



AN-NAJAH NATIONAL UNIVERSITY
FACULTY OF ENGINEERING
AND INFORMATION TECHNOLOGY
COMPUTER ENGINEERING DEPARTMENT

Software Graduation Project:

UniServe



SUBMITTED BY

Shahd Mohammed Alawneh

Yasmine Ahmad Saad

SUPERVISOR

Dr. Anas Toma

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

January, 2026

Acknowledgment

We would like to express our sincere appreciation to everyone who contributed to the successful completion of this graduation project.

We extend our special gratitude to An-Najah National University for providing a supportive academic environment and the resources that enabled us to develop our skills and contribute positively to our community.

Our deepest appreciation and gratitude go to our esteemed supervisor, Dr. Anas Toma, whose guidance, dedication, and continuous support were instrumental throughout the development of this project. Dr. Toma generously shared his time, expertise, and valuable insights, guiding us through challenges and encouraging us to overcome obstacles with confidence and determination. His constructive feedback, patience, and genuine interest in our progress greatly enhanced both the quality of this work and our learning experience.

We are truly grateful for his mentorship, encouragement, and trust in our abilities, which played a significant role in motivating us to achieve our best.

Finally, we would like to thank everyone who supported and encouraged us during this journey. This project stands as a testament to the teamwork, cooperation, and assistance we received, and we present it with sincere gratitude.

Dedication

This work is dedicated to everyone who contributed, directly or indirectly, to its completion, regardless of how small the contribution may have been. We dedicate this project to our families and friends, whose continuous support, patience, and encouragement made this journey possible, and to all those whose passion for knowledge inspired us along the way

DISCLAIMER

This report has been prepared by student(s) from the Computer Engineering Department, Faculty of Engineering, at An-Najah National University. The content of this report may contain grammatical or technical inaccuracies, as it has not undergone professional editing beyond revisions made following the assessment process. The views, conclusions, and recommendations expressed herein are solely those of the student(s).

An-Najah National University assumes no responsibility or liability for any consequences resulting from the use of this report for purposes other than those for which it was originally intended.

Contents

1	Introduction.....	15
2	Constraints and Earlier Coursework	16
2.1	Constraints	16
2.1.1	Difficulty in Information Gathering.....	16
2.1.2	Seamless Work Transfer Between Team Members.....	16
2.2	Earlier Coursework	16
2.2.1	Object-Oriented Programming.....	16
2.2.2	Database Design.....	17
3	Methodology	18
3.1	Tools, Methods and Programming Languages.....	18
3.1.1	Tools.....	18
3.1.2	Programming Language.....	19
	Back end:	19
	Front end:	19
3.1.3	Database.....	20
3.2	System Features Implementation.....	28
3.2.1	Mobile application	28
	Welcome Page: Start Page:	28
	User Registration and Authentication Process:.....	29
	User Sign-In Process:.....	30
		30
	Forgot Password Feature:	31
	Profile Page:.....	32
	Profile Edit Page:	35
	Messaging Feature:	37
	Notification System Feature:	38
	Student Guide Feature:	40
	Activities Management Feature:	42
	Custom Requests Feature:.....	45

Student Progress:	46
AI Features:.....	47
Forms Submissions:	49
Search and AI Assistance Feature:.....	50
FAQs & Student Questions Feature:.....	51
Calendar Feature:	53
Doctor Feature: Student Evaluation and Monitoring:.....	54
Roles & Permissions Management Feature:	55
Confirm Student Feature:.....	57
Assigning Students to Doctors Feature:.....	57
3.2.2 Website:.....	58
Student Role:.....	60
Center Role:	68
Doctor Role:.....	78
Admin Role:.....	79
4 Results and Analysis	86
4.1 System Functionality Results.....	86
4.2 Progress Tracking and Evaluation Results.....	86
4.3 AI-Based Recommendation Results (Student Side)	86
4.4 AI-Supported Request Analysis Results (Service Center Side)	87
4.5 Roles and Permissions Analysis.....	87
4.6 Error Estimation and System Reliability	87
4.7 Summary of Results	87
5 Discussion	88
5.1 Problem Resolution.....	88
5.2 Project Contributions	88
5.3 Implications of the Results.....	88
5.4 Limitations	88
5.5 Future Applications	88
6 Conclusions and Recommendation	89
6.1 Conclusions	89

6.2 Recommendations for Improvement.....	89
6.3 Lessons Learned.....	90
6.4 Future Work and Open Problems.....	90
7 References:.....	91

List of Figures

Figure3. 1: Android Studio	18
Figure3. 2: Visual Studio Code (VS Code).....	18
Figure3. 3: GitHub	18
Figure3. 4: Firebase	18
Figure3. 5: MySQL	18
Figure3. 6: Postman	18
Figure3. 7: Google Drive	19
Figure3. 8: Ollama AI	19
Figure3. 9: Node JS.....	19
Figure3. 10: Flutter	19
Figure3. 11: UniServe Database Schema.....	20
Figure3. 12: Users Table	20
Figure3. 13: Students Table	21
Figure3. 14: Doctors Table.....	21
Figure3. 15: Center Table.....	21
Figure3. 16: Services Table.....	22
Figure3. 17: Student Services Table	22
Figure3. 18: Student Status Table	22
Figure3. 19: Activity Submissions Table	23
Figure3. 20: Students Hours Summary Table	23
Figure3. 21: Service Form Table.....	23
Figure3. 22: Students Custom Requests Table.....	24
Figure3. 23: Volunteer Requests Table	24
Figure3. 24: Student Questions Table	24
Figure3. 25: FAQs Table	25
Figure3. 26: Messages Table.....	25
Figure3. 27: Notifications Table.	25
Figure3. 28: Remainders Table.	26
Figure3. 29: Device Tokens Table.	26
Figure3. 30: Roles Table.	26
Figure3. 31: Permissions Table.....	26
Figure3. 32: Role Permissions Table	27
Figure3. 33: Admin Activity Log Table.	27
Figure3. 34: Student Doctor Table.....	27
Figure3. 35: Welcome Page	28
Figure3. 36: Start Page.....	28
Figure3. 37: Sign Up.....	29
Figure3. 38: Center Sign In.....	30

Figure3. 39: Blocked Sign in	30
Figure3. 40: Send Verification Code.....	31
Figure3. 41: Center Home Page.....	32
Figure3. 42: Student Home Page1.	32
Figure3. 43: Student Home Side Par.....	33
Figure3. 44:Admin Dashboard1.....	33
Figure3. 45: Doctor Home Side Par.....	33
Figure3. 46: Admin Dashboard2.....	34
Figure3. 47:Admin Dashboard1.....	34
Figure3. 48: Admin Dashboard3.....	34
Figure3. 49: Center Edit Profile Page	35
Figure3. 50: Center Profile View Page.	35
Figure3. 51: Student Profile View.....	35
Figure3. 52: Student Edit Profile	35
Figure3. 53: Doctor Profile View.....	36
Figure3. 54: Set Or Change Photo Page.	36
Figure3. 55: Doctor Edit Profile.	36
Figure3. 56:A conversation between a student and the center.	37
Figure3. 57: Center All Messages.....	37
Figure3. 58:Messages between the student and the service center ("student's side").....	37
Figure3. 59: Web Notification.	38
Figure3. 60: Mobile Notification.	38
Figure3. 61: Volunteer Requests Notifications.	39
Figure3. 62:ALL Center Notifications.....	39
Figure3. 63: All Student Notifications.....	39
Figure3. 64: Custom Requests Notifications.	39
Figure3. 65: Student Notifications Categories.....	39
Figure3. 66: Notifications Overview	40
Figure3. 67: Profile Navigation.	40
Figure3. 68: Student Guide – Welcome Card	40
Figure3. 69: Navigation and Calendar Overview	40
Figure3. 70: Search Feature	41
Figure3. 71: Available Activities Overview	41
Figure3. 72: Activity Status Tracking.	41
Figure3. 73: Recent Activities.....	41
Figure3. 74: Submit Forms.	41
Figure3. 75: Add New Activity.	42
Figure3. 76: Center Update Activity Page.	43
Figure3. 77: Center View Activities Page.....	43
Figure3. 78: Center View Specific Activity.	43

Figure3. 79: Student View Activities.	44
Figure3. 80: Approvals Volunteer Requests.	44
Figure3. 81: Student Volunteer Requests List.....	44
Figure3. 82: Add Custom Requests.	45
Figure3. 83: Student Custom Service Requests Interface.....	45
Figure3. 84: Student View Its Requests.....	45
Figure3. 85: Center Custom Requests.	46
Figure3. 86: View Approvals Custom Requests.	46
Figure3. 87: Student Progress.	46
Figure3. 88: AI Recommendations.	47
Figure3. 89: AI-Powered Request Analysis Home Page.	48
Figure3. 90: AI-Powered Request Analysis Requests Page.....	48
Figure3. 91: Student Submitted Form.....	49
Figure3. 92: Student Upload File.....	49
Figure3. 93: Center View All Submissions.	49
Figure3. 94: Center View Specific Student Submission.	50
Figure3. 95: Center Approved Submissions.	50
Figure3. 96: Search Results Display.	50
Figure3. 97: Ask AI Interface.....	50
Figure3. 98: Activities Search Interface.....	50
Figure3. 99: Service Center Student Search Interface.	51
Figure3. 100: Add New FAQ.	52
Figure3. 101: Student Questions List.....	52
Figure3. 102: Students Questions.	52
Figure3. 103: FAQs Section.....	52
Figure3. 104: Ask a New Question.	52
Figure3. 105: Service Center View of All Students with Access to Individual Student Details.	53
Figure3. 106: Doctor Calendar.....	53
Figure3. 107: Student Calendar.....	53
Figure3. 108: Center Calendar.	53
Figure3. 109: Students Summary.....	54
Figure3. 110: Doctor's Students List.	54
Figure3. 111: Final Results Status.	54
Figure3. 112: Create a new role dynamically.	55
Figure3. 113: Roles & Permissions Overview..	55
Figure3. 114: Admin Permissions Management.	56
Figure3. 115: Assign roles, activate or deactivate users, and manage access.	56
Figure3. 116: Service Center Permissions Management.	56
Figure3. 117: Import Student Sheet.	57

