automated Code Review System:** The system will offer automated code review features together with syntax highlighting across various programming languages and AI-based feedback to enhance student coding abilities and project excellence during their internship period.
3. **Real-Time Progress Monitoring:** The system provides real-time student progress monitoring through automated weekly reports and task completion tracking and performance analytics which enables prompt support during necessary times.
4. **Integrated Communication Platform:** The platform integrates video conferencing and notification systems which cut down communication delays while keeping all stakeholders informed throughout the internship period.
5. **Performance and Security Standards:** The platform will operate at high performance levels while maintaining security standards and educational data protection compliance while supporting multiple universities and hundreds of concurrent users.
6. **Educational Technology Innovation:** The project will present an example of modern educational technology innovation through its combination of contemporary web development frameworks with artificial intelligence and user-centered design principles to build a future-oriented educational platform for IT internship management.

1.3 Significance

InternSphere delivers substantial value to educational technology and IT program management through multiple dimensions.

1. **Educational Institution Impact:** The IT department operations undergo transformation through automated workflows and AI-powered analytics which enables data-driven program improvement and simplified accreditation compliance with quantifiable metrics.
2. **Industry Partnership Enhancement:** The system unifies company-university partnerships through its code review systems and real-time progress tracking which enhances talent acquisition processes and decreases administrative costs for technology companies.
3. **Student Career Development:** The system improves learning results through its intelligent internship matching system and complete skill tracking features and structured feedback mechanisms that enhance student readiness for professional technology careers.
4. **Technical Innovation Contribution:** The educational technology field advances through the successful implementation of artificial intelligence and modern web frameworks and user-centered design principles in multi-stakeholder educational platforms.
5. **Scalable Solution Framework:** The platform architecture provides a reusable framework which enables other institutions to adapt it for collaborative improvement and broad implementation of best practices in IT internship management.

1.4 Report Organization

The report follows a structured format to show the complete development process of InternSphere starting from conceptual design until implementation and evaluation. Chapter 2 conducts an extensive literature review to analyze existing internship management systems and artificial intelligence applications in educational platforms and modern web development frameworks that match the project requirements. The third chapter explains the methodology and system features by describing the technical approach and programming languages and tools selection and complete implementation of core functionalities including authentication and user management and task systems and AI integration. The fourth chapter explains system architecture and design by showing the database schema and API architecture and security frameworks and technology stack decisions which enable platform scalability and reliability. The fifth chapter explains the implementation details through a description of Node.js and Express.js backend development and React.js frontend implementation and Python-based AI microservice integration. The sixth chapter shows results and discussion by presenting system testing outcomes and performance evaluations and user feedback analysis and AI system effectiveness assessment. The final chapter of the report presents project outcomes together with recommendations for future improvements and research directions for IT-focused internship management systems.

2 Literature Review

2.1 Internship Management Systems

Educational institutions currently use traditional internship management systems which depend on inefficient manual procedures that require extensive administrative work. (Johnson et al., 2021) found that manual processes absorb about 40% of administrative staff time within computer science departments thus demonstrating major inefficiencies. Commercial platforms Handshake and Symplicity provide fundamental internship posting features yet they do not include specialized tools needed for technical programs including automated code review and AI-powered student-internship matching and real-time progress tracking. The current limitations prevent these platforms from effectively supporting Information Technology internship programs because they require technical skill assessment and project-based learning.

2.2 Artificial Intelligence in Educational Platforms

Artificial intelligence integration in educational technology demonstrates substantial potential to enhance both assessment precision and individualized learning approaches. Machine learning algorithms achieve assessment accuracy rates of 85% when used for automated code evaluation compared to human evaluators thus making AI integration most beneficial for technical education programs (Zhang et al., 2021). The development of large language models through Google's Gemini AI and OpenAI's GPT systems has created new opportunities for intelligent educational assistants and automated content generation and sophisticated recommendation systems. The implementation of AI in educational production environments faces ongoing challenges because of complex integration requirements and bias reduction needs and maintaining educational standards through AI-based student assessment and support systems.

2.3 Communication and Collaboration in Educational Technology

The success of internship programs depends heavily on effective communication systems because research shows that poor supervisor-student communication leads to 45% of unsuccessful internship experiences (Kumar et al., 2023). Modern educational platforms need multiple notification channels together with video conferencing functionality and messaging systems that adapt to user contexts to meet the needs of students and faculty supervisors and industry mentors and administrative personnel. The COVID-19 pandemic has demonstrated the necessity of strong digital communication tools which enable remote collaboration and real-time progress sharing and seamless coordination between academic institutions and industry partners. The communication requirements serve as fundamental elements for complete internship management platforms which function in modern educational settings.

3 Methodology

The chapter explains the complete methodology behind Intern-Sphere development through its technical approach and system architecture design and implementation strategies. The methodology includes technology selection and programming languages and development frameworks and architectural patterns which fulfill the platform's multi-stakeholder requirements. The chapter explains the structured process for implementing features which includes authentication systems and user management and AI integration and communication frameworks.

3.1 Tools, Methods and Programming Languages

3.1.1 Client Side

The client-side implementation focuses on creating a responsive and cross-platform user interface:

- **React.js Framework:** The React.js Framework serves as the main frontend framework which allows developers to build user interfaces through components while providing efficient state management and virtual DOM rendering.
- **Modern JavaScript Features:** The implementation uses ES6+ features including arrow functions, destructuring, and async/await patterns for seamless API integration
- **Capacitor.js Cross-Platform Runtime:** The React web application functions as a native mobile application on both iOS and Android platforms through this technology which provides native device features to a single codebase.
- **Android APK Generation:** Through Capacitor.js we generated complete Android projects which produced APK files for direct Android device installation thus converting the web application into a native mobile application.
- **Responsive Design:** The frontend architecture uses CSS Grid and Flexbox layouts to implement responsive design principles which deliver optimal user experiences on desktop and tablet and mobile devices.
- **State Management:** The application uses React Hooks together with Context API to manage data flow and component communication efficiently.

3.1.2 Server Side

The server-side architecture employs a microservices approach with multiple specialized services:

- **Gateway API Service:** The Gateway API Service functions as a central API gateway which directs requests between all microservices through a unified client application entry point.
- **User Management and Authentication Service:** The service controls user sign-up processes and authentication functions and implements role-based access control and profile administration through JWT authentication with bcrypt password encryption.
- **Internship Management Service:** The Internship Management Service operates as a system to handle internship postings and applications and task assignments and progress tracking and performance evaluation workflows.
- **AI Exposure Service:** AI Exposure Service operates as a Python-based Flask microservice which provides artificial intelligence operations through specific endpoints for application rating and code evaluation and internship recommendation systems.
- **Deployment Infrastructure:** The deployment system uses Google Cloud Run with Docker containerization for serverless container management and automatic scaling capabilities.
- **Backend Framework:** The Node.js + Express.js Foundation serves as the main backend framework which enables RESTful API development through its robust validation and error handling and security features.
- **AI Integration:** The AI service connects Google Gemini 2.5 Flash with special system instructions through Google Generative AI API for natural language processing and intelligent assessment capabilities (Google DeepMind, 2023).
- **File Storage System:** Google Cloud Storage operates as a file storage system which provides secure access controls and scalable storage capacity for document uploads and resume files and multimedia content.
- **Web Hosting Platform:** The Google Cloud Static Web Hosting platform serves as a website hosting solution which deploys frontend applications through Google Cloud Storage static web hosting for worldwide content delivery.
- **Security Middleware:** The security middleware includes CORS configuration alongside rate limiting and input sanitization and logging features for security and performance monitoring.

3.1.3 Database

The database implementation utilizes MongoDB for flexible document storage:

- **Database Architecture:** The database architecture uses MongoDB with Mongoose ODM to support complex educational data structures.
- **Security Framework:** Security framework established with JWT authentication, role-based access control, and data encryption
- **Cloud Storage Integration:** The system uses Google Cloud Storage for file management and static web hosting through its cloud storage integration.
- **API Architecture:** API architecture developed with comprehensive RESTful endpoints supporting all system functionalities

3.2 System Architecture

3.2.1 Architectural Pattern

InternSphere implements microservices architecture to divide its core application platform from its specialized artificial intelligence processing components. The system uses Model-View-Controller design patterns to establish clear organizational boundaries between data storage mechanisms, business logic operations, and user interface presentations. The architectural methodology enables system modularity which simplifies ongoing maintenance procedures and provides scalable infrastructure for expanding user populations and flexible foundations for future technological capabilities integration. The platform uses RESTful API standards with JSON data formatting for its communication protocols to ensure cross-platform compatibility and external institutional system connectivity. The security architecture relies on JSON Web Token authentication protocols to manage stateless sessions in distributed system environments. The system implements access control through role-based permissions which divide users into five categories: students can submit applications and track their progress and professors oversee academic supervision through supervisory tools and company personnel access mentoring features and internship management capabilities and training administrators handle cross-departmental program management and super administrators control complete system governance. The educational management system uses MongoDB's document-oriented storage architecture to handle complex hierarchical structures while implementing strategic database indexing and aggregation frameworks to optimize query performance for analytical reporting and dashboard functionality. (Auth0, 2024)

3.2.2 Database Design

The main database solution uses MongoDB which stores data through documents to handle the intricate nested data structures found in educational and internship management systems. The database design implements Mongoose Object Document Mapper (ODM) to validate schemas and manage relationships and optimize queries. The design includes discriminator patterns for user type management and embedded documents for performance optimization and indexed fields for efficient querying and data retrieval.

3.2.3 Security Framework

The security implementation consists of multiple layers which include input validation, authentication, authorization and data protection measures. Role-based access control (RBAC) ensures appropriate permission management across five distinct user types: students, professors, company personnel, training administrators, and super administrators. Password security utilizes bcrypt hashing with salt rounds, while sensitive data transmission employs HTTPS encryption and secure cookie handling for session management.

3.3 System Implementation

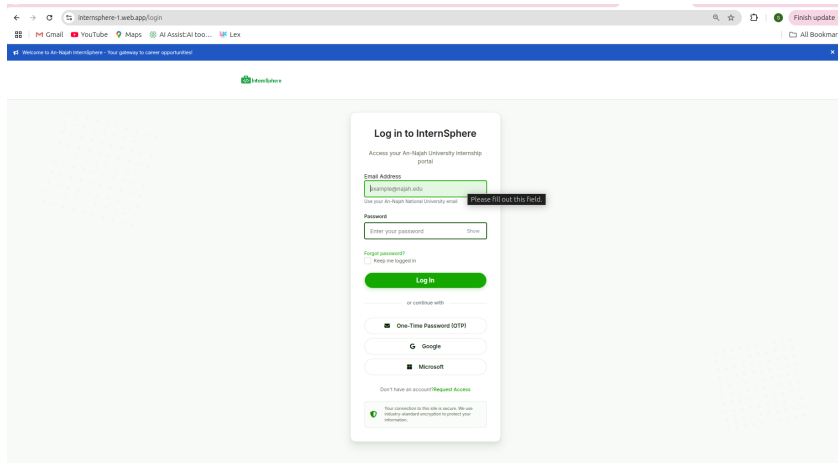


Figure 3.1: Login Page

The login page is unified for all users, providing a single point of access for students, company representatives, training center staff, and university administrators. This streamlined approach simplifies user authentication and ensures a consistent user experience across different roles within the system.

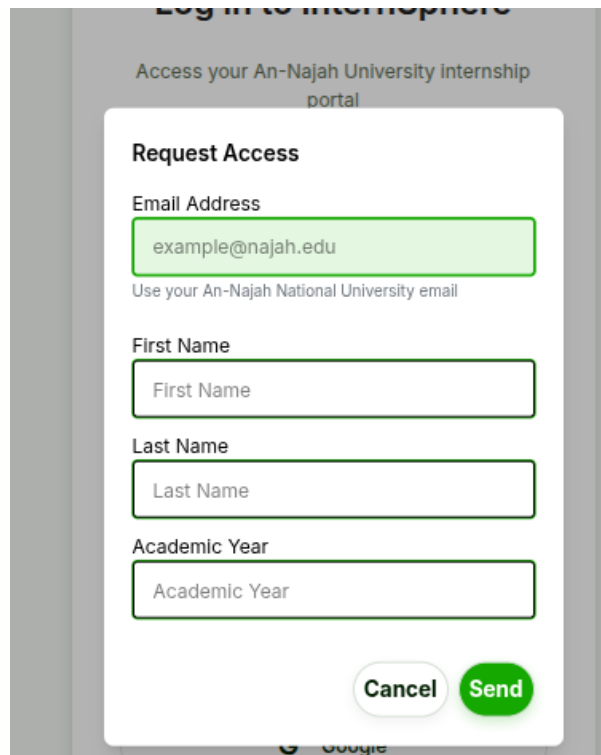


Figure 3.2: Request Access for Students

The request access dialog is designed for students to submit a request for their accounts to be activated. This feature ensures that only verified and approved users can access the system, enhancing both security and administrative control.

3.3.1 Student Interface

The student interface enables students to manage all aspects related to their internships, including task submission, internship application, report submission, and more.

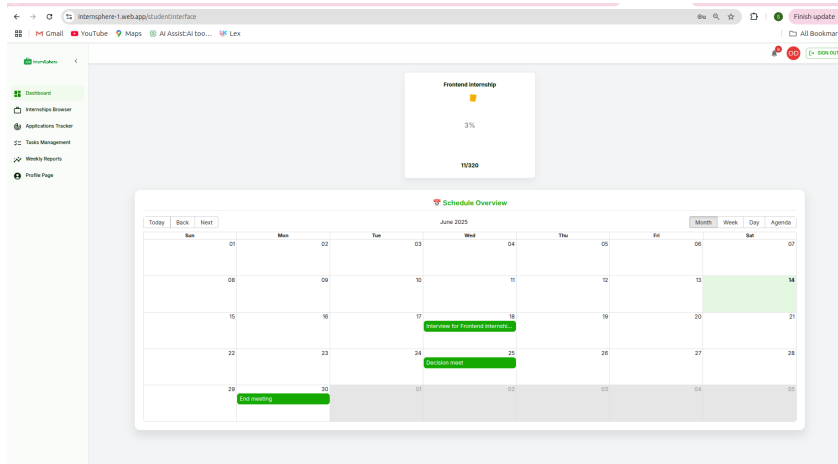


Figure 3.3: Student Dashboard

The main dashboard allows students to view their progress and access their meetings through the calendar.