Figure3. 118: Assign Student Screen.	57
Figure3. 119: Assigned Students List.	57
Figure3. 120: Select Doctor Screen.	57
Figure3. 121: Welcome Page Web.	58
Figure3. 122: Sign in Web	58
Figure3. 123: Sign In Center.	59
Figure3. 124: Student Home Web.	59
Figure3. 125: Student Home Web2	60
Figure3. 126: View Activities Web.	60
Figure3. 127: Student Messages Web.	61
Figure3. 128: Student Chat Web.	61
Figure3. 129: Student Notifications.	62
Figure3. 130: Student Submissions.	62
Figure3. 131: Help and Support Web.	63
Figure3. 132: Student Questions Web.	63
Figure3. 133: Add new Question.	64
Figure3. 134: Student View Profile Web	64
Figure3. 135: Student Edit Profile Web.	65
Figure3. 136: Student Suggest Activity Web.	65
Figure3. 137: Student Suggestions Web.	66
Figure3. 138: Student Progress Web.	66
Figure3. 139: Student Calendar Web.	67
Figure3. 140: Student Calendar.	67
Figure3. 141: Center Home Web.	68
Figure3. 142: Center Activities Web.	68
Figure3. 143: Update Activity Web.	69
Figure3. 144: Volunteer Requests Web.	69
Figure3. 145: Custom Requests Web.	70
Figure3. 146: Volunteer Approvals Web.	70
Figure3. 147: Custom Requests Approvals.	71
Figure3. 148: Student Submissions Web.	71
Figure3. 149: Specific Submission.	72
Figure3. 150: Student Questions.	72
Figure3. 151: Add New FAQ.	73
Figure3. 152: Center Calendar Web.	73
Figure3. 153: Service Center Profile.	74
Figure3. 154: Center Edit Profile.	74
Figure3. 155: Update Password.	75
Figure3. 156: View All Students.	75
Figure3. 157: Create New Activity1.	76

Figure3. 158: Create New Activity2.	76
Figure3. 159: : Create New Activity3.	77
Figure3. 160: Center Search Web.	77
Figure3. 161: Doctor Home Page.	78
Figure3. 162: Student Summary.	78
Figure3. 163: Doctor Calendar.	79
Figure3. 164: Admin Dashboard1.	79
Figure3. 165: Admin Dashboard2.	80
Figure3. 166: Admin Dashboard3.	80
Figure3. 167:: Admin Dashboard4.	81
Figure3. 168: Select Doctor.	81
Figure3. 169: Users Control.	82
Figure3. 170: Roles & Permissions.	82
Figure3. 171: Create New Role.	83
Figure3. 172: Student Permissions.	83
Figure3. 173: Center Permissions.	84
Figure3. 174: Doctor Permissions.	84
Figure3. 175: Admin Permissions.	85

List of Abbreviations

AI ----Artificial Intelligence.

API---Application Programming Interface.

DB----Database.

FCM--Firebase Cloud Messaging.

IDE---Integrated Development Environment.

JSON-JavaScript Object Notation.

JWT---JSON Web Token.

UI-----User Interface.

UX----User Experience.

Abstract

UniServe is a mobile and web application developed to simulate and facilitate the Community Service course at An-Najah National University. The project aims to simplify and manage the entire community service process by providing an integrated application that supports all stages of the course, starting from student registration and activity assignment, and ending with the submission of required printed documents confirming the completion of service tasks.

UniServe enables students to register for community service activities, follow up on their assigned tasks, and submit necessary files through a structured and centralized workflow. In parallel, the application assists supervisors and administrators in monitoring student progress and managing course requirements more efficiently.

By digitizing the community service workflow through a unified mobile and web application, UniServe reduces manual procedures, enhances organization, and improves the overall experience for students and academic staff.

1 Introduction

Community service courses play a vital role in enhancing students' sense of responsibility, social engagement, and practical involvement within their communities. At An-Najah National University, the Community Service course requires students to participate in approved activities, follow specific procedures, and submit official documentation to confirm the completion of their service hours. Despite the importance of this course, the current process relies heavily on manual procedures, which can be time-consuming, difficult to track, and prone to organizational challenges.

The management of community service activities involves multiple stakeholders, including students, supervisors, and administrative staff. Coordinating registrations, monitoring progress, assigning tasks, and handling documentation through traditional methods increases the likelihood of delays, miscommunication, and data inconsistencies. These challenges highlight the need for a more efficient, organized, and accessible solution that supports all participants throughout the course lifecycle.

In response to these challenges, UniServe was developed as a mobile and web application aimed at streamlining the Community Service course workflow. The application provides a unified digital environment that allows students to register for activities, track their progress, and submit required files, while enabling supervisors and administrators to efficiently manage activities and monitor student performance.

By adopting modern software technologies, UniServe seeks to enhance transparency, reduce manual effort, and improve overall efficiency in managing community service programs. This project demonstrates how a well-designed application can modernize academic administrative processes and provide a more structured and user-friendly experience for all stakeholders involved.

2 Constraints and Earlier Coursework

2.1 Constraints

Throughout the development of the **UniServe** application, several constraints were encountered that influenced the design and implementation process. Each constraint required careful analysis and practical solutions to ensure the successful completion of the project. The most significant constraints are outlined below.

2.1.1 Difficulty in Information Gathering

One of the primary challenges faced during the development of UniServe was the difficulty of gathering comprehensive and accurate information related to the Community Service course workflow. This challenge arose from the continuous changes in available activities and organizational procedures from one academic semester to another, which required the system to accommodate frequently updated requirements and dynamic course structures.

2.1.2 Seamless Work Transfer Between Team Members

Another significant constraint was ensuring a smooth and efficient transfer of work between team members, particularly given that the project involved both mobile and web application development. Coordinating tasks, managing code changes, and maintaining consistency across different system components required an effective collaboration strategy.

To address this challenge, **GitHub** was adopted as the primary version control and collaboration platform. This enabled team members to manage source code efficiently, track changes, review contributions, and resolve conflicts in an organized manner. The use of GitHub significantly enhanced teamwork, ensured code consistency, and supported a well-coordinated development process.

2.2 Earlier Coursework

The development of UniServe was strongly supported by knowledge and skills acquired through earlier coursework in the Computer Engineering program. Several courses played a key role in shaping the technical and organizational aspects of the project.

2.2.1 Object-Oriented Programming

The Object-Oriented Programming course introduced fundamental concepts such as encapsulation, inheritance, abstraction, and modular design using the Java programming language. Although Java was not directly used in this project, the same object-oriented principles were applied when developing the application using **Dart** within the **Flutter** framework. These principles contributed to creating a structured, maintainable, and scalable codebase.

2.2.2 Database Design

Database Design was a critical course that directly influenced the project, as most mobile and web applications rely on efficient data storage and retrieval. Through this course, database modeling, normalization, and query design using **MySQL** were studied. These concepts were applied in UniServe to design and implement a structured MySQL database that supports user management, activity tracking, and document handling.

2.2.3 Software Engineering

The Software Engineering course played a major role in the success of this project. It provided essential knowledge about the software development lifecycle, requirement analysis, system design, and best practices for writing clean and maintainable code. Additionally, the course emphasized the use of version control systems such as Git and collaboration platforms like GitHub, which were essential for managing the project efficiently.

2.2.4 Critical Thinking and Research Skills

The Critical Thinking and Research Skills course enhanced the team's ability to conduct research, analyze information, and present technical content in a structured academic manner. The skills acquired in this course were essential for writing this report, organizing ideas clearly, and documenting the project in a professional and coherent way.

3 Methodology

3.1 Tools, Methods and Programming Languages

3.1.1 Tools

Figure3. 1: Android Studio



Android Studio is an Integrated Development Environment (IDE) used for developing and testing the mobile application. It provides a comprehensive set of tools for building, debugging.

Figure3. 2: Visual Studio Code (VS Code)



Visual Studio Code is a lightweight and flexible code editor used for both front-end and back-end development. It supports multiple extensions that facilitate writing, debugging, and managing Flutter and Node.js code efficiently.

Figure3. 3: GitHub



A web-based platform used for version control and collaboration through Git. It enabled effective teamwork by managing source code, tracking changes, and maintaining a well-organized and coordinated development process among team members.

Figure3. 4: Firebase



A platform for mobile and web app development, providing tools like authentication, real-time database, and cloud functions, we used it in Chatting and Notifications.

Figure3. 5: MySQL



MySQL is a relational database management system used to store and manage application data. It provides structured data storage, efficient querying, and data integrity, which are essential for supporting core system functionalities.

Figure3. 6: Postman



API development and testing tool, simplifying the process of creating and managing APIs for efficient communication between different components, and we used it to test Node JS code.

Figure3. 7: Google Drive



Google Drive was used as a cloud-based storage and collaboration tool for managing project documentation, sharing files, and maintaining backups throughout the development process.

Figure3. 8: Ollama AI



Ollama AI was utilized to integrate artificial intelligence capabilities within the system. It enabled local AI model execution and experimentation, supporting intelligent features and enhancing system functionality.

3.1.2 Programming Language

Mobile applications and web pages were developed using a combination of programming languages:

Back end:

Figure3. 9: Node JS



For the back end, we chose Node JS as a programming language. which is a Java Script language that assists server-side development.

Front end:

Figure3. 10: Flutter



We employed Dart language, especially the Flutter framework. Dart language is a language that focuses on designing user interfaces, while Flutter offers a range of widgets and tools that make the pages more interactive and beautiful.

3.1.3 Database

In this project, a MySQL database was used as the primary solution for data storage and management. The choice of MySQL was motivated by its reliability, scalability, ease of use, and strong support for relational data structures. Since the UniServe application relies heavily on structured relationships between users, activities, and course requirements, a relational database was considered the most suitable option.

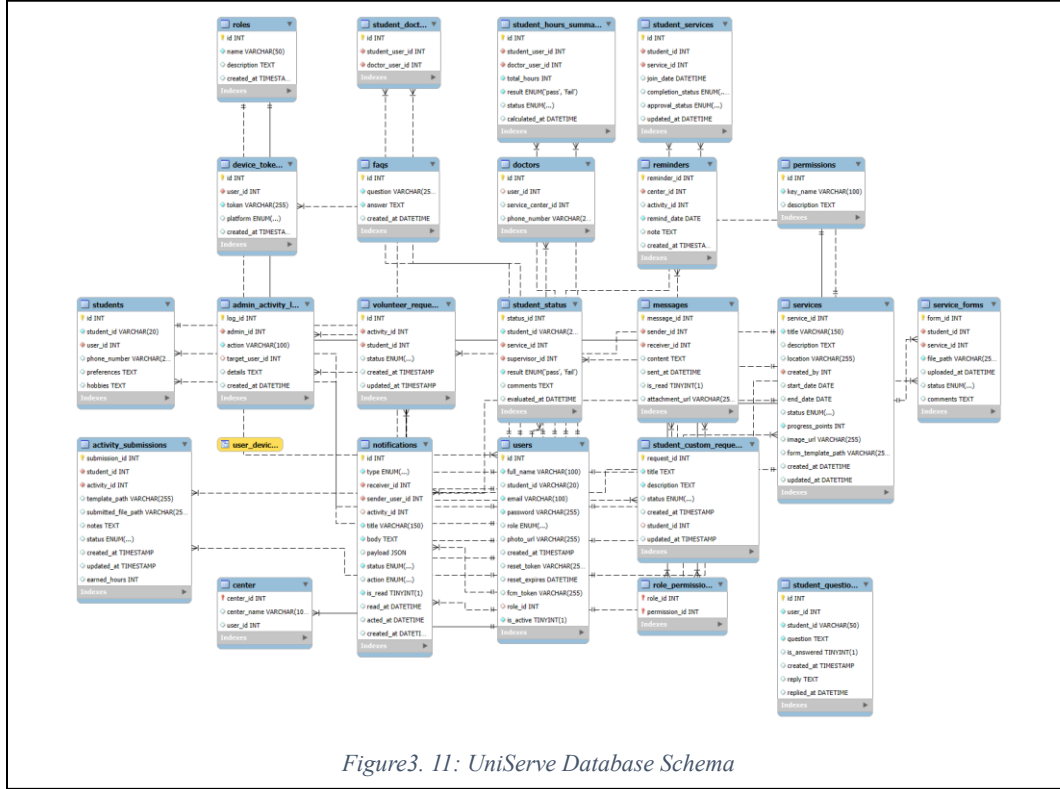


Figure3. 11: UniServe Database Schema

Database Tables Description:

Users Table: Stores general user information, including students, supervisors, and administrators, and manages authentication, roles, and account status.

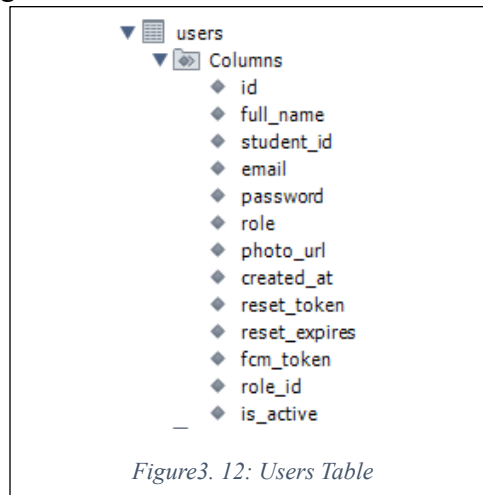
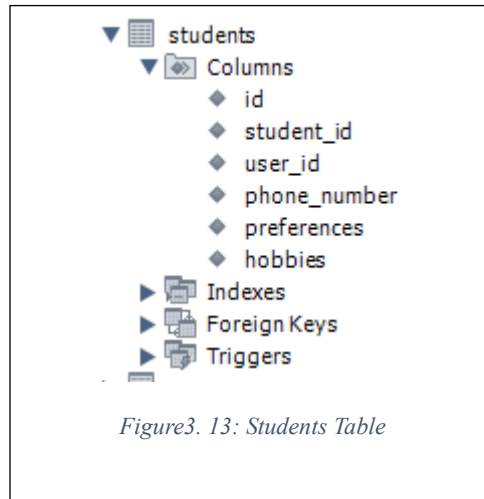
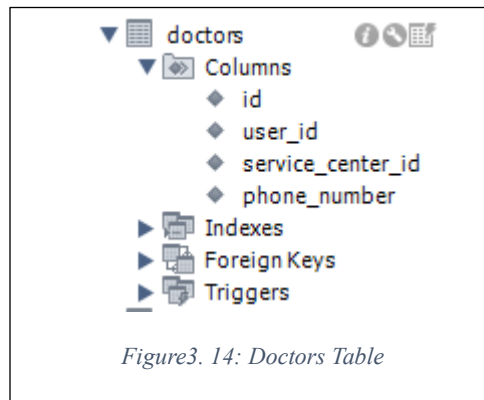


Figure3. 12: Users Table

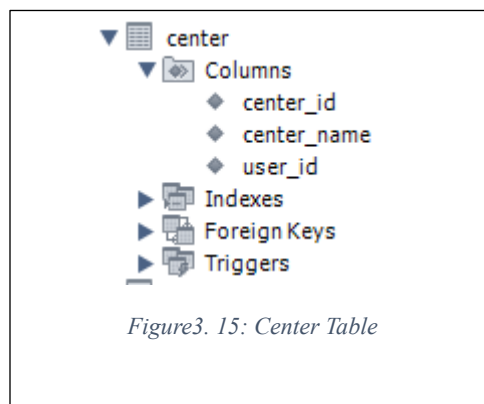
Students Table: Contains additional details related to students, such as preferences, contact information, and links to their system user accounts.



Doctors Table: Stores information related to supervisors, including contact details and their associated service centers.



Service Center Table: Represents community service centers, including center details and the user responsible for managing each center.



Services Table: Stores information about available community service activities, including descriptions, locations, duration, and progress points.

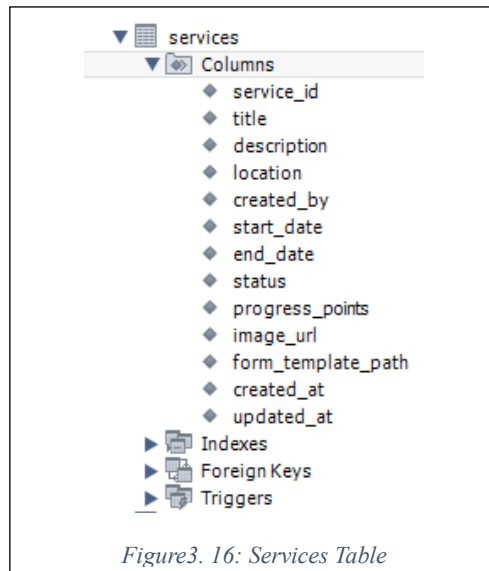


Figure3. 16: Services Table

Student Services Table: Manages the relationship between students and services, tracking enrollment, participation status, and completion results.

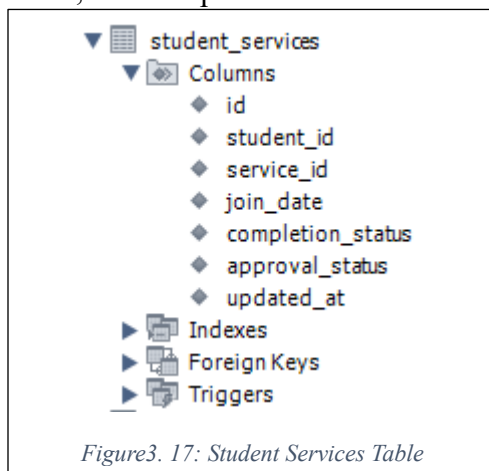


Figure3. 17: Student Services Table

Student Status Table: Tracks the evaluation status of students for each service, including approval status, supervisor feedback, and final results.

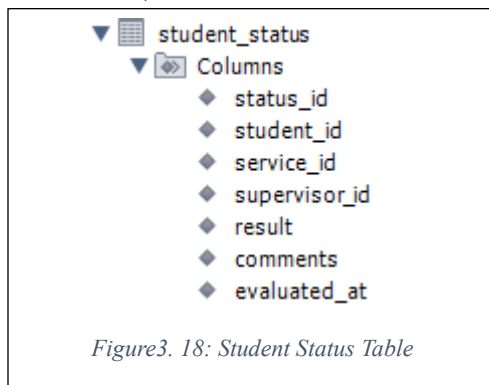


Figure3. 18: Student Status Table

Activity Submissions Table: Stores student submissions related to community service activities, including uploaded files, notes, earned hours, and submission status.

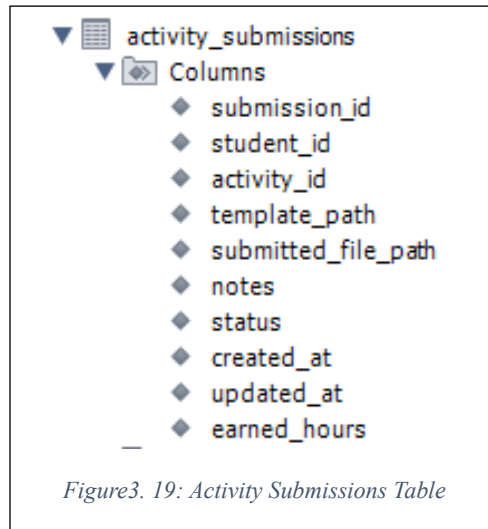


Figure3. 19: Activity Submissions Table

Student Hours Summary Table: Maintains a summary of total service hours completed by each student, along with the final evaluation result.