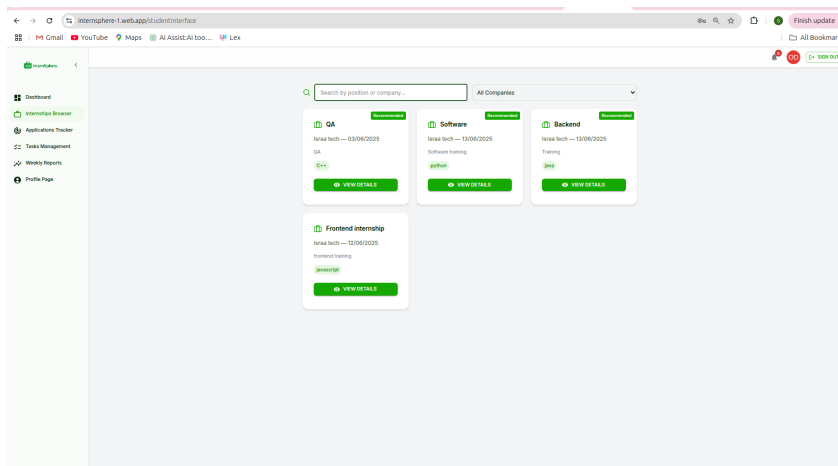


Figure 3.4: Internship Browser

The internship browser interface provides students with an intuitive search and filtering system to discover relevant opportunities.

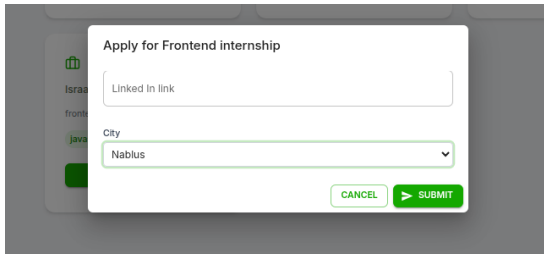


Figure 3.5: Internship Application Form

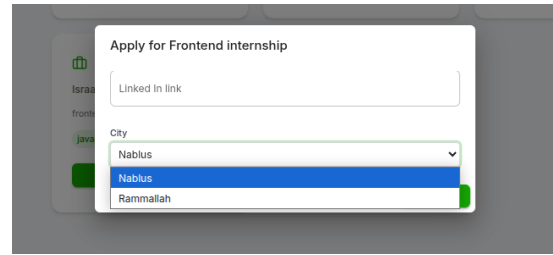


Figure 3.6: Internship Application Form (Cont'd)

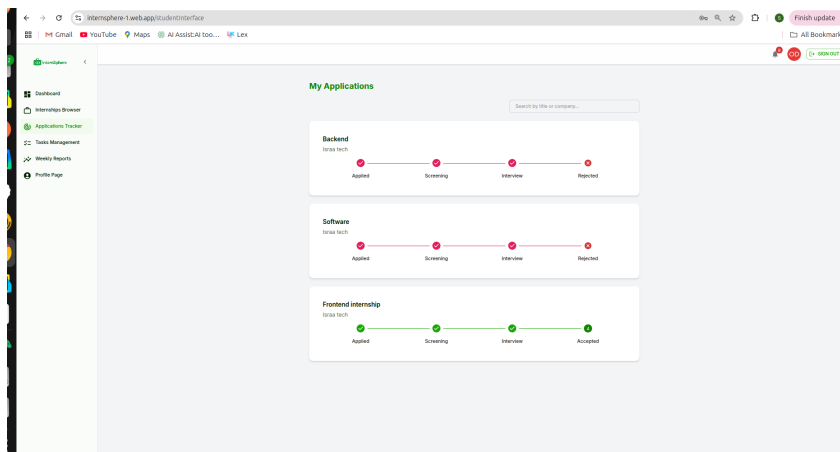


Figure 3.7: Applications Tracker

The application tracker provides students with real-time visibility into their submission status and progress through the three-stage acceptance process.

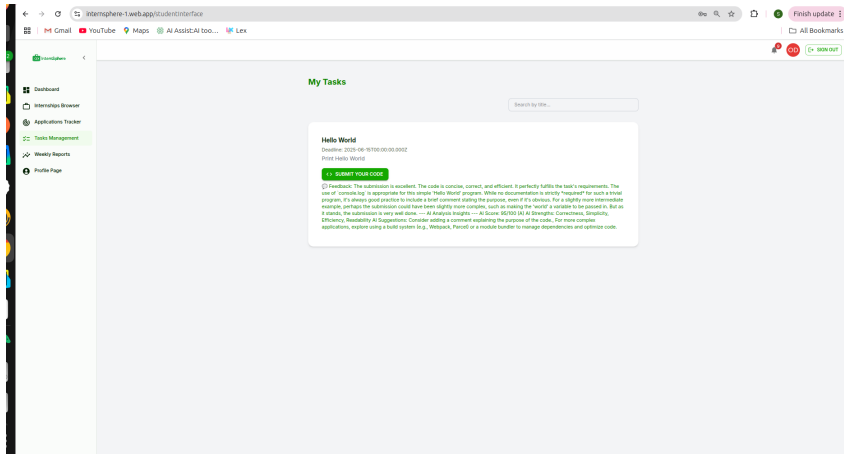


Figure 3.8: Tasks

The task management and submission system enables students to receive assignments from mentors and submit code for comprehensive AI-powered review and feedback.

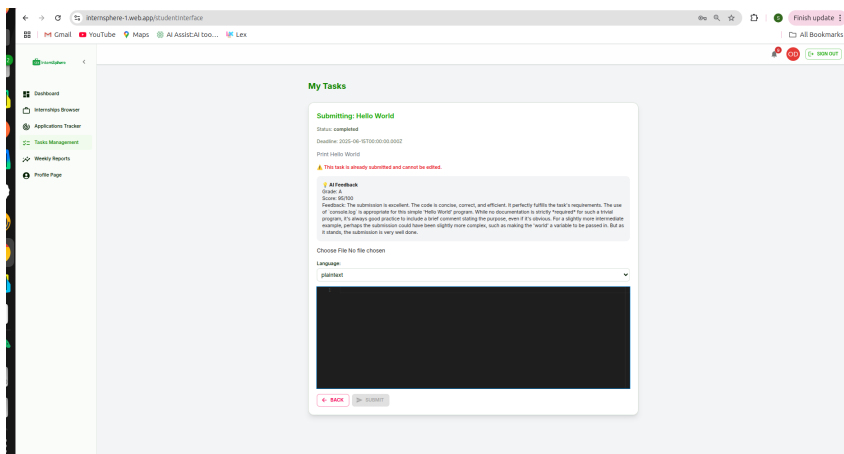


Figure 3.9: Task Submission

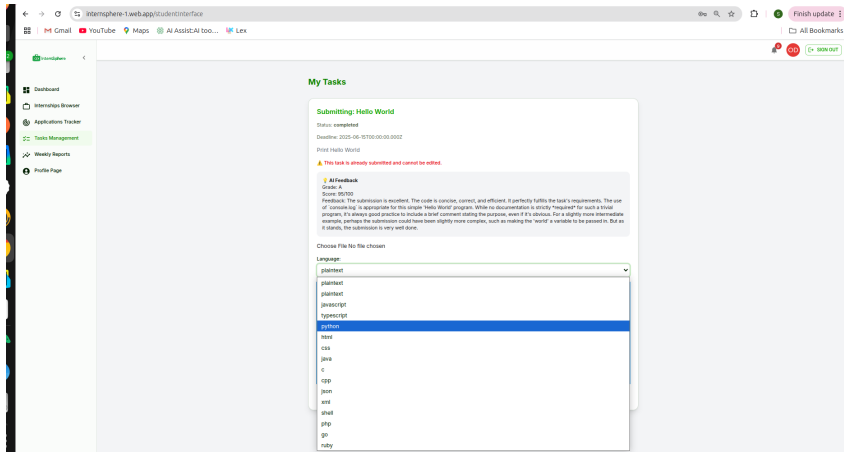


Figure 3.10: Task Submission (Cont'd)

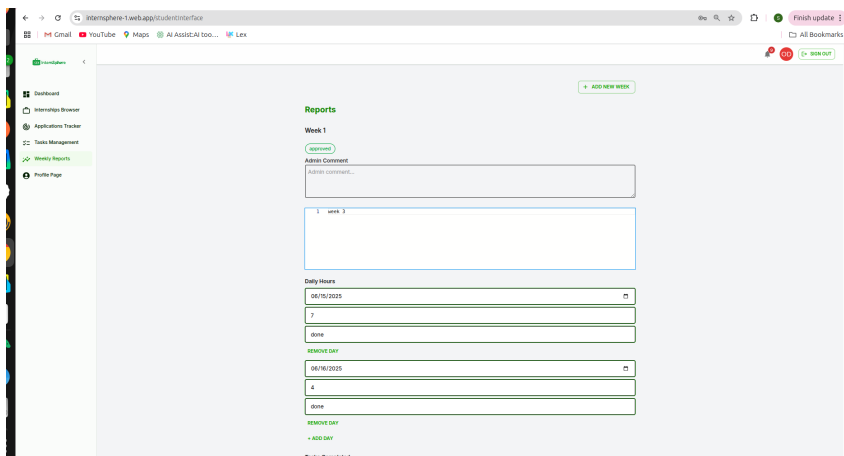


Figure 3.11: Weekly Reports - Main

The weekly reports system allows students to document their daily activities, hours worked, and learning outcomes while providing the training center with structured progress tracking.

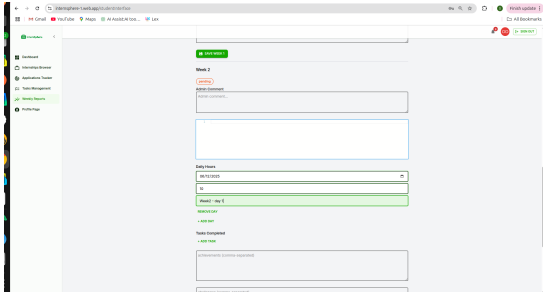


Figure 3.12: Weekly Report (Cont'd)

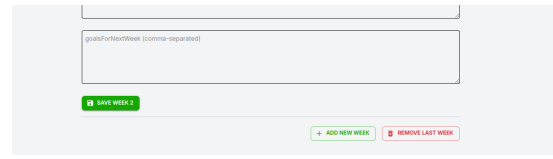


Figure 3.13: Weekly Reports Form

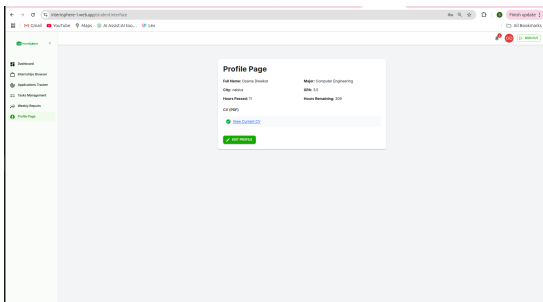


Figure 3.14: Profile Page

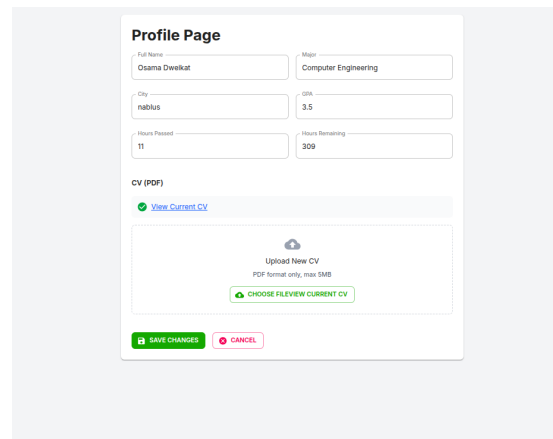


Figure 3.15: Resume Submission

The profile and resume submission feature enables students to upload resumes for companies and maintain student information.

3.3.2 Company Personnel Interface - Training Lead/HR

The training lead interface enables companies to manage internship application forms and allocate staff resources for incoming interns.

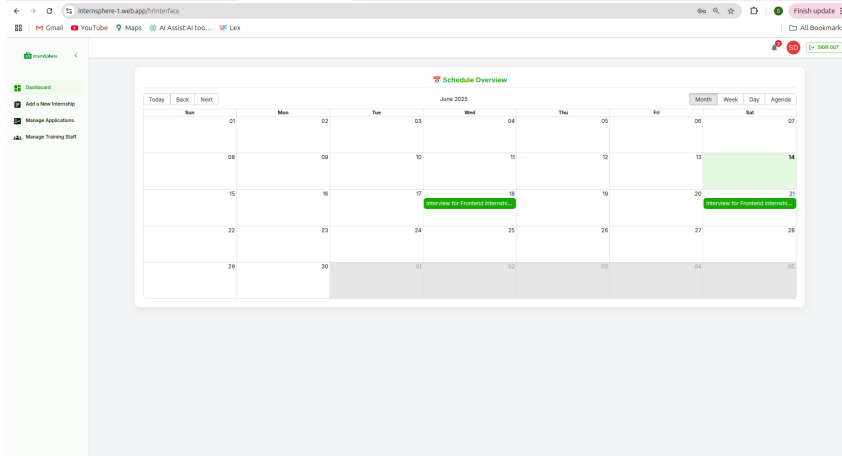


Figure 3.16: Meetings Calendar (Training Lead)

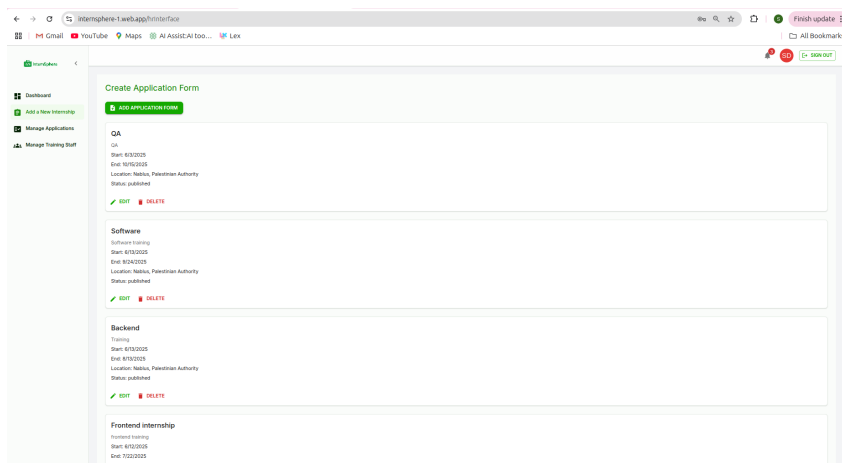


Figure 3.17: Internships Creation Page

This page allows the training lead to create internships and corresponding application forms for them. RetryClaude can make mistakes. Please double-check responses.

Add Application

Title

Description

Requirements

Start Date
mm/dd/yyyy

End Date
mm/dd/yyyy

Application Deadline
mm/dd/yyyy

Work Mode
On-site

Location

Address

City

Country

Is Remote

Required Skills
+ ADD SKILL

Application Questions
+ ADD QUESTION

CANCEL SAVE

Required Skills

Skill Name

Level
Beginner

REMOVE SKILL

+ ADD SKILL

Application Questions

Question 1 Description

Question Type

Text

Radio Box

Select Box

Combo Box

Checkbox

SAVE

Figure 3.19: Internships & Application Forms Creator (Cont'd)

Figure 3.18: Internships & Application Forms Creator

Training lead can create new internships and application forms with different types of questions.