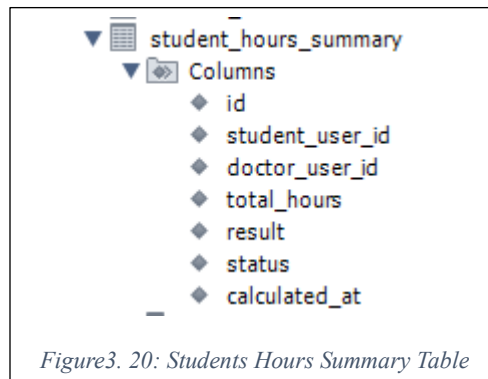


Figure3. 20: Students Hours Summary Table

Service Forms Table: Stores forms submitted by students for specific services, including uploaded documents, comments, and approval status

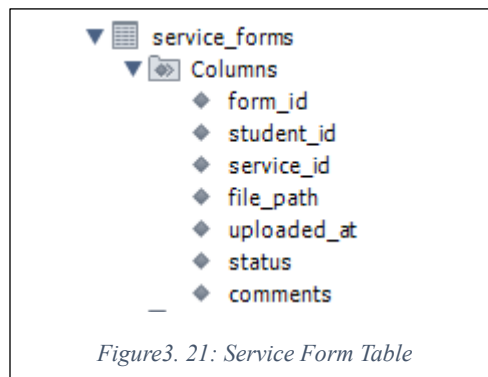
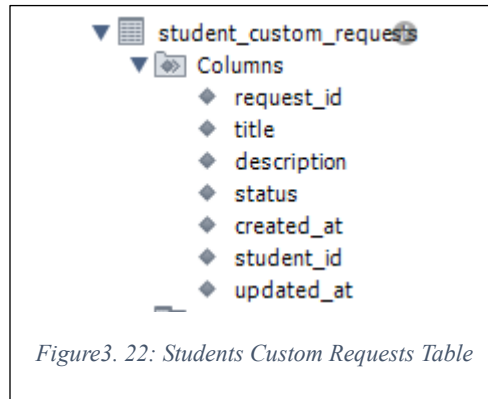
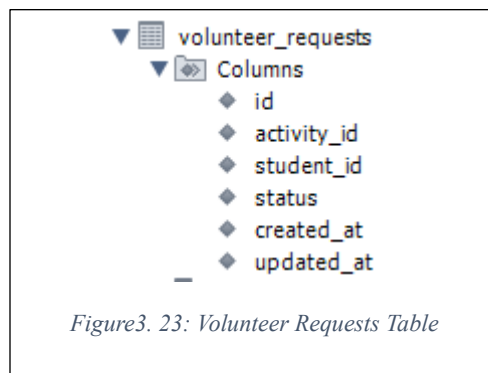


Figure3. 21: Service Form Table

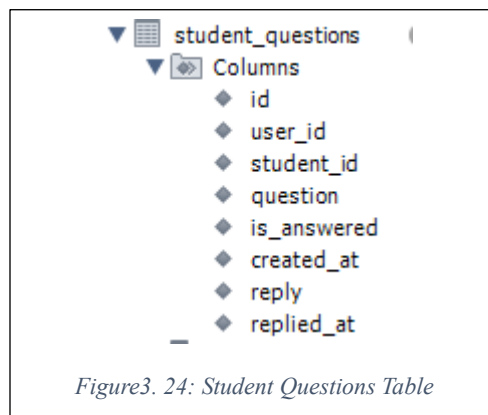
Student Custom Requests Table: Handles custom service requests submitted by students, including request details, status, and timestamps.



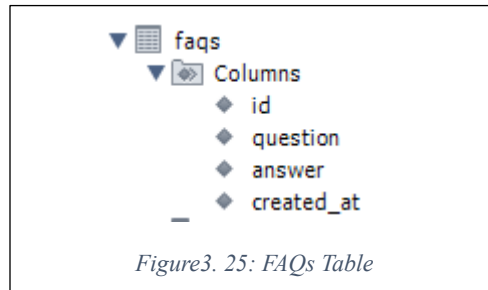
Volunteer Requests Table: Stores volunteer participation requests for activities, tracking student involvement and request approval status



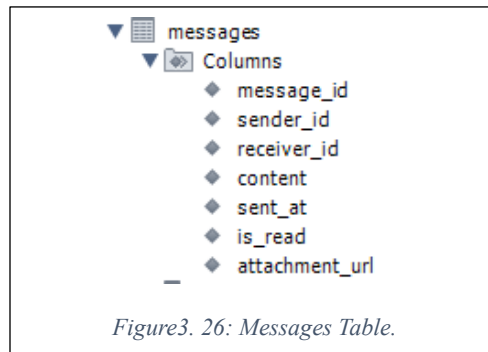
Student Questions Table: Manages questions submitted by students, including responses, reply status, and timestamps.



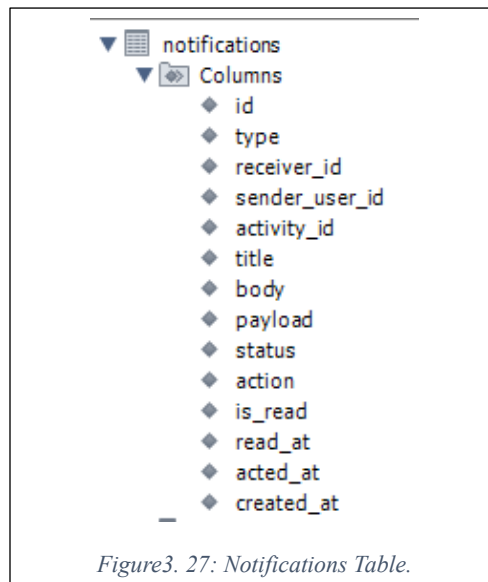
FAQs Table: Stores frequently asked questions and their corresponding answers to support student guidance.



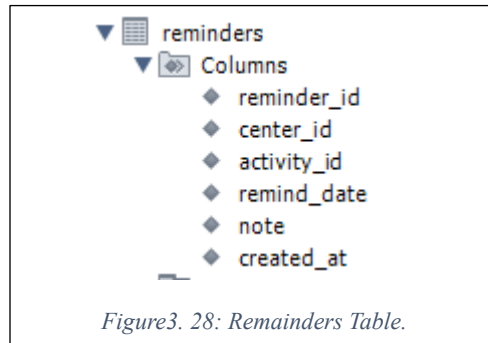
Messages Table: Handles internal messaging between users, including message content, attachments, read status, and timestamps.



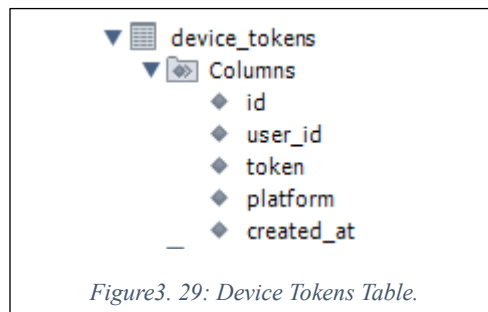
Notification Table: Manages in-app notifications, tracking notification type, status, read state, and related activity data.



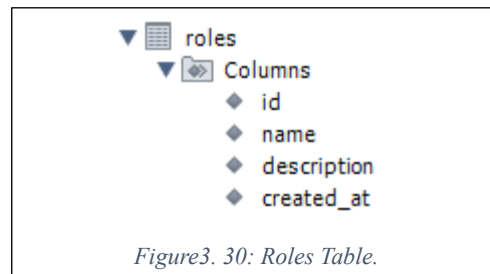
Reminders Table: Stores reminder information related to activities or deadlines, including reminder dates and notes.



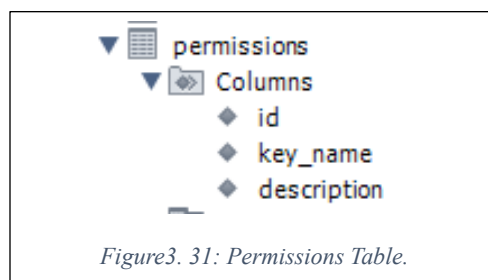
Device Tokens Table: Manages notification tokens for user devices, enabling push notification delivery across platforms.



Roles Table: Defines system roles and their descriptions, supporting role-based access control.



Permissions Table: Stores individual system permissions that define allowed actions within the application.



Role Permissions Table: Manages the relationship between roles and permissions, enabling flexible access control.

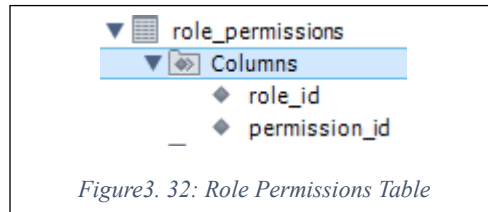


Figure3. 32: Role Permissions Table

Admin Activity Logs: Tracks administrative actions performed within the system for monitoring, auditing, and accountability purposes.

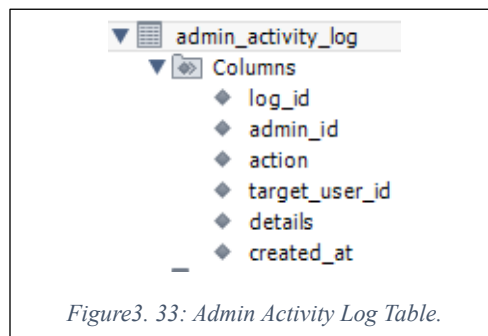


Figure3. 33: Admin Activity Log Table.

Student Doctor Table: This table defines the association between students and their assigned supervisors (doctors) by linking student user IDs with doctor user IDs. It supports supervisor assignment and tracking within the community service workflow.

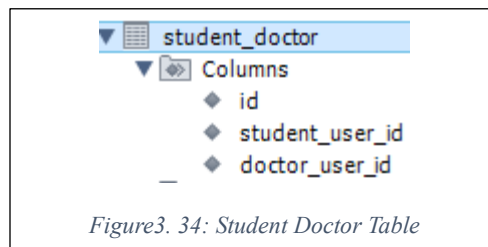


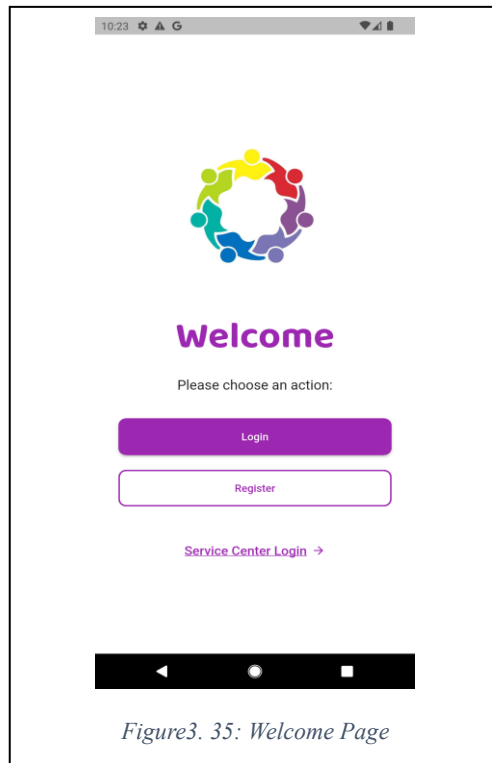
Figure3. 34: Student Doctor Table

3.2 System Features Implementation

The **UniServe** application provides an integrated and structured system designed to facilitate the Community Service course at An-Najah National University. The system includes multiple features that aim to simplify course management, enhance user experience, and ensure efficient communication between students, supervisors, and administrators. The main system features are described in the following sections.

3.2.1 Mobile application

Welcome Page:



Start Page:

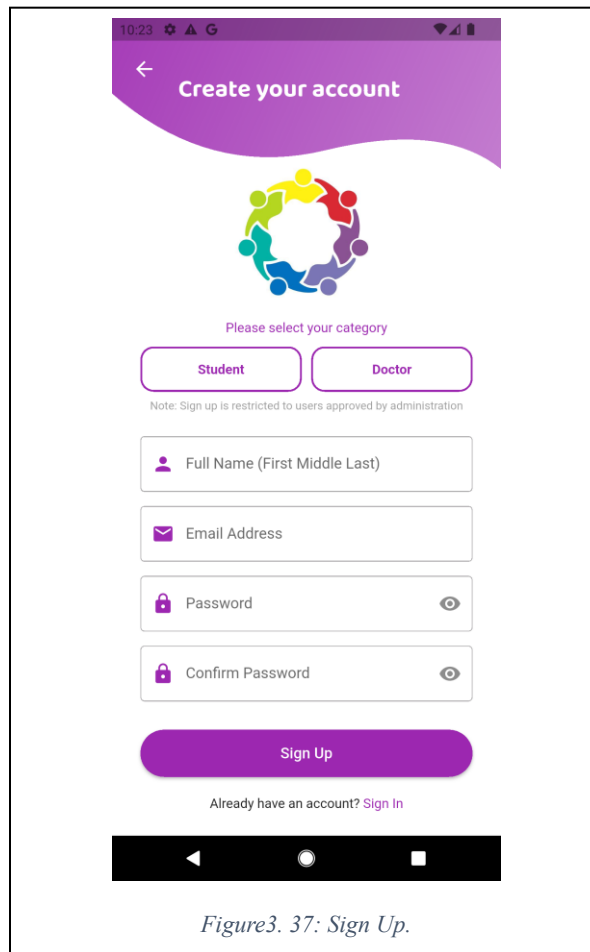


User Registration and Authentication Process:

The UniServe system adopts a controlled registration mechanism to ensure security, data integrity, and the authenticity of all users within the platform. The sign-up functionality is exclusively available for **Community Service Centers**, allowing authorized centers to create accounts and access the system directly.

In contrast, **students and supervisors (doctors)** are not allowed to register independently. Their accounts are created and activated by the system administrator after verification. This approach ensures that only students who are officially enrolled in the Community Service course and only supervisors who are legitimately responsible for delivering the course are granted access to the system.

By restricting self-registration and relying on administrative approval, the system enhances security, prevents unauthorized access, and guarantees that all participating users are valid members of the community service process. This design choice plays a crucial role in maintaining the credibility and reliability of the UniServe application.



10:23

← Create your account



Please select your category

Student Doctor

Note: Sign up is restricted to users approved by administration

Full Name (First Middle Last)

Email Address

Password

Confirm Password

Sign Up

Already have an account? [Sign In](#)

Figure3. 37: Sign Up.

User Sign-In Process:

The UniServe system allows **Community Service Centers** to sign in after successfully completing the sign-up process. Once registered, centers can securely access the system using their credentials.

For **students and supervisors (doctors)**, sign-in access is restricted to verified users only. Students are permitted to sign in only if they are officially enrolled in the Community Service course, while supervisors can sign in only if their accounts have been created and activated by the system administrator. This controlled sign-in process ensures secure access and guarantees that all users are authorized participants within the system.

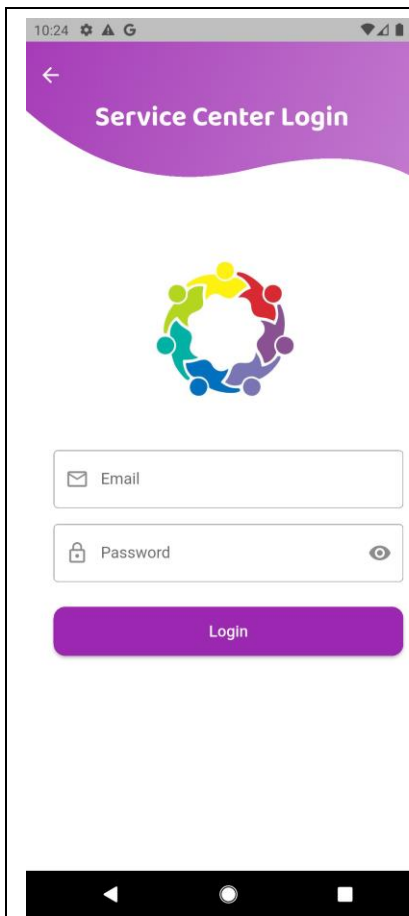


Figure3. 38: Center Sign In.

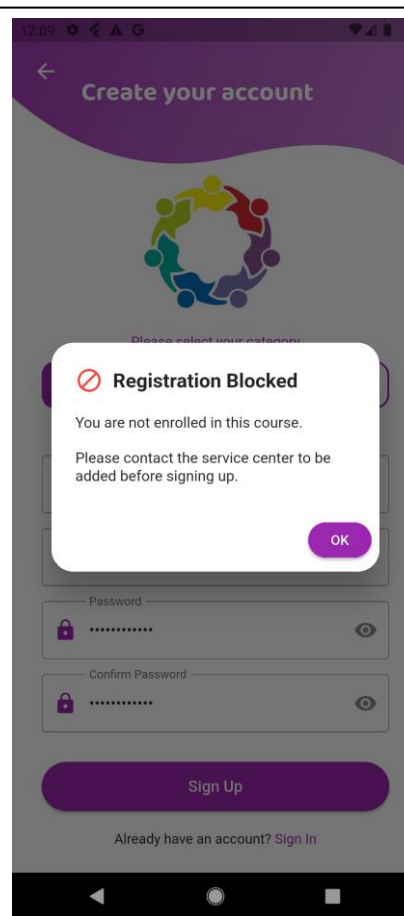


Figure3. 39: Blocked Sign in

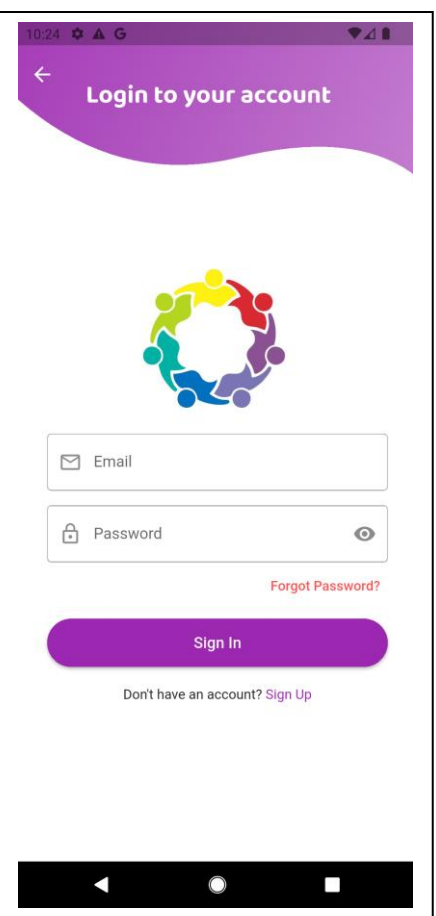
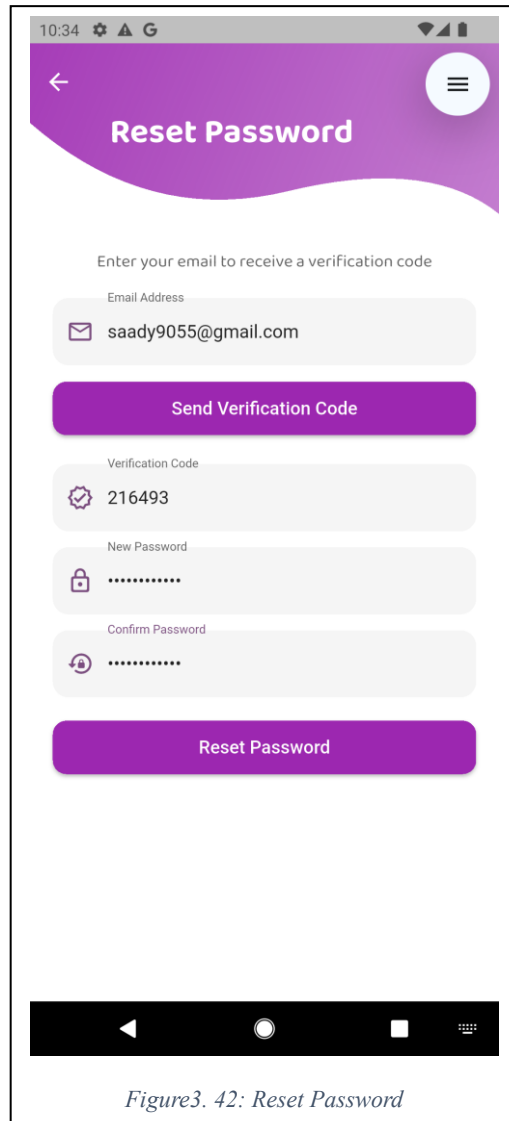
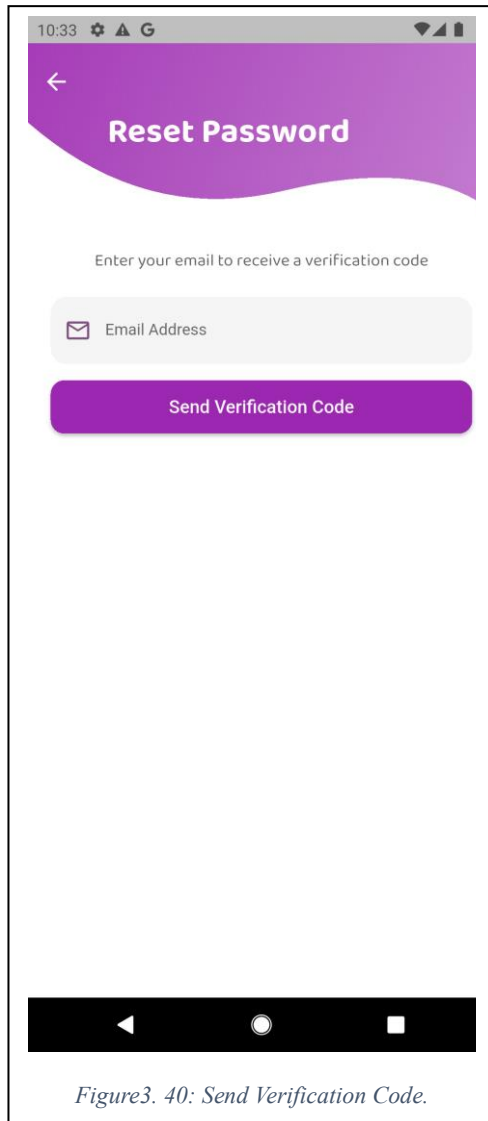


Figure3. 40: Students&Doctors Sign In

Forgot Password Feature:

The UniServe system provides a secure password recovery mechanism for users who forget their credentials. Upon requesting a password reset, a verification code is sent to the user, allowing them to securely reset their password. This feature enhances system security while ensuring a smooth and reliable account recovery process.