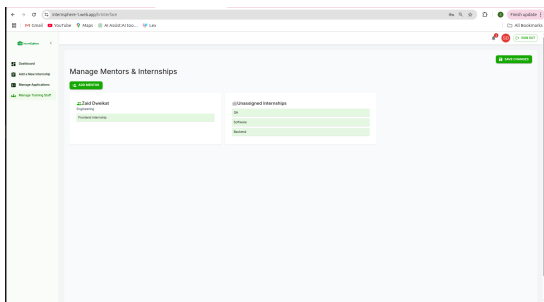


Figure 3.20: Mentors Management

A modal form titled "Add Mentor" is displayed over a dark background. The form contains four input fields: "First Name", "Last Name", "Email", and "Phone Number". At the bottom right of the form, there are two buttons: a green "CANCEL" button and a green "ADD" button.

Figure 3.21: Mentors Addition Form

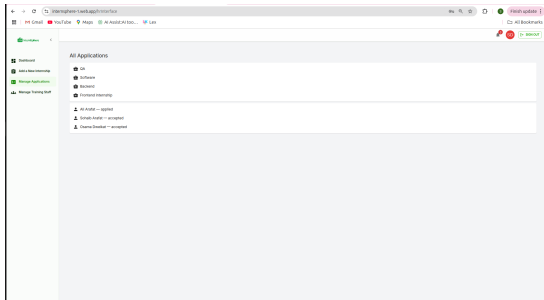


Figure 3.22: Student Applications Tracker

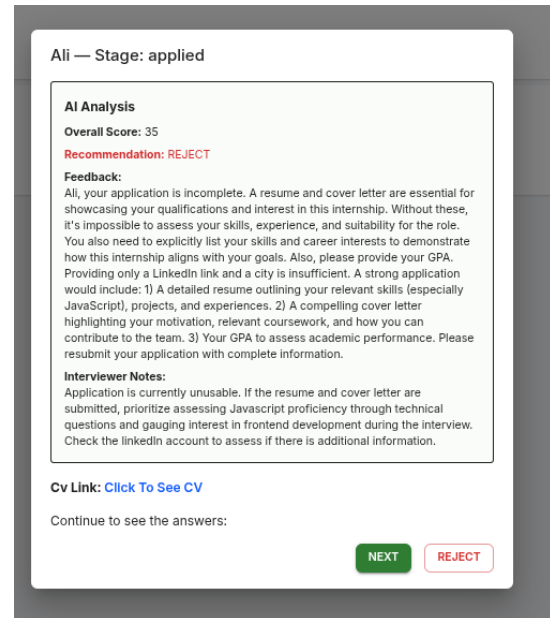


Figure 3.23: Student Applications Forms AI Powered Review

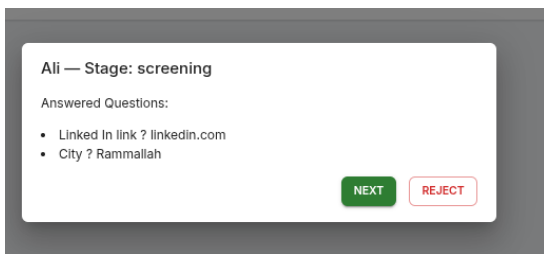


Figure 3.24: Student Applications Forms Answers

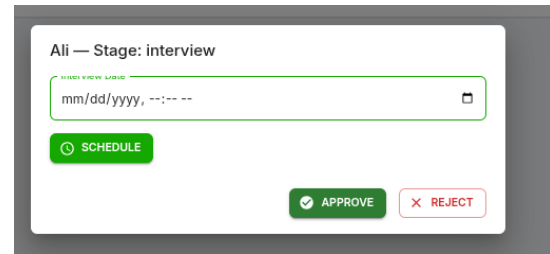


Figure 3.25: Final Decision & Interview Scheduling

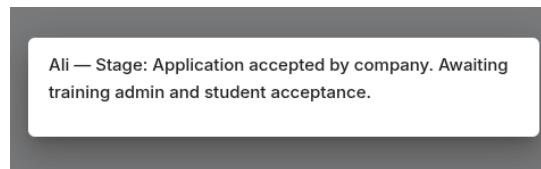


Figure 3.26: Final Application Decision

3.3.3 Company Personnel Interface - Trainer/Mentor

This page is to help training staff/mentors track create new submissions track student tasks and schedule meetings and calls with them.

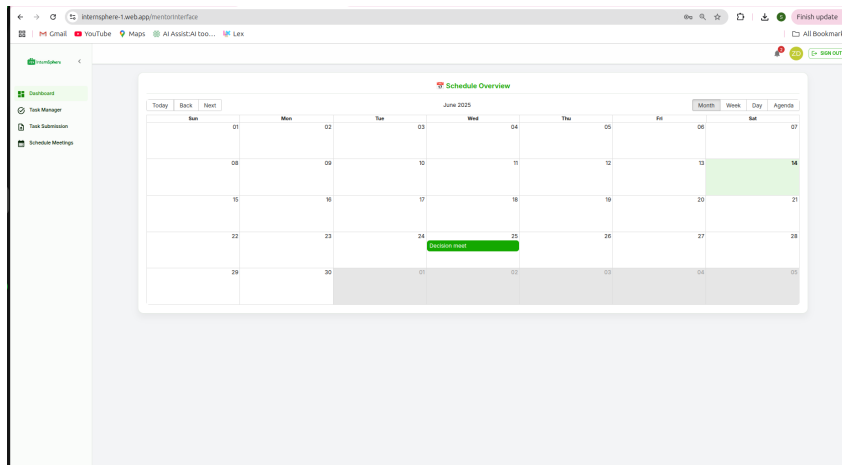


Figure 3.27: Meetings Calendar (Mentors)

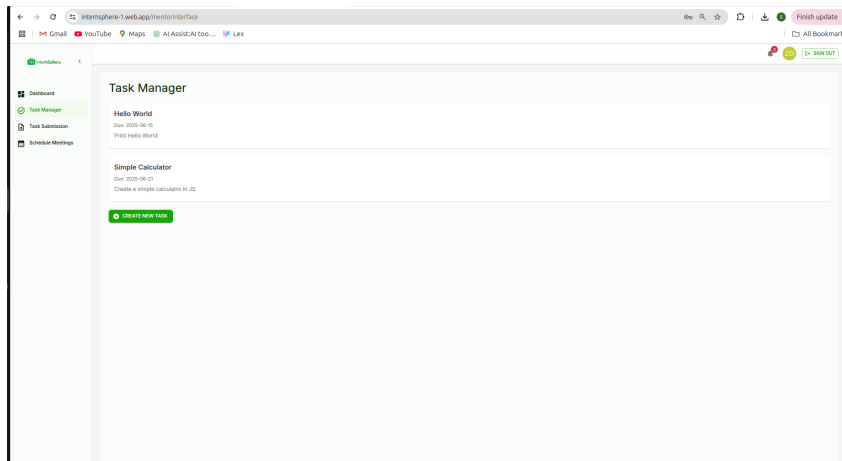


Figure 3.28: Tasks Manager

The task management page enables the mentor to create tasks and assign it to students.

Create New Task

Title

Due Date
mm/dd/yyyy

Description

Assign to Students

Ali Arafat Sohalb Arafat

Osama Dweikat

✕ CANCEL ✓ CREATE

Figure 3.29: Task Creation Form

Task Submissions

Osama Dweikat - main.txt

```
1 - simple tag: <h1> world </h1>
```

The submission is excellent. The code is concise, correct, and efficient. It perfectly fulfills the task's requirements. The use of "simple tag" is appropriate for this simple "hello world" program. While no documentation is strictly "required" for such a trivial program, it's always good practice to include a brief comment within the program, even if it's obvious. For a slightly more intermediate exercise, perhaps the submission could have been slightly more complex, such as making the "world" a variable to be passed in. But as it stands, the submission is very well done.

functionality: 100	codeQuality: 100
performance: 100	documentation: 80
bestPractices: 100	creativity: 90

Global Rating: 👍 👎

📄 Attachments 🗑️ Select

Task ID: 12345

Figure 3.30: Task Submissions

The submission page is to view students submission for feedback and with AI scores and recommendation to help the mentor.

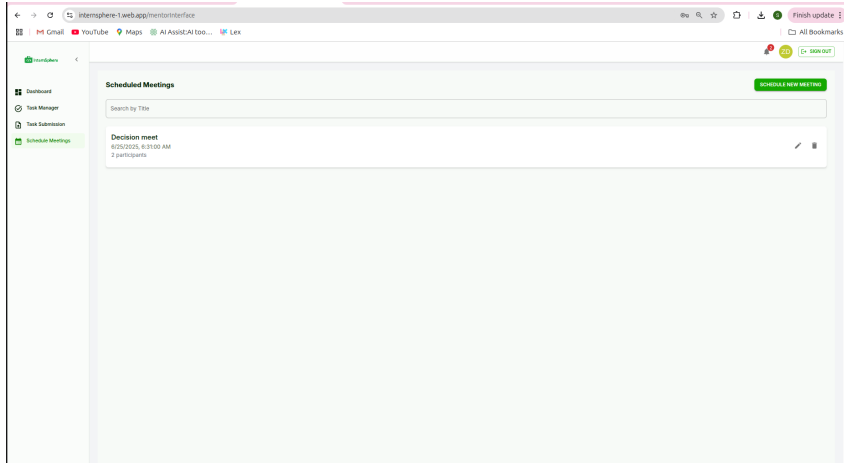


Figure 3.31: Meetings Scheduler

The scheduling page is to help the mentor to schedule meetings with students.

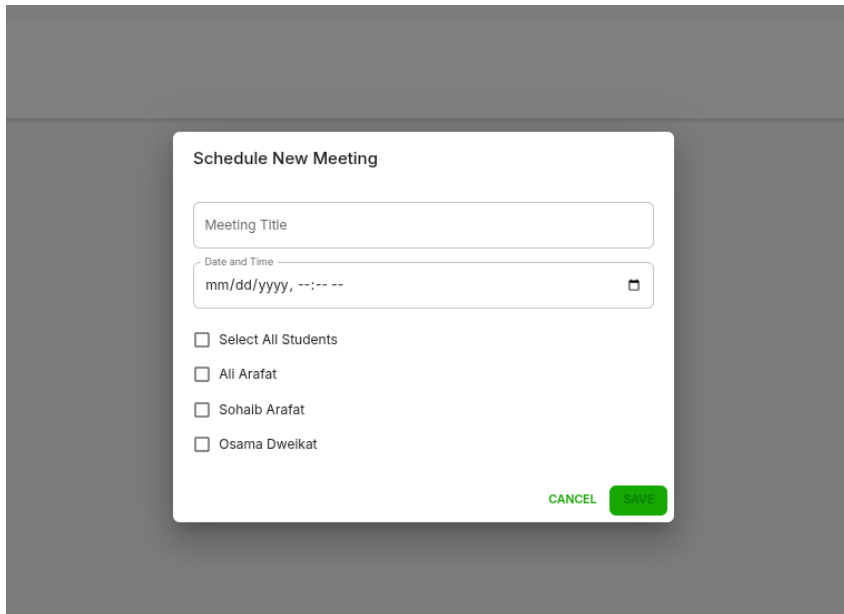


Figure 3.32: Meeting Scheduler Form

3.3.4 Practical Training Center Interface

The practical training center interface is to help the training center track all aspects of the training programs from giving students access to the platform to approving internships and training programs and weekly reports, monitoring student progress, coordinating with companies and universities, managing documentation and compliance requirements, generating institutional reports, overseeing mentor assignments, and facilitating communication between all stakeholders throughout the entire internship lifecycle.

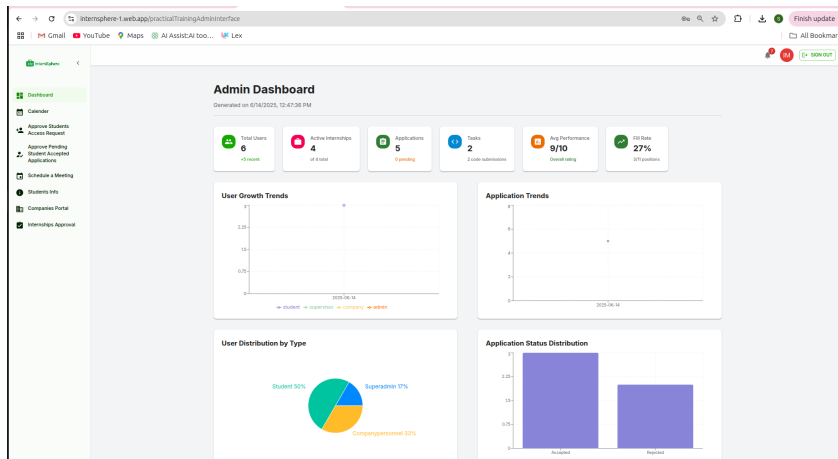


Figure 3.33: Admin Dashboard 1-3

The admin dashboard displays detailed stats and info about the overall system information including user registration metrics, internship application trends, completion rates, performance analytics, system usage patterns, user engagement data, company participation statistics, university partnership status, task completion analytics, meeting scheduling frequency, report submission compliance.