Profile Page:

The Profile Page serves as a central access point that allows users to view and manage all functionalities related to their accounts. Through this page, users can access the features they need, review their information, and perform required actions efficiently based on their assigned roles. This page enhances usability by providing organized and convenient navigation to all system services.

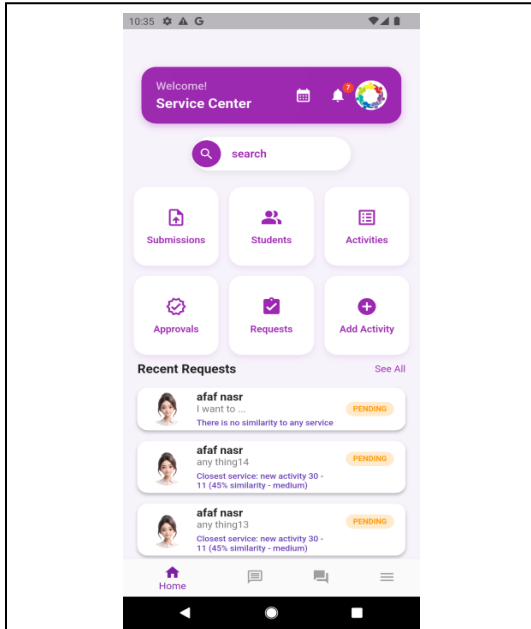


Figure3. 41: Center Home Page.

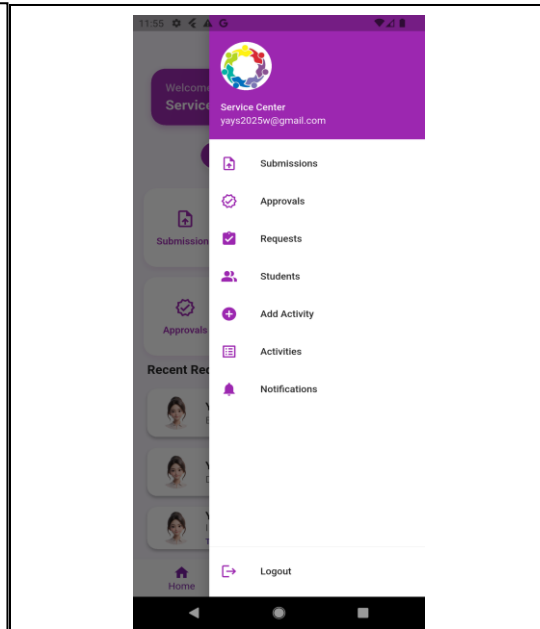


Figure3. 43: Center Home Side Par.

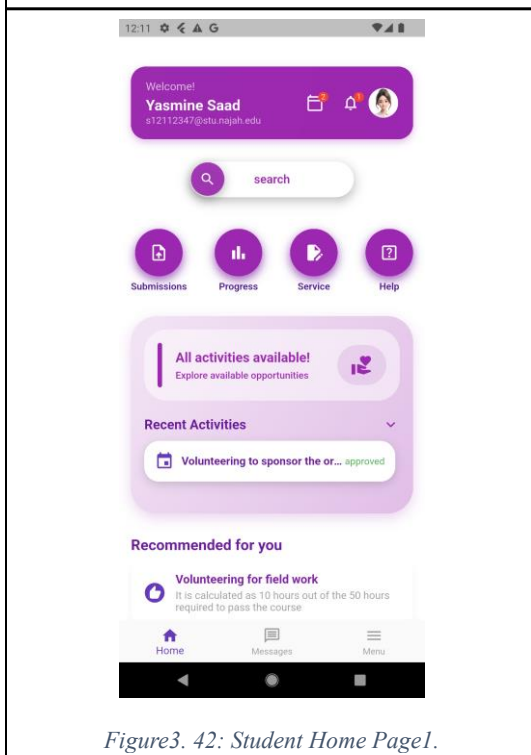


Figure3. 42: Student Home Page1.

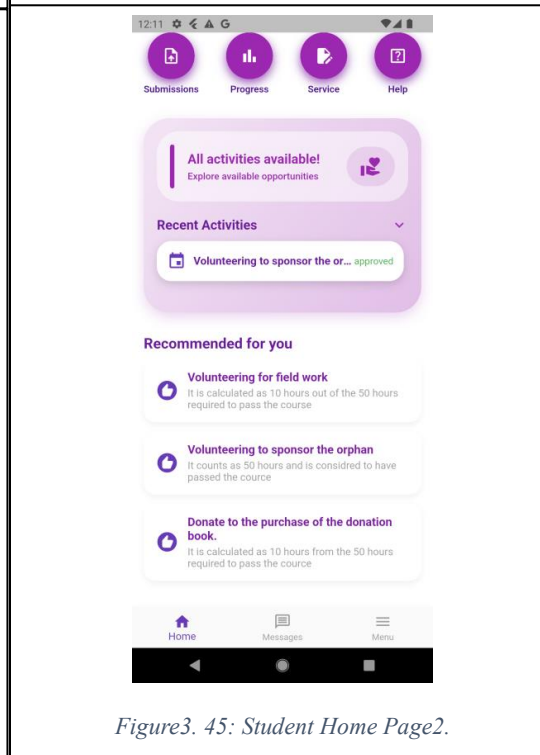


Figure3. 45: Student Home Page2.

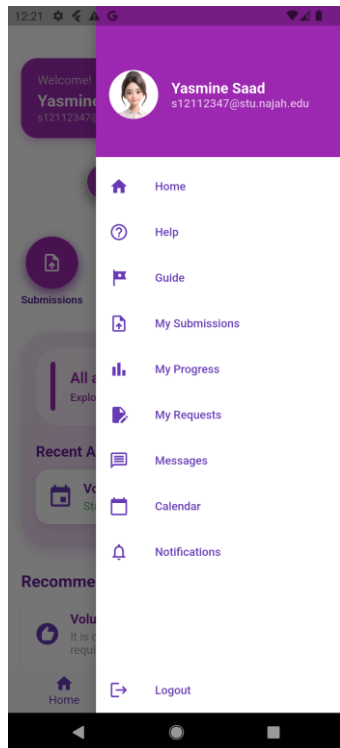


Figure3. 43: Student Home Side Par.

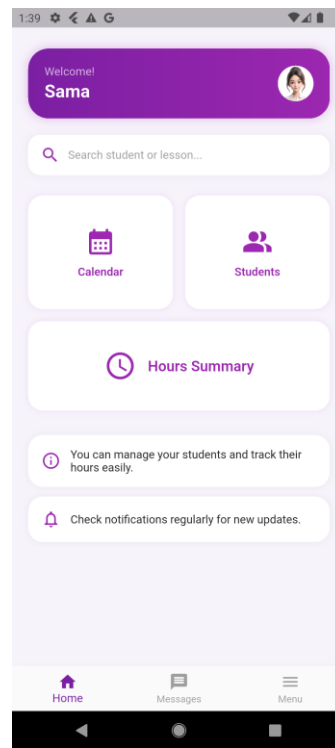


Figure3. 47: Doctor Home Page

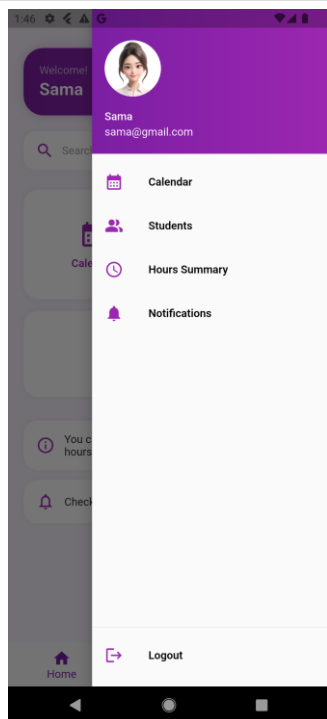


Figure3. 45: Doctor Home Side Par.

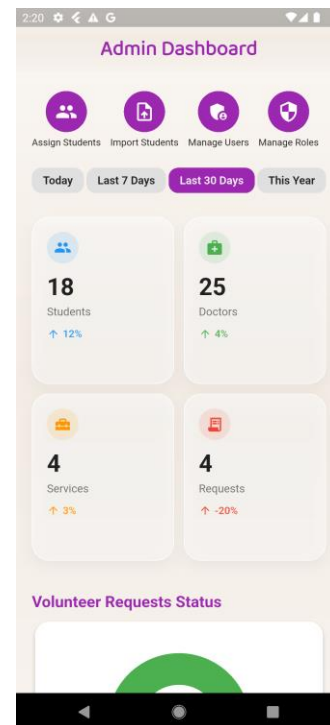


Figure3. 44:Admin Dashboard1.