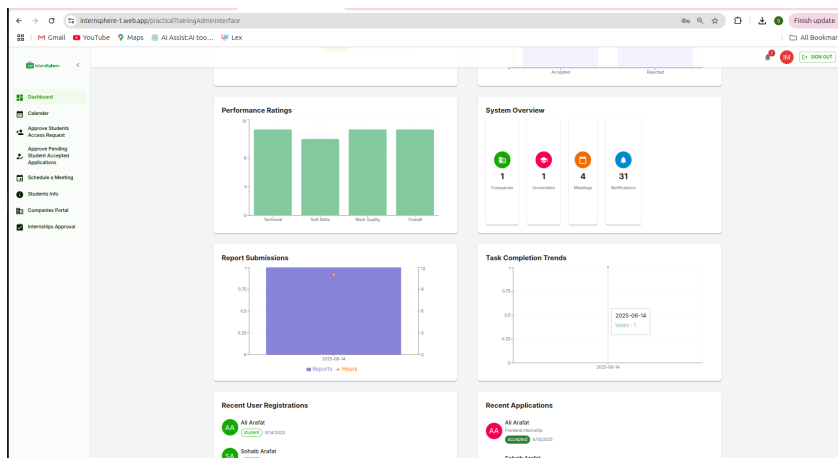


Figure 3.34: Admin Dashboard 2-3

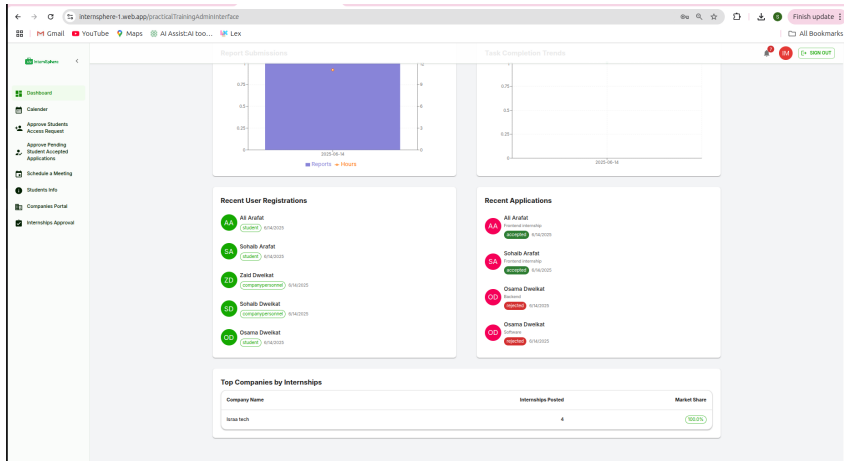


Figure 3.35: Admin Dashboard 3-3

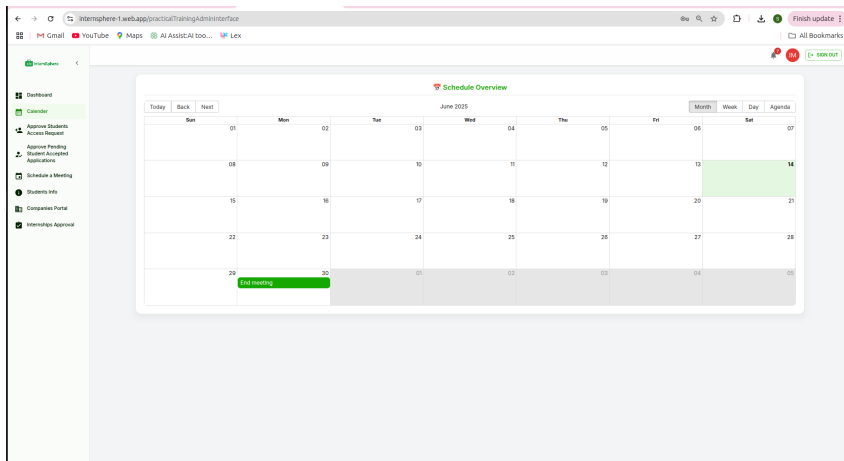


Figure 3.36: Meetings Scheduler

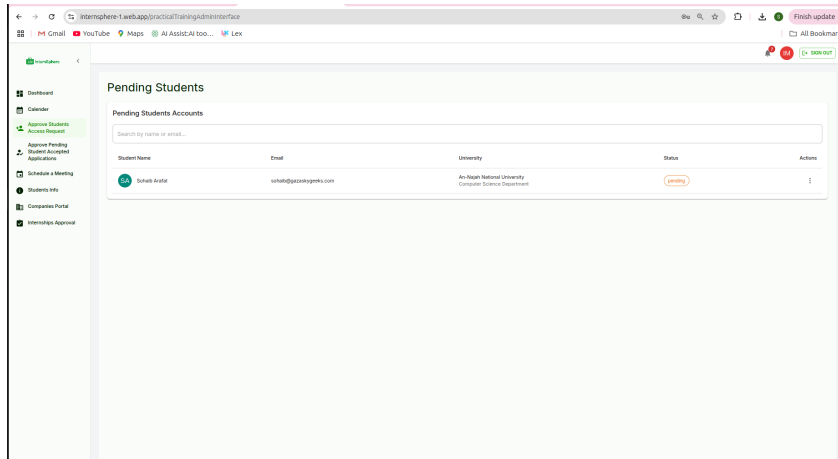


Figure 3.37: Pending Students Accounts

This page is to approve student access requests to the platform.

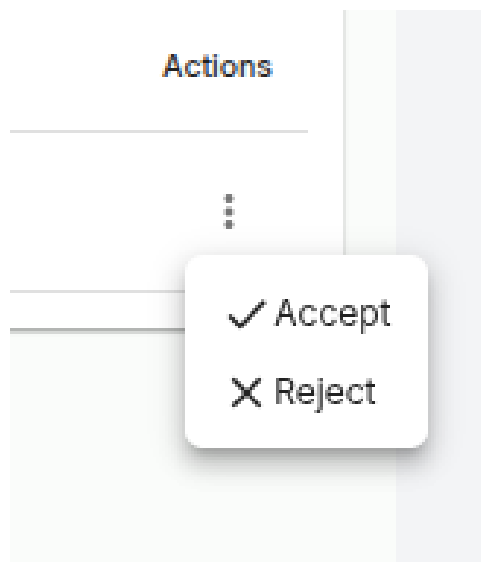


Figure 3.38: Student Accept/Reject

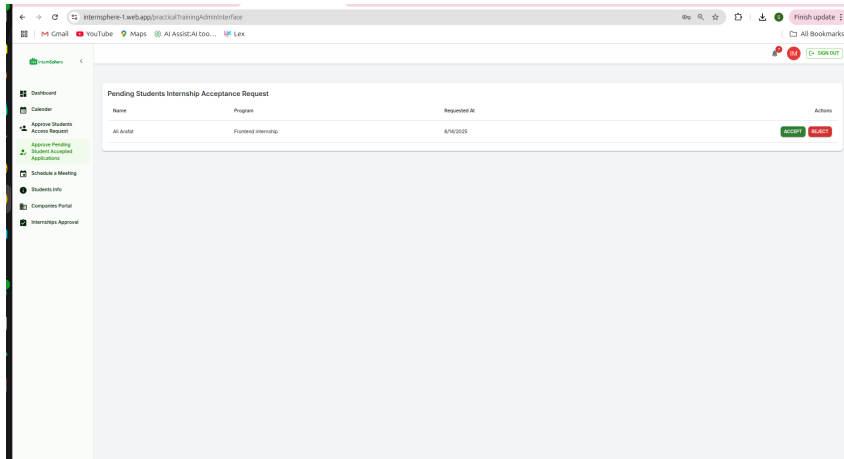


Figure 3.39: Pending Internships Application Approvals

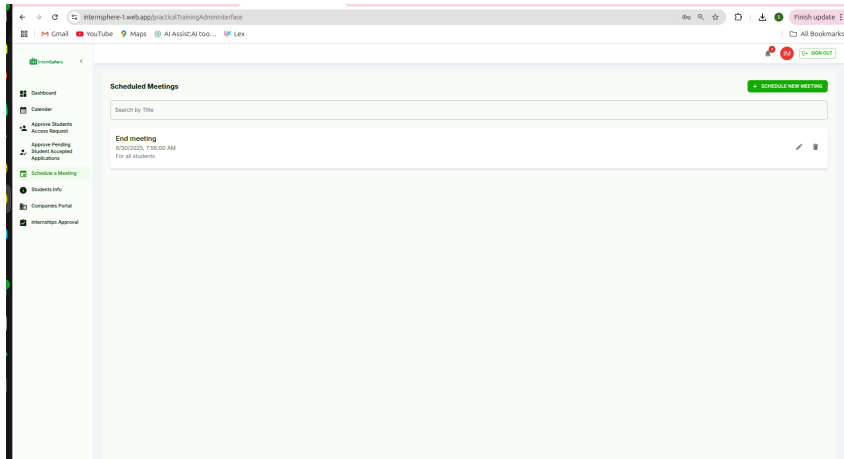


Figure 3.40: Meetings Scheduler

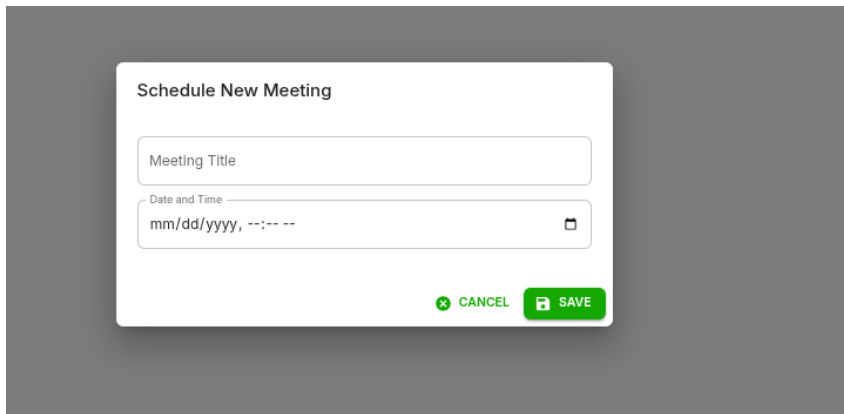


Figure 3.41: Meetings Scheduler

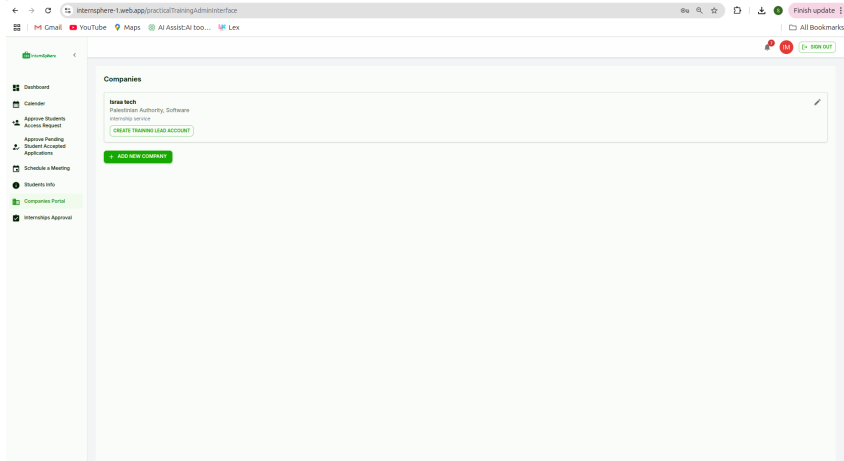


Figure 3.44: Companies Management Portal

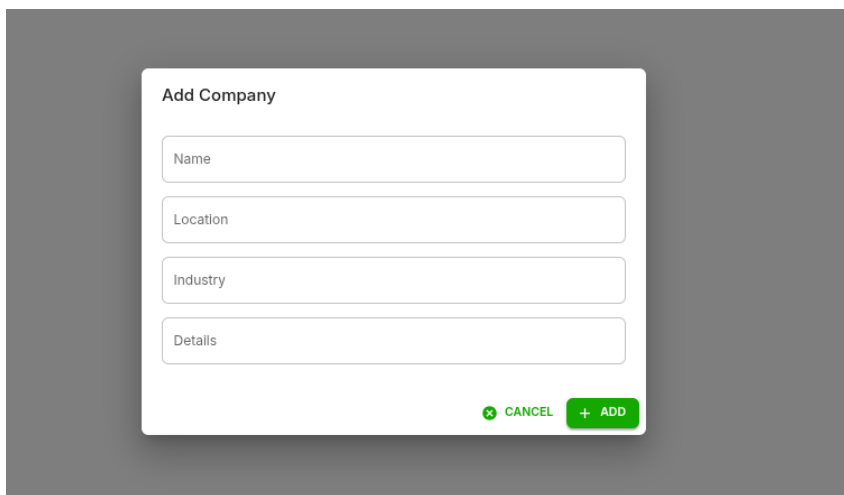


Figure 3.45: Company Addition Form

Create Training Manager (HR)

First Name

Last Name

Email

Phone Number

✕ CANCEL + CREATE

Figure 3.46: Training Manager Account Creator

Pending Internship Approvals

Title	Company	City	Mode	Actions
GA	hana tech	Noida	on-site	APPROVE REJECT

Figure 3.47: Internships Approval

3.3.5 Mobile Interfaces

InternSphere mobile Android app is a full reflection of the web app with full functionalities including file uploads and meeting client, push notifications for real-time updates, offline capability for essential features, biometric authentication for enhanced security, camera integration for document scanning, GPS location services for internship site verification, dark mode support, responsive design optimized for various screen sizes, background sync for seamless data updates, and native performance optimizations for smooth user experience across all internship management activities.

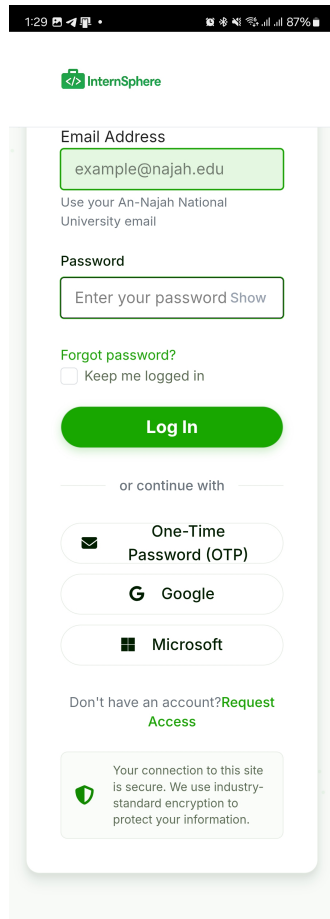


Figure 3.48: Login Page - Mobile

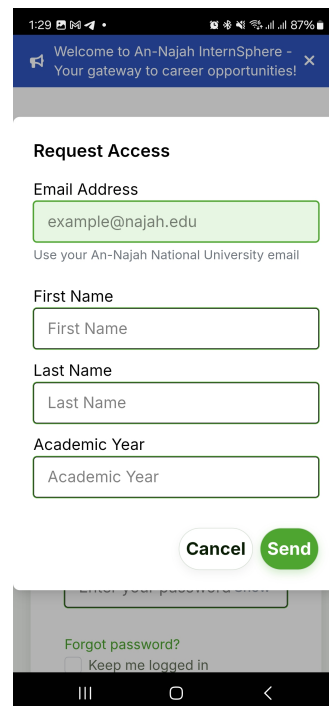


Figure 3.49: Request Access for Students - Mobile

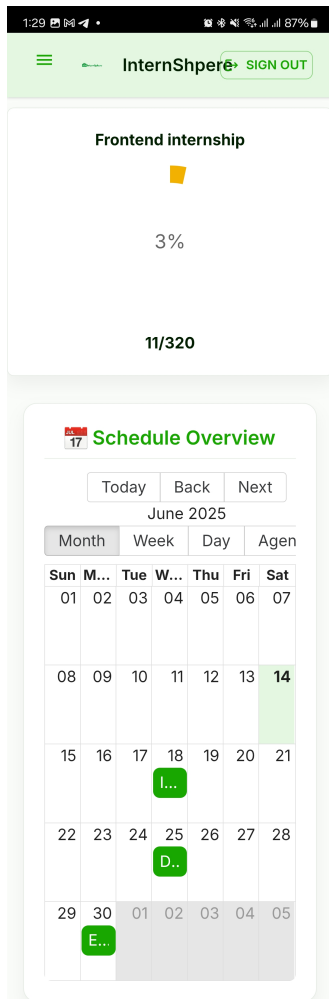


Figure 3.50: Student Dashboard - Mobile

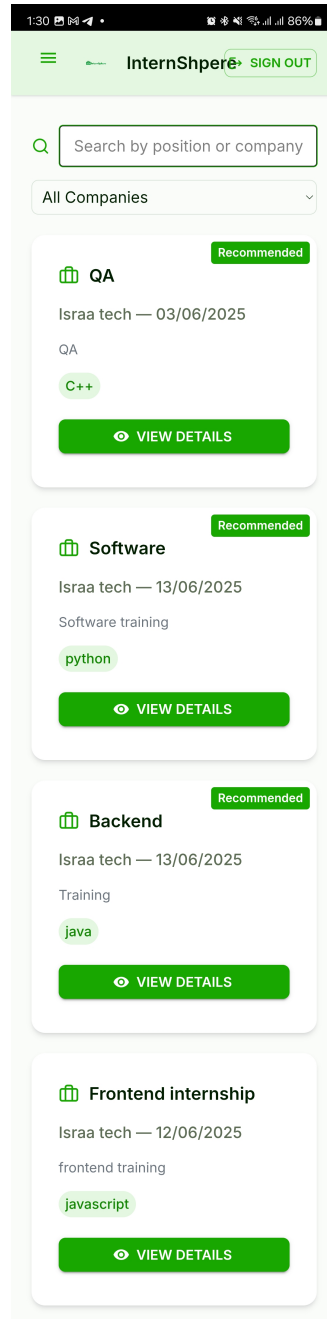


Figure 3.51: Internship Browser - Mobile

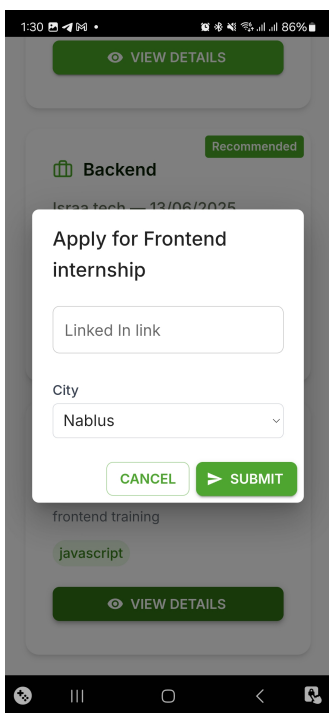


Figure 3.52: Internship Application Form - Mobile

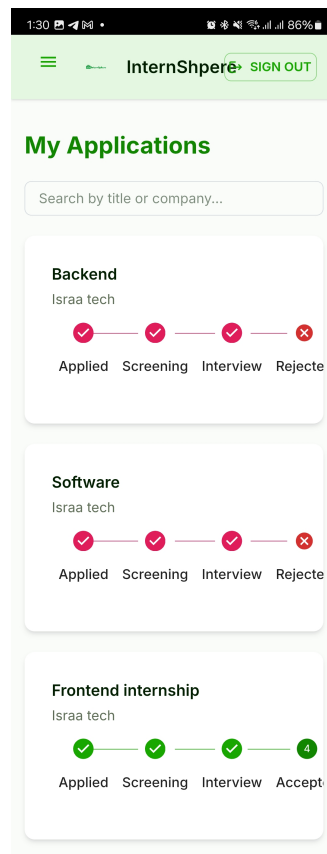


Figure 3.53: Applications Tracker - Mobile

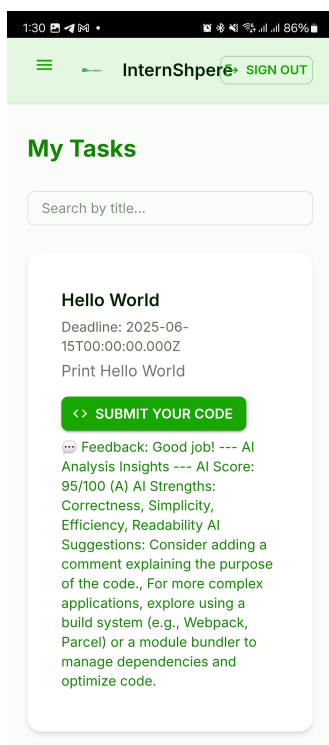


Figure 3.54: Tasks - Mobile

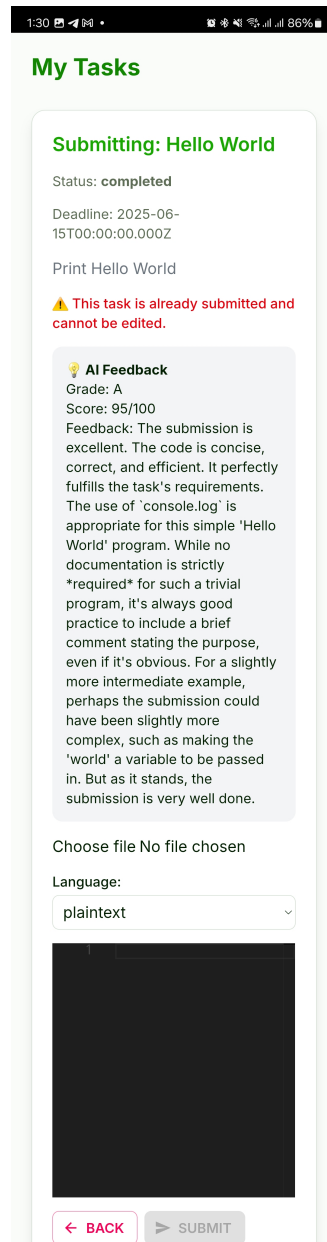


Figure 3.55: Task Submission - Mobile

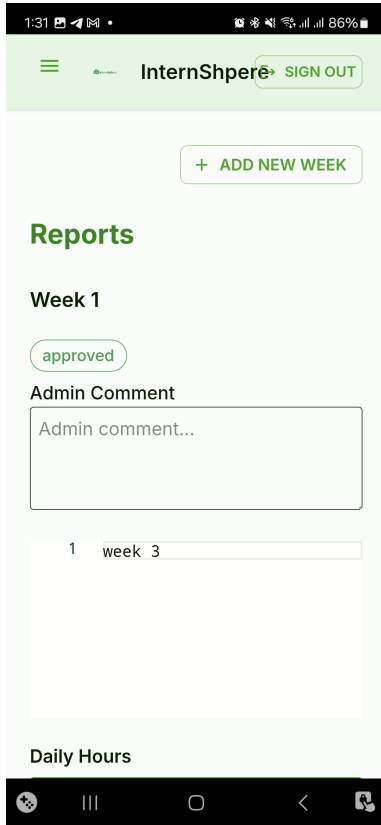


Figure 3.56: Weekly Reports Main - Mobile

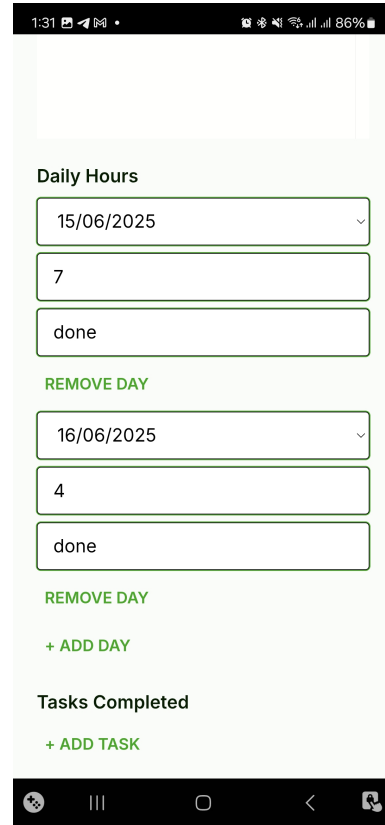


Figure 3.57: Weekly Reports Form - Mobile

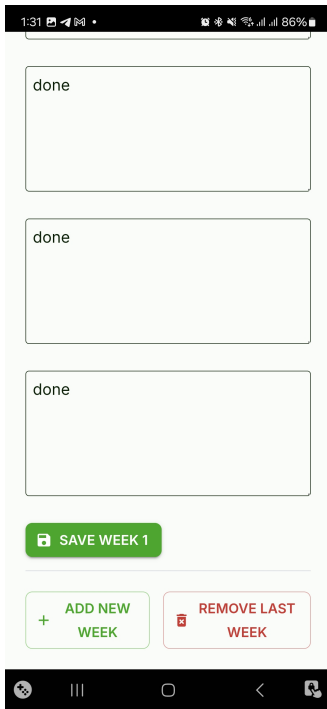


Figure 3.58: Weekly Report (Cont'd) - Mobile

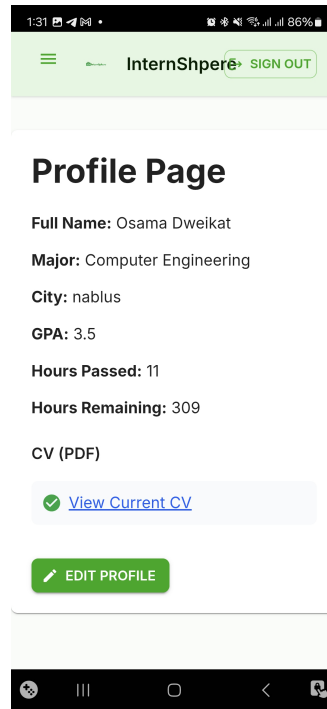


Figure 3.59: Profile Page - Mobile

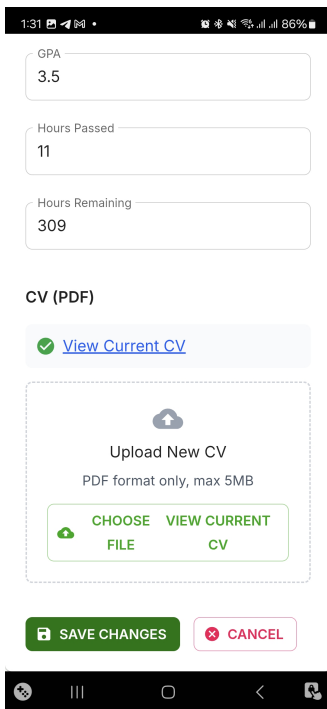


Figure 3.60: Resume Submission - Mobile

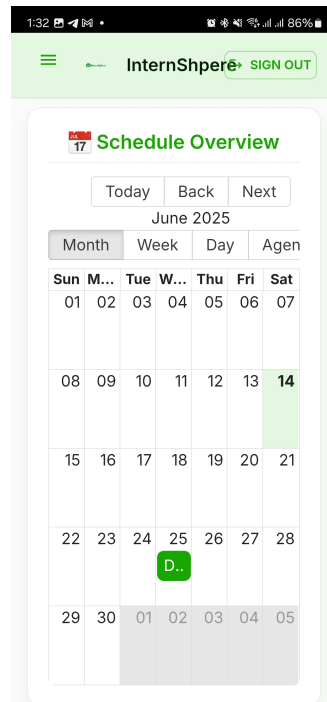


Figure 3.61: Meetings Calendar (Mentors) - Mobile

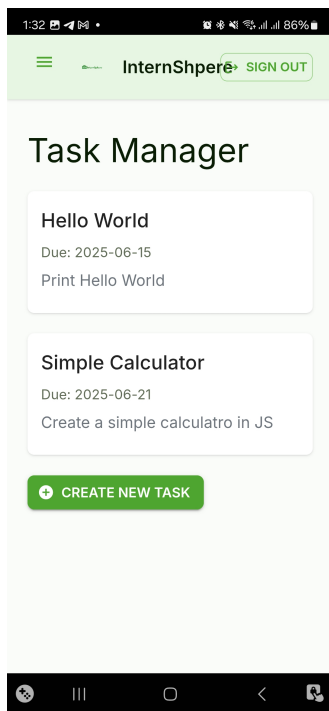


Figure 3.62: Tasks Manager - Mobile

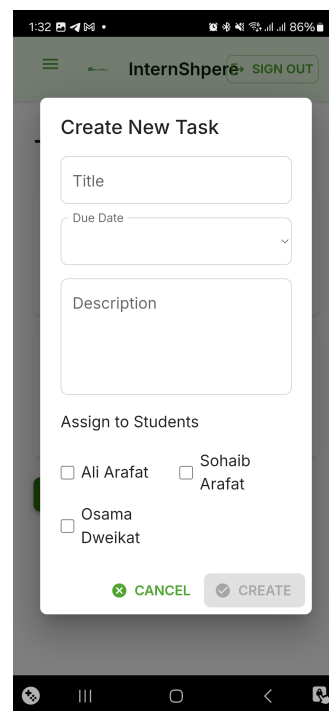


Figure 3.63: Task Creation Form - Mobile

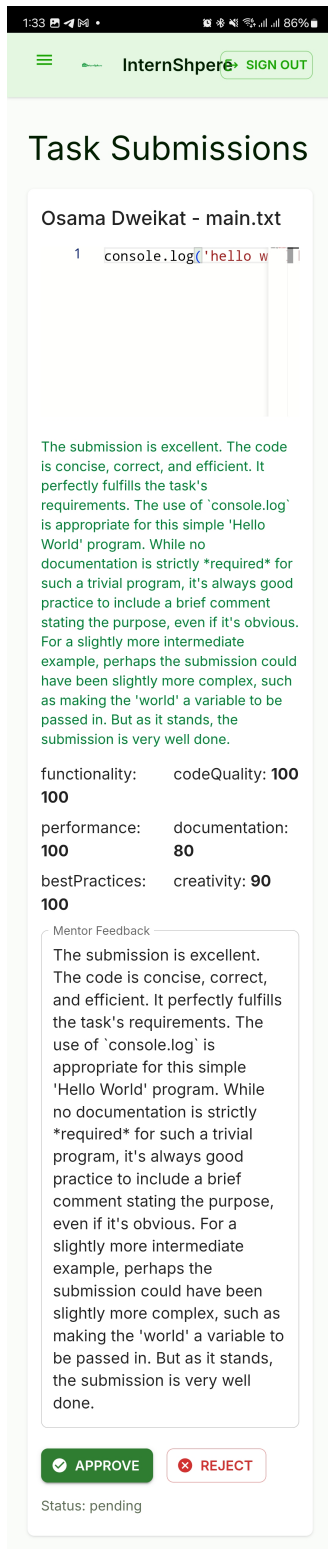


Figure 3.64: Task Submissions - Mobile

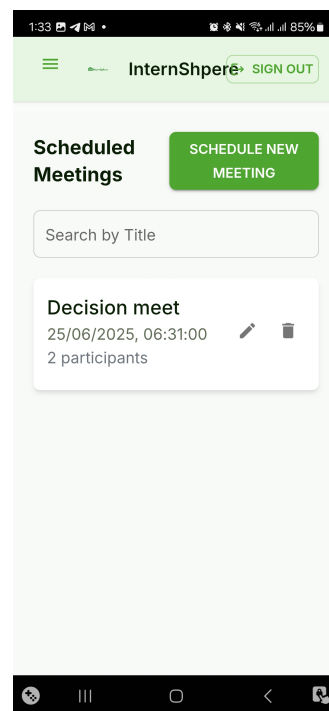


Figure 3.65: Meetings Scheduler - Mobile

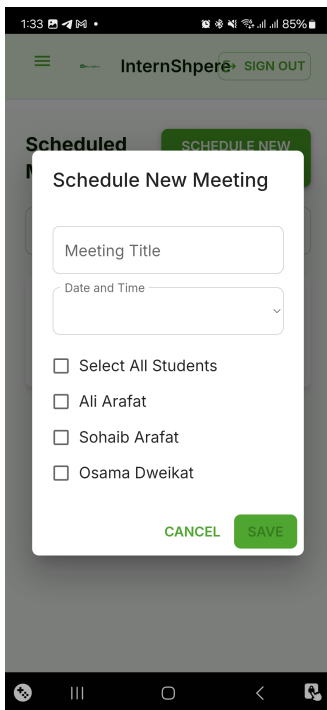


Figure 3.66: Meeting Scheduler Form - Mobile

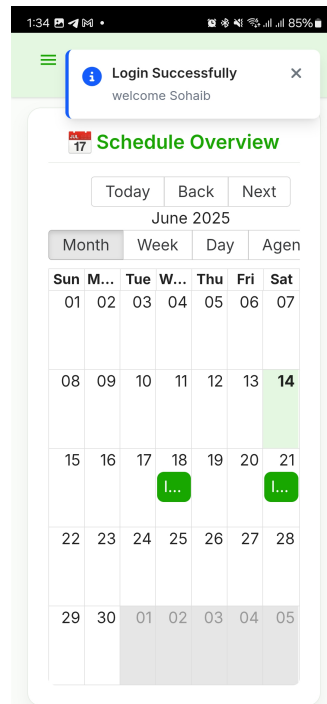


Figure 3.67: Meetings Calender (Training Lead) - Mobile

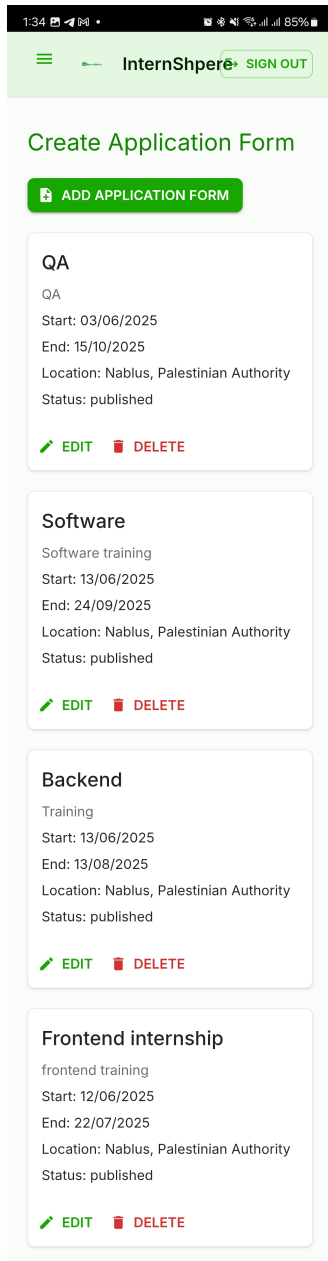


Figure 3.68: Internships Creation Page - Mobile