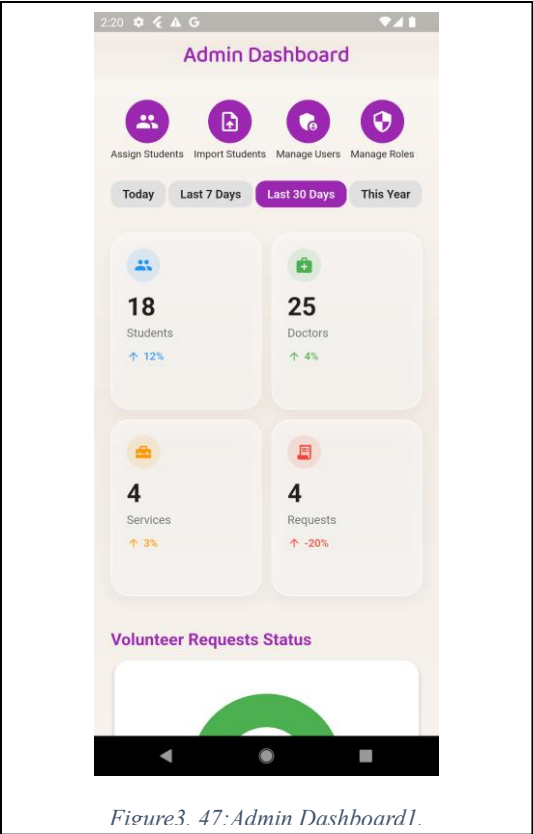


Figure3. 47: Admin Dashboard1.

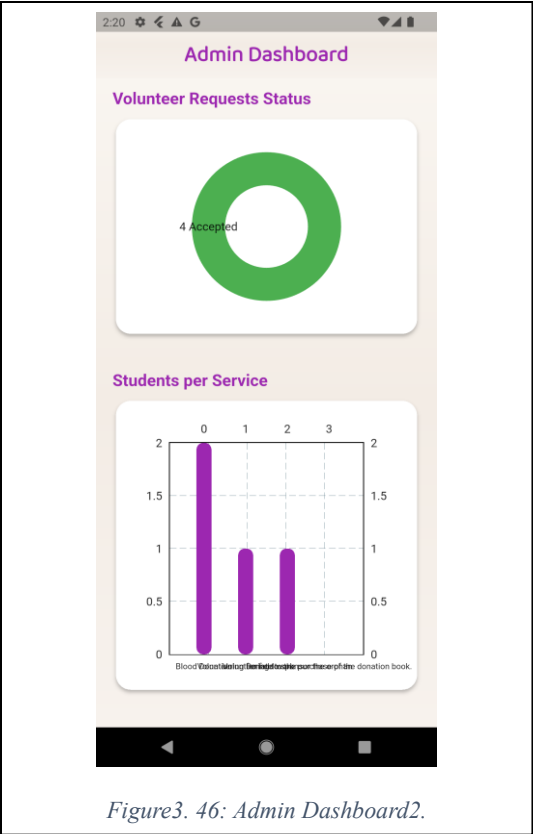


Figure3. 46: Admin Dashboard2.

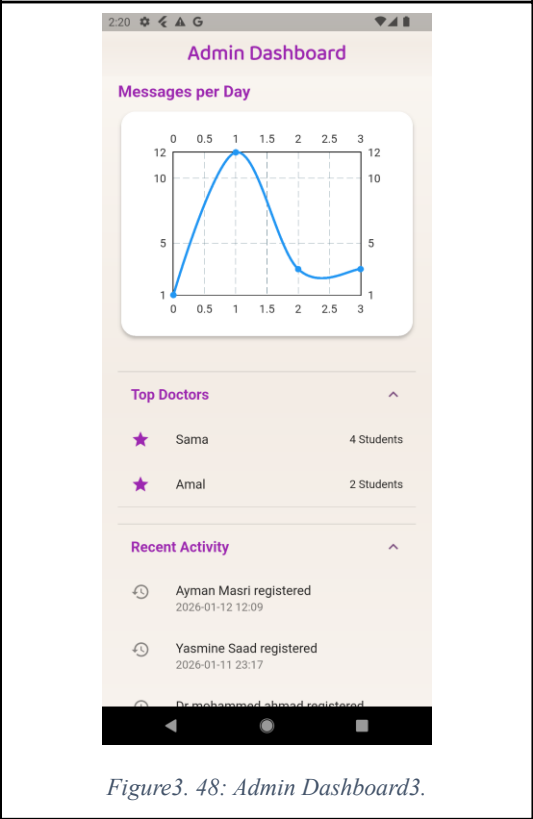


Figure3. 48: Admin Dashboard3.

Profile Edit Page:

The Profile Page provides users with centralized access to all features related to their accounts, allowing them to view information and perform required actions efficiently based on their roles.

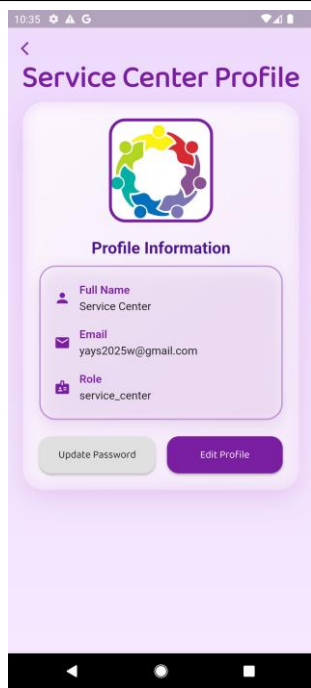


Figure3. 50: Center Profile View Page.

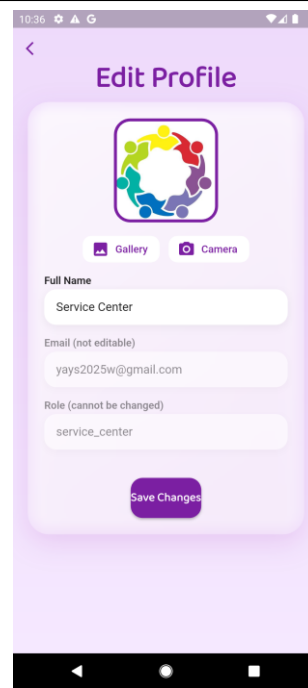


Figure3. 49: Center Edit Profile Page

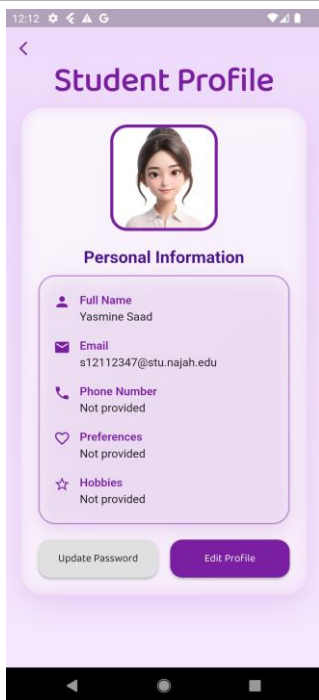


Figure3. 51: Student Profile View.



Figure3. 52: Student Edit Profile

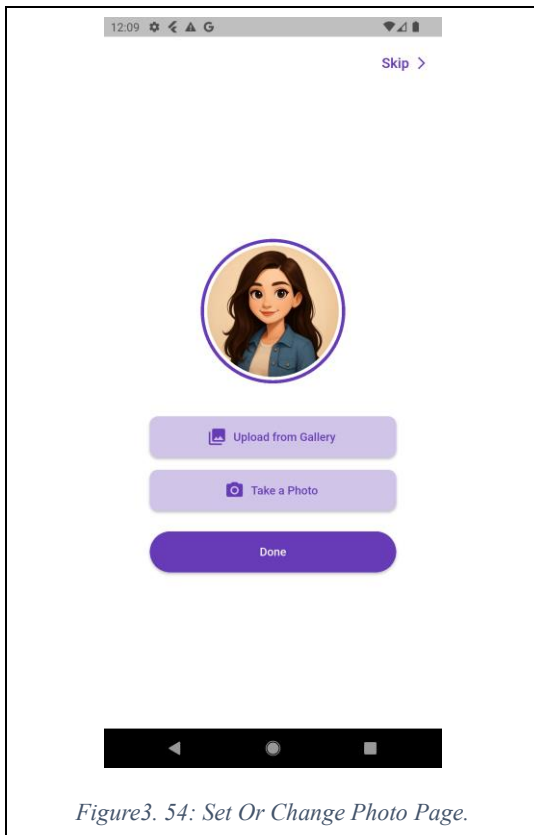


Figure3. 54: Set Or Change Photo Page.

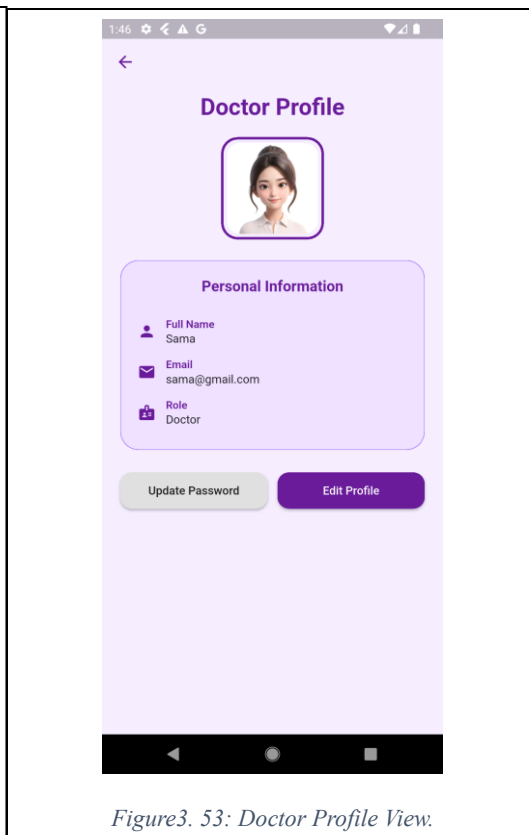


Figure3. 53: Doctor Profile View.

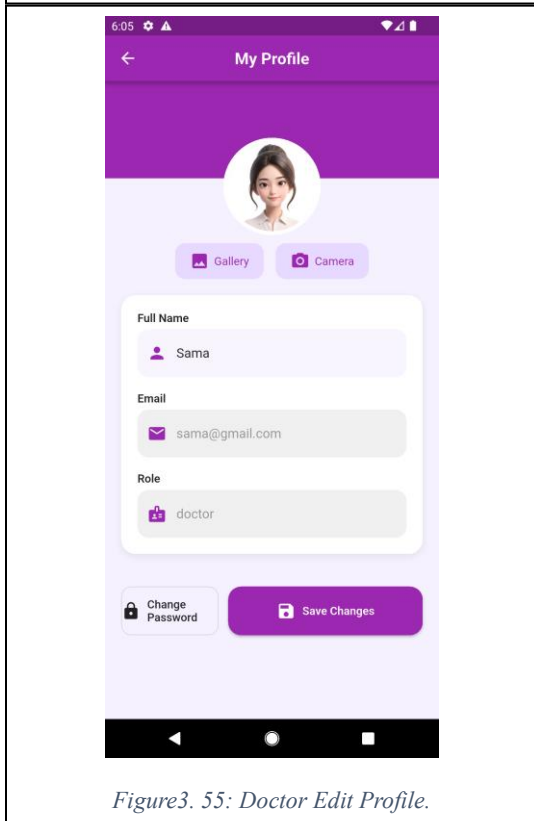


Figure3. 55: Doctor Edit Profile.