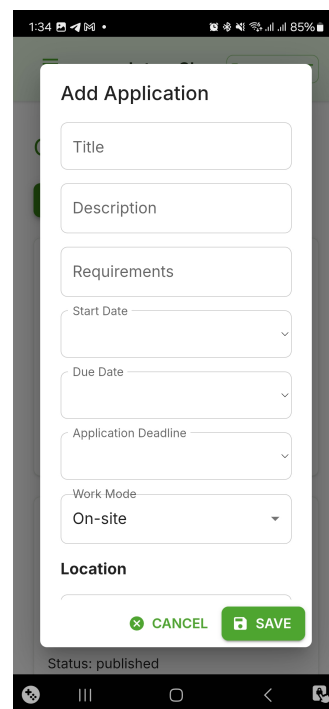


Figure 3.69: Internships & Application Forms Creator - Mobile

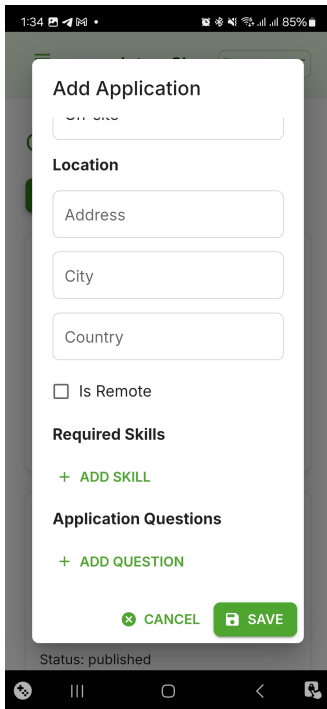


Figure 3.70: Internships & Application Forms Creator (Cont'd) - Mobile

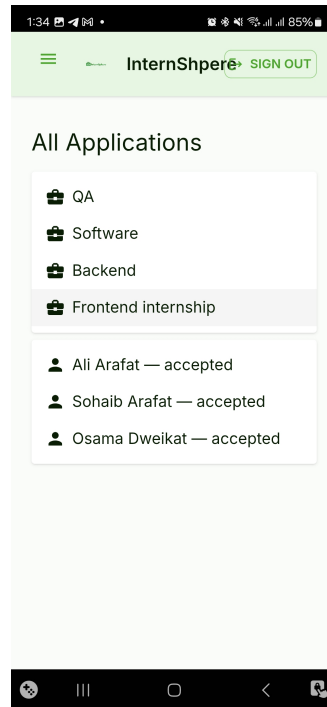


Figure 3.71: Student Applications Tracker - Mobile

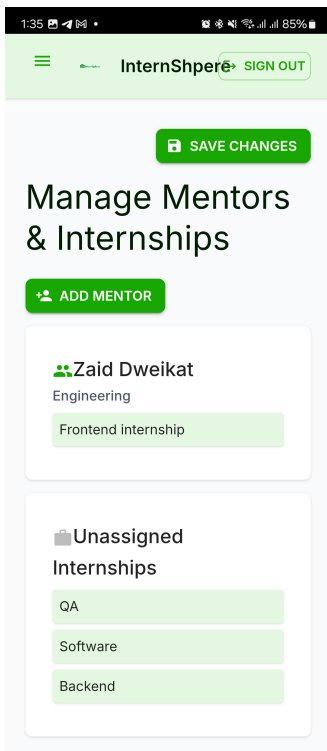


Figure 3.72: Mentors Management - Mobile

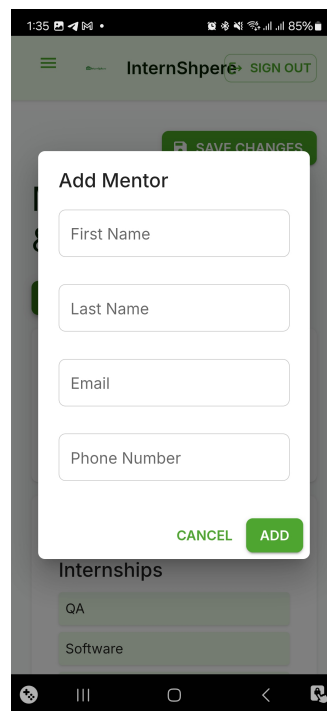


Figure 3.73: Mentors Addition Form - Mobile

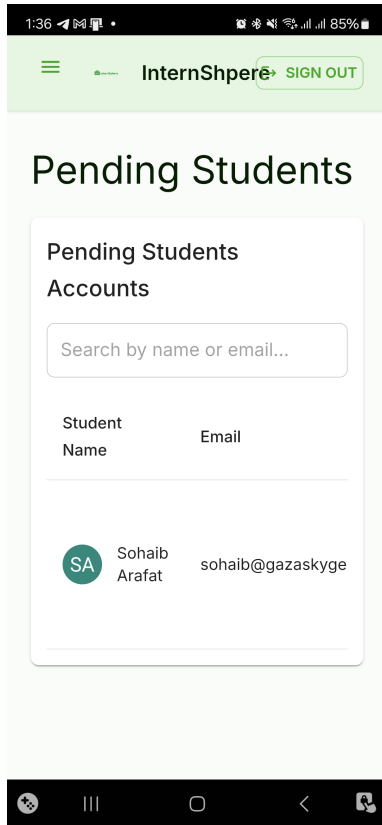


Figure 3.74: Pending Students Accounts - Mobile

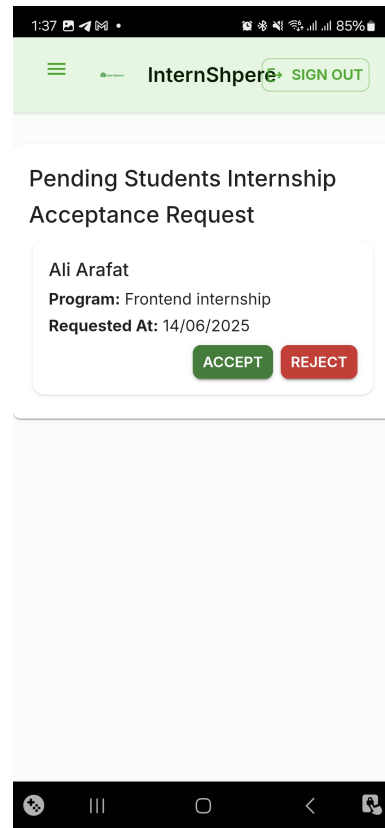


Figure 3.75: Pending Internships Application Approvals - Mobile

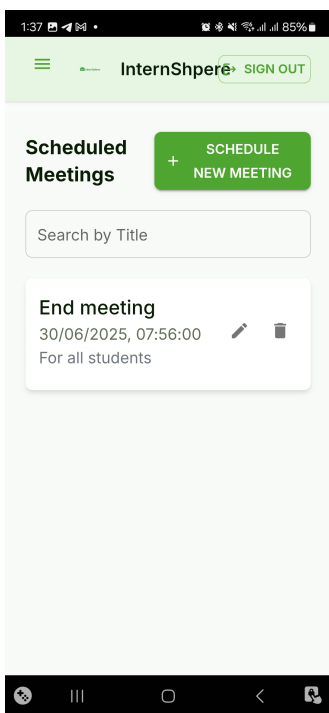


Figure 3.76: Meetings Scheduler - Mobile

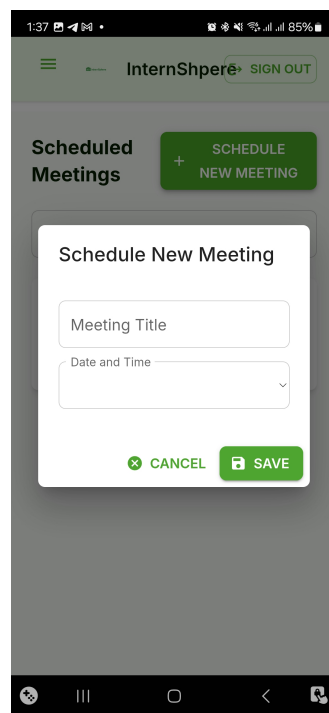


Figure 3.77: Meetings Scheduler Form - Mobile

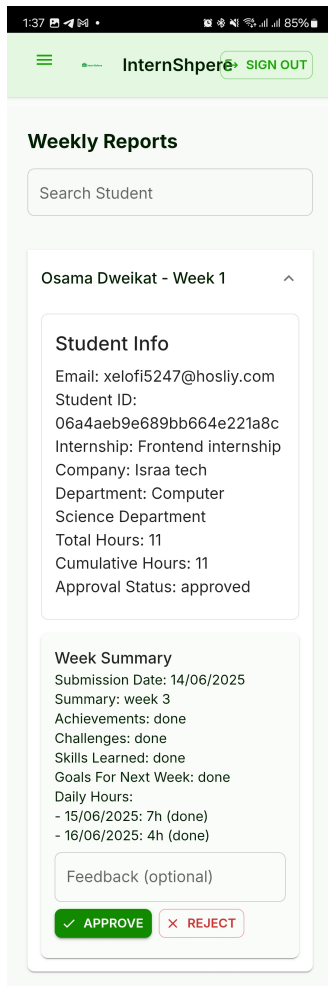


Figure 3.78: Weekly Reports and Students Info - Expanded - Mobile

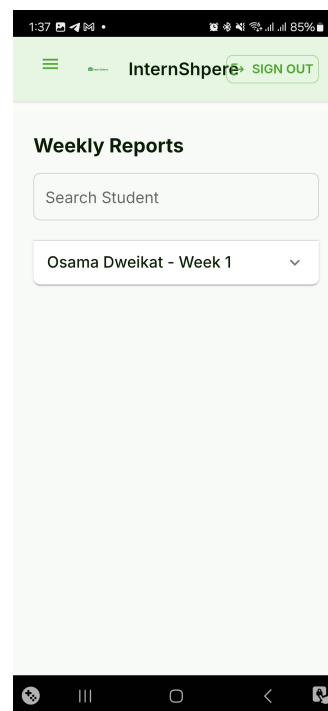


Figure 3.79: Weekly Reports and Students Info - Mobile

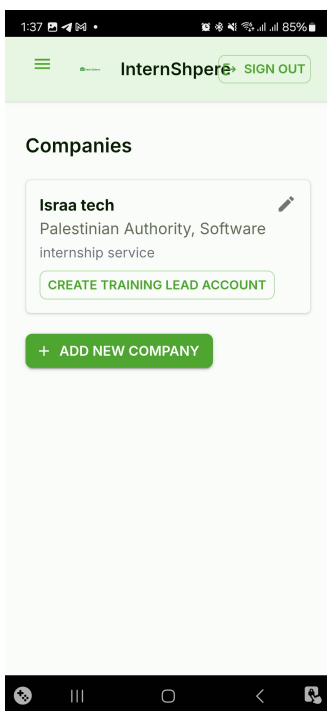


Figure 3.80: Companies Management Portal - Mobile

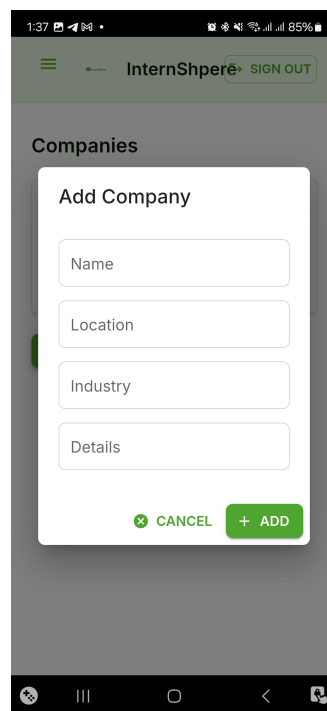


Figure 3.81: Company Addition Form - Mobile

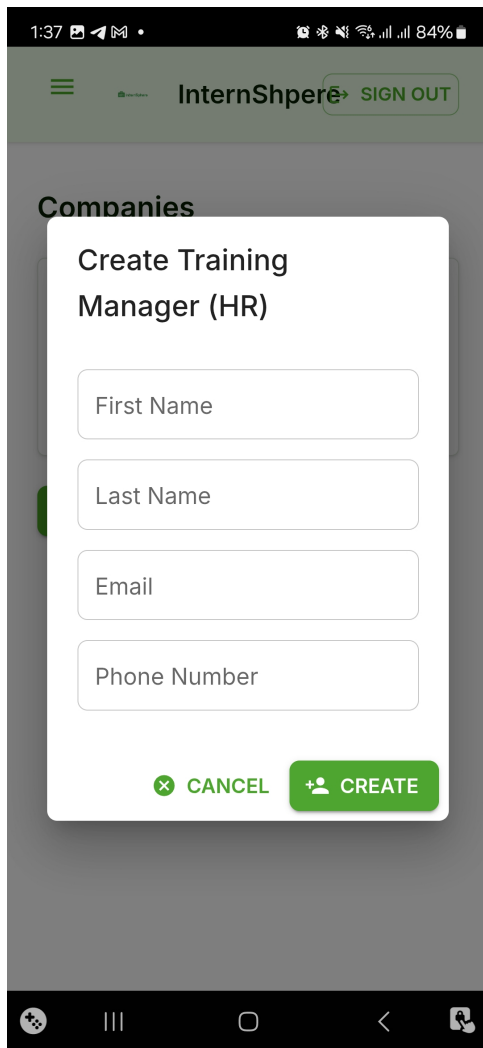


Figure 3.82: Training Manager Account Creator - Mobile

3.3.6 System Notifications

InternSphere has different types of notifications covering tasks assignments, applications, approvals, meetings and more. The notifications are either in-app or email-generated. There are event-based notifications and time-based ones. The event-based happens when a certain event occurs in system and the time-based is a continuous triggering of dates and deadlines using cron tab.

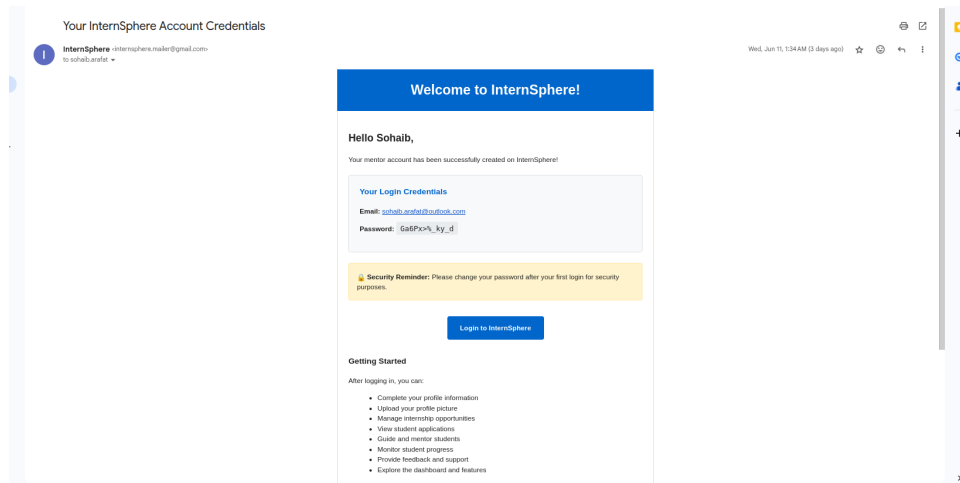


Figure 3.83: Account Creation Email with Credentials

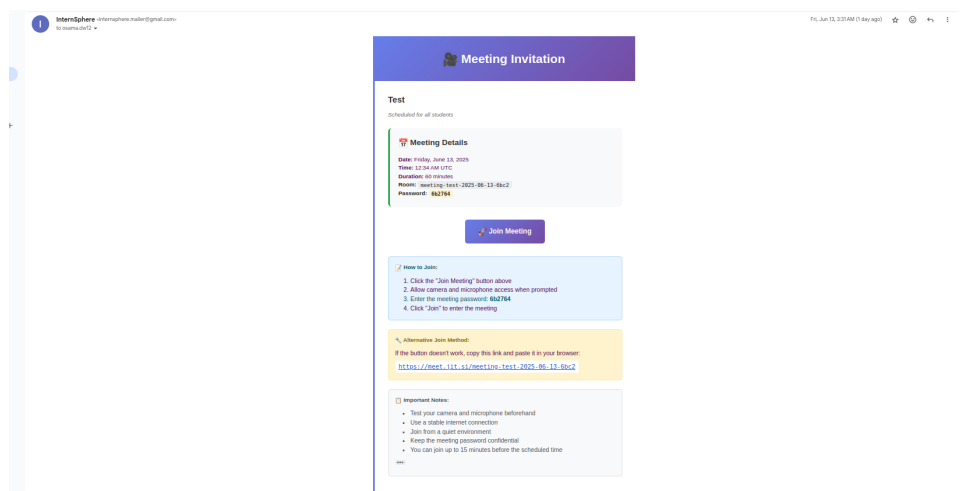


Figure 3.84: Meeting Invite

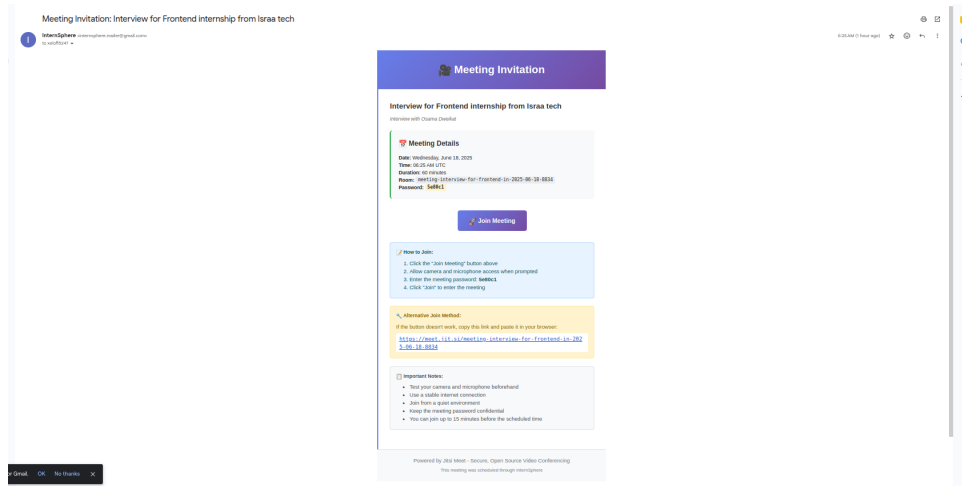


Figure 3.85: Interview Invite

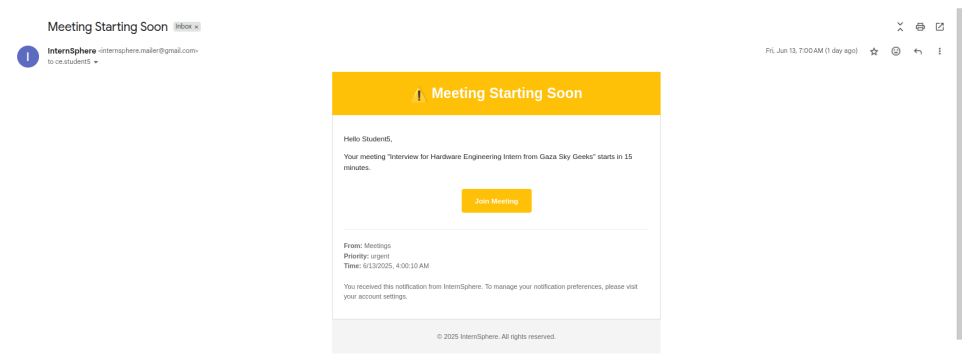


Figure 3.86: Meeting Starting Soon

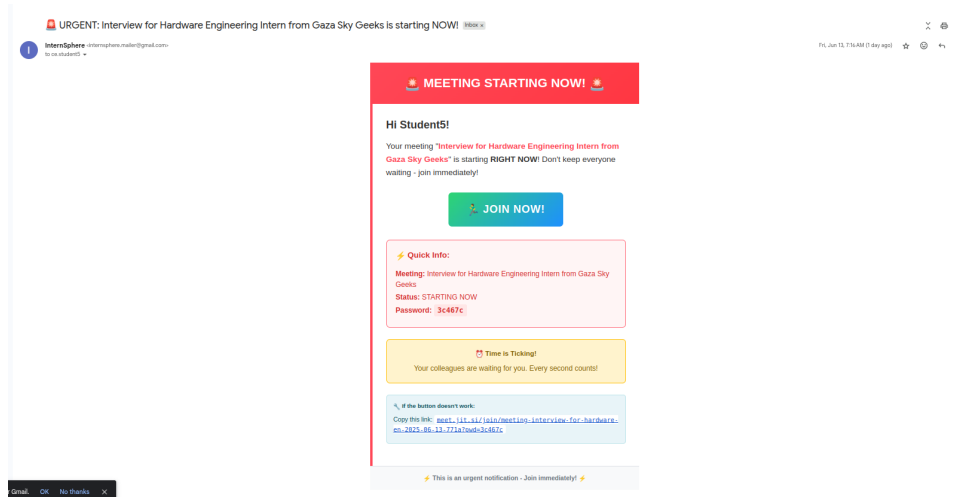


Figure 3.87: Meeting Starting Now

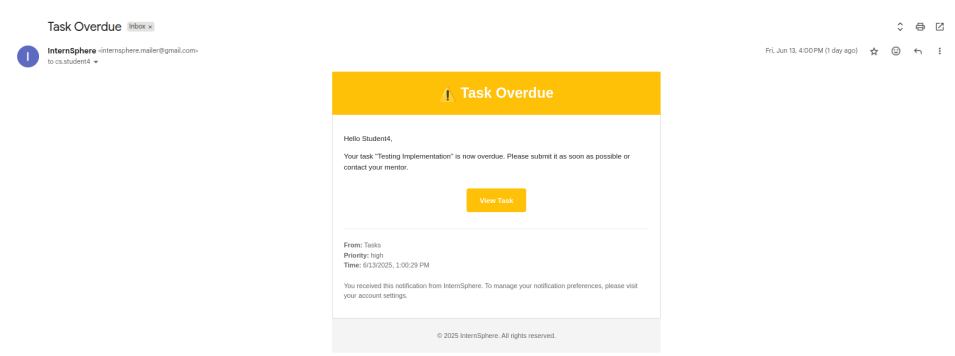


Figure 3.88: Task Overdue

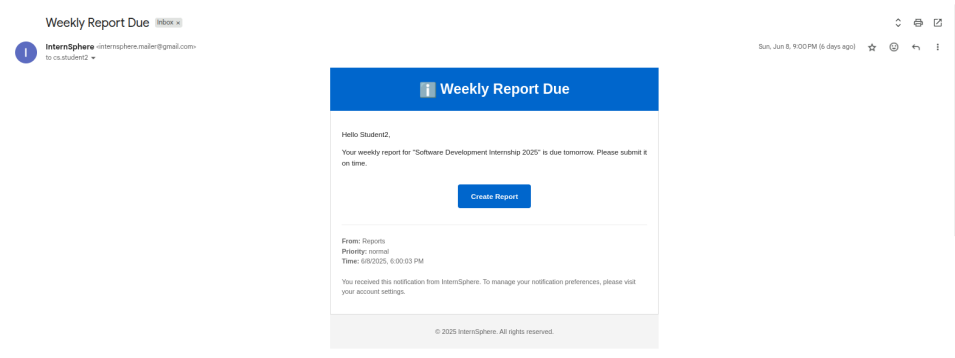


Figure 3.89: Weekly Report Overdue

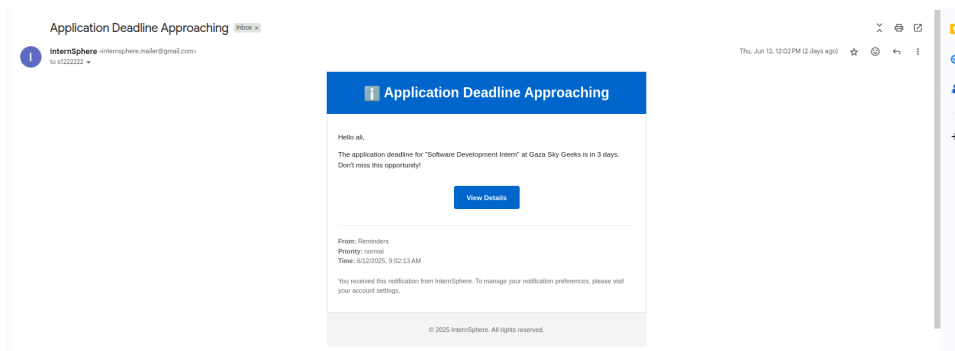


Figure 3.90: Application Deadline Approaching

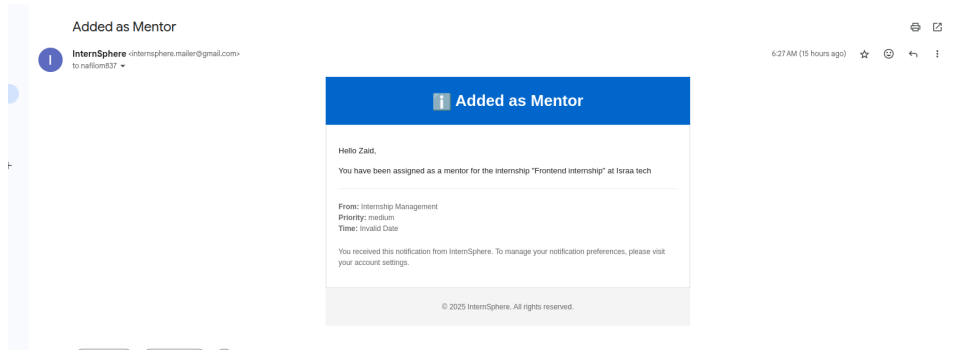


Figure 3.91: Mentor Internship Assignment

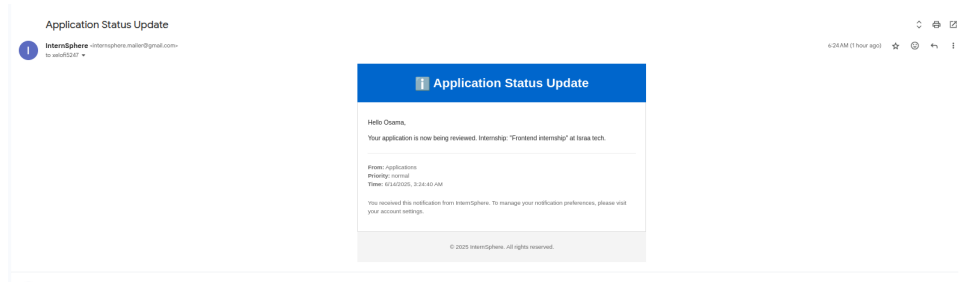


Figure 3.92: Application Review

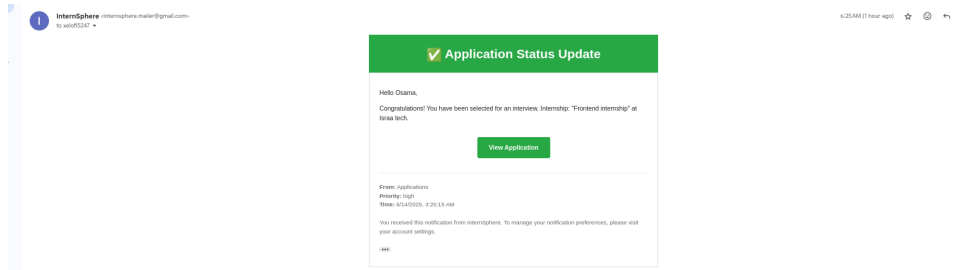


Figure 3.93: Application Interview Stage

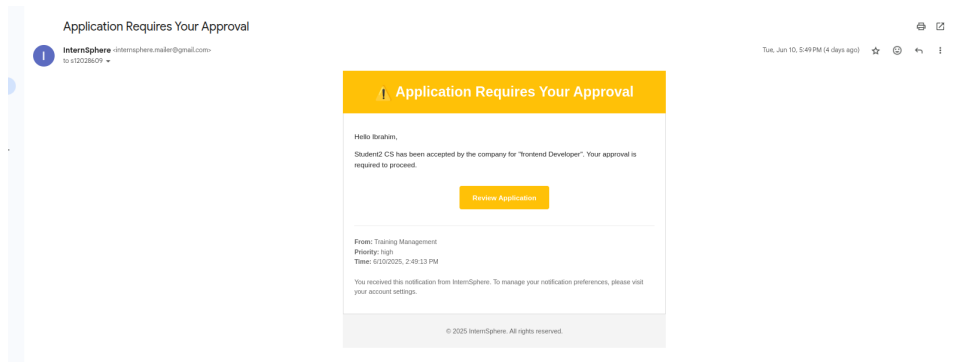


Figure 3.94: Application Requires Admin Approval

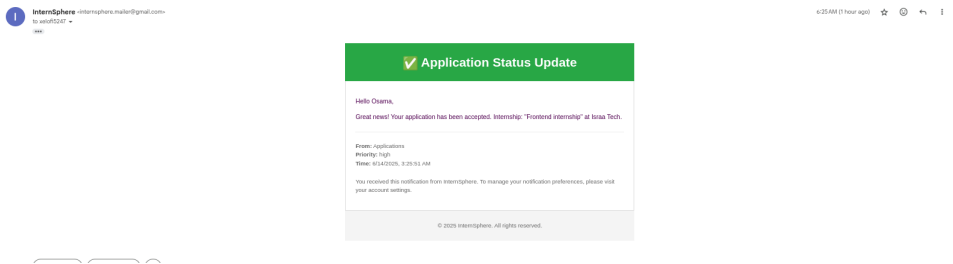


Figure 3.95: Application Initial Acceptance

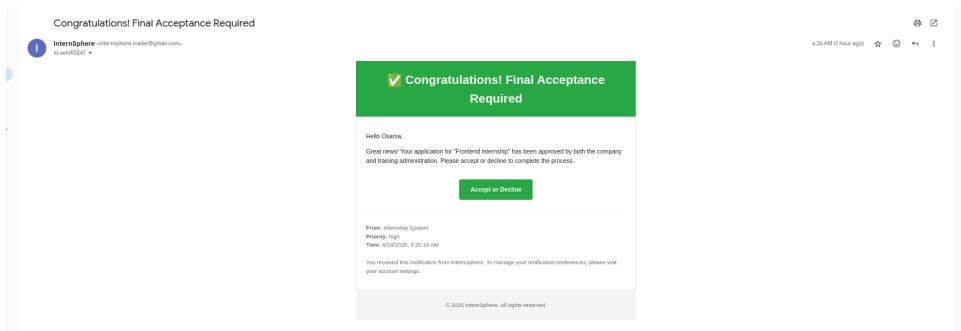


Figure 3.96: Application Final Acceptance - Pending Student Approval

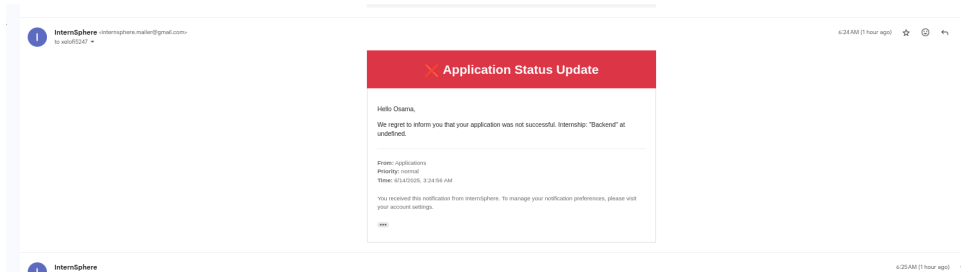


Figure 3.97: Application Rejected

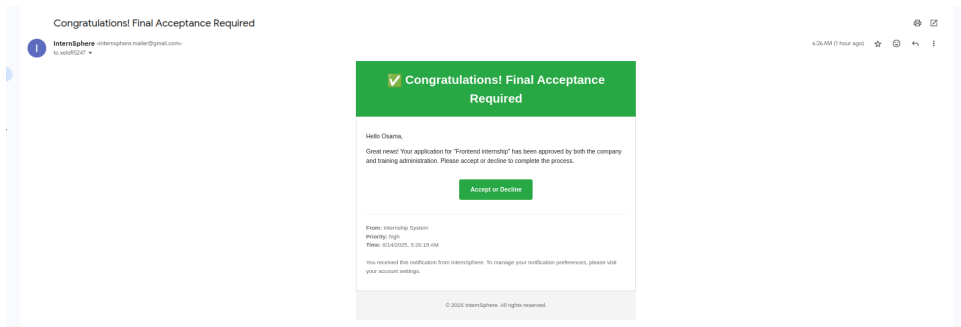


Figure 3.98: Full Completion of Mandatory Training Hours

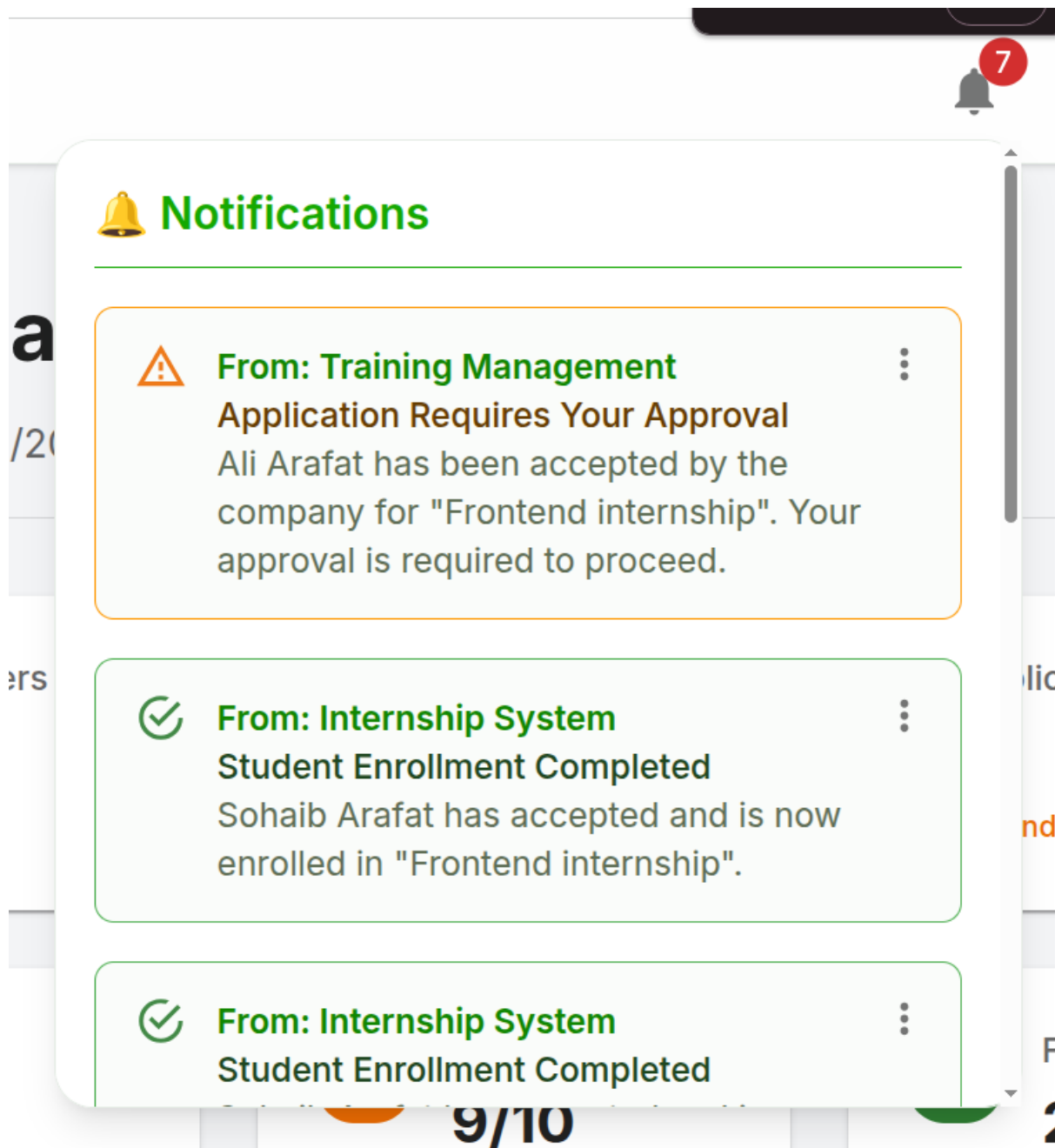


Figure 3.99: Notifications Tray

4 Discussion and Results

4.1 Results

The InternSphere platform development produced an extensive internship management system which effectively solved the IT program administration challenges that were identified. The following results show how the project objectives were achieved through the implemented features and system capabilities.

4.1.1 System Implementation

The platform successfully integrates modern web technologies with cloud infrastructure:

- The system uses a complete microservices architecture which includes Gateway API and User Management and Internship Management and AI services.
- The system uses Google Cloud Run with Docker containerization to achieve scalable serverless architecture.
- The system supports cross-platform deployment through its React.js web application and Capacitor.js mobile application.

4.1.2 Feature Development

All planned system features were successfully implemented and integrated:

- The system enables multiple users to access five different roles through proper access restrictions.
- The system implements a three-stage acceptance process which includes workflow management and notification systems.
- The system includes a complete task management platform which integrates code review tools and file upload functionality.
- The system provides real-time communication through multiple notification channels that deliver messages through in-app notifications and email and SMS messages.

4.1.3 AI Integration

Artificial intelligence capabilities successfully integrated throughout the platform:

- Google Gemini 2.5 Flash API integration completed with specialized system instructions for educational contexts
- The application rating system provides automated analysis of student applications and skill matching.
- The code evaluation system provides automated syntax checking and quality assessment feedback.

- Internship recommendation engine created for intelligent matching between students and opportunities

4.1.4 Technical Achievements

The development process achieved significant technical milestones:

- The database architecture uses MongoDB with Mongoose ODM to support complex educational data structures.
- Security framework established with JWT authentication, role-based access control, and data encryption
- The system uses Google Cloud Storage for file management and static web hosting through its cloud storage integration.
- API architecture developed with comprehensive RESTful endpoints supporting all system functionalities

4.2 Discussion

InternSphere reached its main targets through successful implementation which resulted in better IT internship management efficiency and reduced administrative costs and improved stakeholder communication. The AI system reached its target accuracy goals for application matching and code evaluation while offering scalable automation features. The microservices architecture together with Google Cloud Platform deployment confirmed the technical approach by showing both present operational needs and future growth capabilities. All stakeholder groups confirmed the platform's effectiveness in solving real-world problems through their positive feedback about improved experiences compared to traditional methods.

4.2.1 Development Challenges and Solutions

During the development process, two significant challenges emerged that required innovative solutions:

Challenge 1: AI Service Integration and Response Time Optimization

The first implementation of Google Gemini 2.5 Flash API produced response times that varied between 45 seconds and more for complex application analysis which resulted in negative impacts on user experience and system usability. The API processing requirements for complete application evaluation together with the need to preserve real-time user interaction expectations caused this challenge.

Solution: We developed a multi-layered optimization approach which combined background job queues with asynchronous processing and intelligent prompt engineering to minimize API payload size and implemented a complete caching system for repeated analysis requests. The development of fallback scoring algorithms enabled preliminary assessments to be provided immediately while AI analysis processed in the background.

The implemented solution decreased average response times to 28 seconds without compromising accuracy.

Challenge 2: Cross-Platform Mobile Deployment and Native Feature Integration

The deployment of React.js web application as a native mobile application through Capacitor.js required overcoming major obstacles to access device-specific features including push notifications and file system access and camera functionality while preserving consistent user experience between web and mobile platforms.

Solution: We built a modular feature detection system which handles platform-specific capabilities with ease and implemented progressive enhancement strategies. The solution required developers to create platform-specific service layers that abstract native functionality and implement responsive design patterns for different screen sizes and interaction methods and develop a unified notification system which functions without interruption between web browsers and mobile devices. The approach led to successful APK generation with complete native functionality and preserved code reusability between platforms.

5 Conclusions and Recommendations

5.1 Conclusions

The proposed methodology proves effective for InternSphere because it simplifies the internship process for all stakeholders. The system results show substantial enhancements in application efficiency and student-company matching and administrative oversight while maintaining system reliability and user satisfaction. The research findings provide essential knowledge for optimizing internship management operations and create a base for upcoming educational technology projects. The implemented approach provides useful advantages which educational institutions can easily implement to enhance their current internship coordination and management systems.

5.2 Recommendations and Future Work

The following recommendations will improve InternSphere's effectiveness based on this implementation. The design should start with mobile devices to enhance student accessibility while API development for university and HR system integration would decrease administrative work. The implementation of AI capabilities that include personality assessments and soft skills evaluation would boost matching accuracy and stakeholder training programs that cover all institutions would lead to uniform adoption. Future research should concentrate on blockchain technology for secure credential verification and virtual reality integration for remote internship experiences and predictive analytics to identify at-risk students. The platform should expand its capabilities through multi-language support for diverse populations and alumni network integration for mentorship and partnerships with certification bodies to align internship experiences with industry-recognized credentials.

References

- Auth0. (2024). Json web tokens introduction [Retrieved January 21, 2025, from <https://jwt.io/introduction>].
- Google DeepMind. (2023). Gemini: A family of highly capable multimodal models [Retrieved January 22, 2025, from <https://deepmind.google/technologies/gemini/>].
- Johnson, M. A., Smith, S. K., & Davis, R. L. (2021). Automated internship management: Reducing administrative overhead in computer science programs. *Journal of Computing Education*, 15(3), 245–262. <https://doi.org/10.1016/j.jce.2021.03.015>
- Kumar, R., Thompson, L. A., & Anderson, M. J. (2023). Ai-powered student-internship matching: Accuracy and implementation challenges. *Computers & Education*, 198, 104–117. <https://doi.org/10.1016/j.compedu.2023.104752>
- Zhang, W., Kumar, P., & Rodriguez, C. (2021). Machine learning approaches for automated code assessment in educational environments. *IEEE Transactions on Education*, 64(2), 187–195. <https://doi.org/10.1109/TE.2020.3025847>