Messaging Feature:

The UniServe system provides a role-based messaging feature that enables each user role to access and exchange messages relevant to its responsibilities. This feature facilitates secure and organized communication between students, supervisors, centers, and administrators, ensuring efficient coordination within the community service workflow.

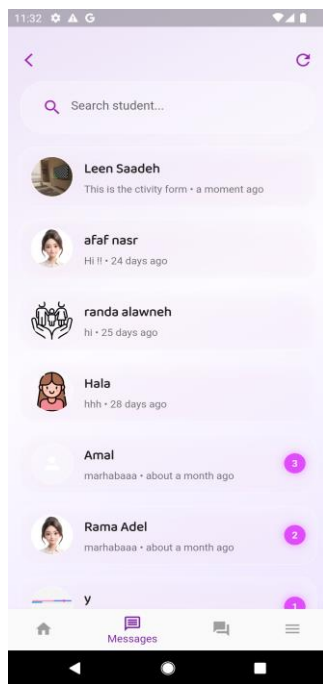


Figure3. 57: Center All Messages.

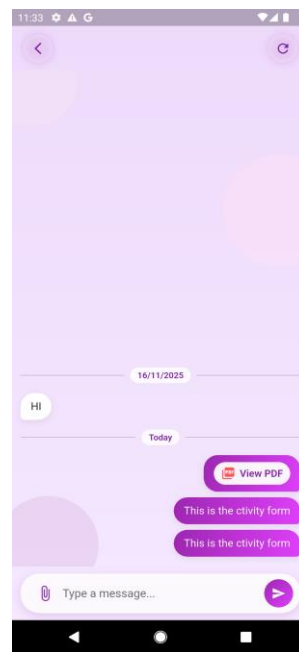


Figure3. 56: A conversation between a student and the center.



Figure3. 58: Messages between the student and the service center ("student's side").

Notification System Feature:

The UniServe application includes a role-based notification system available on both mobile and web platforms. Each user receives notifications relevant to their assigned role, ensuring timely updates about activities, messages, deadlines, and system actions. This feature enhances communication, improves user engagement, and ensures that important information is delivered efficiently across all platforms.

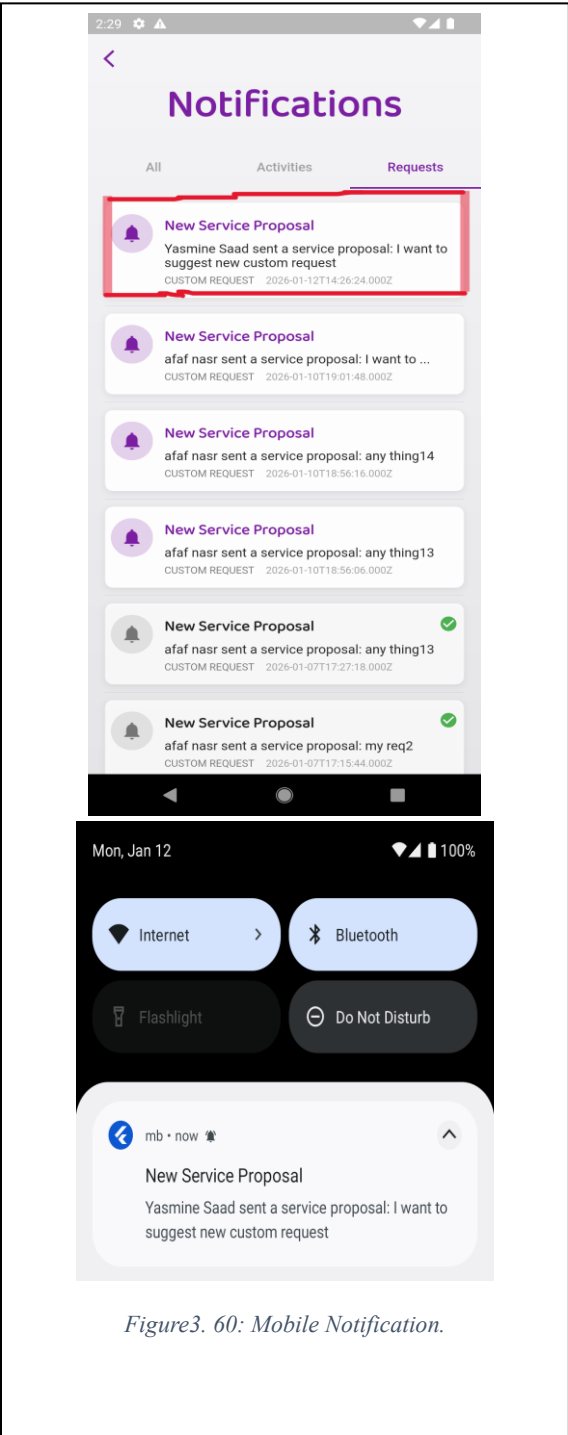


Figure3. 60: Mobile Notification.

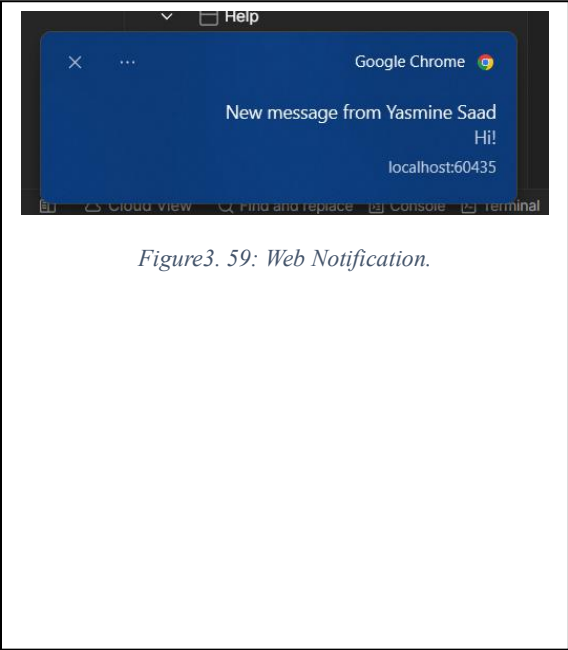


Figure3. 59: Web Notification.

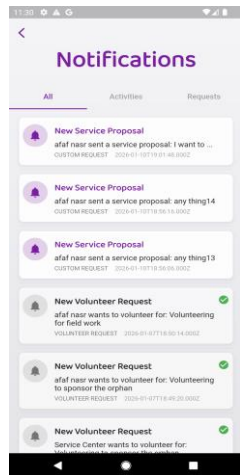


Figure3. 62: ALL Center Notifications.

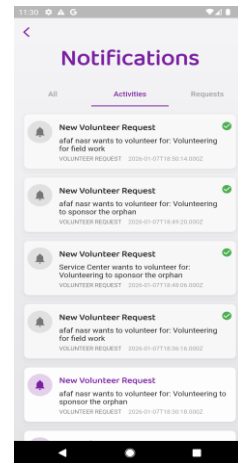


Figure3. 61: Volunteer Requests Notifications.

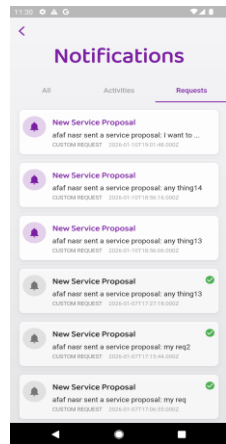


Figure3. 64: Custom Requests Notifications.

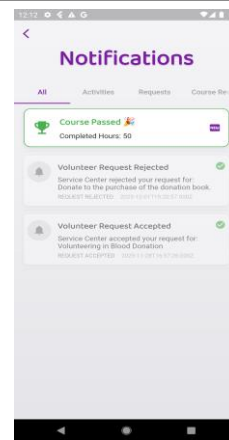


Figure3. 63: All Student Notifications.

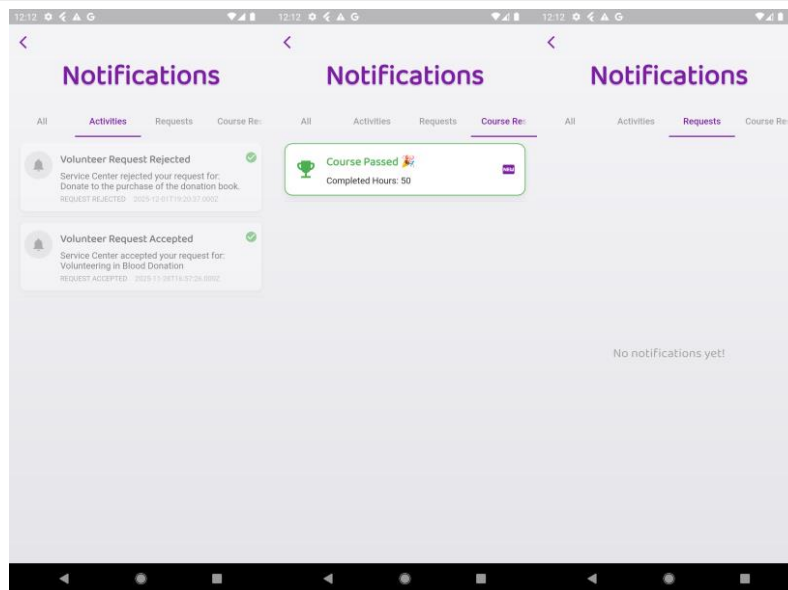


Figure3. 65: Student Notifications Categories

Student Guide Feature:

The UniServe system provides a Student Guide feature that assists students in understanding the Community Service course requirements and system usage. It offers clear instructions, guidance, and relevant information to help students complete their tasks correctly and efficiently.

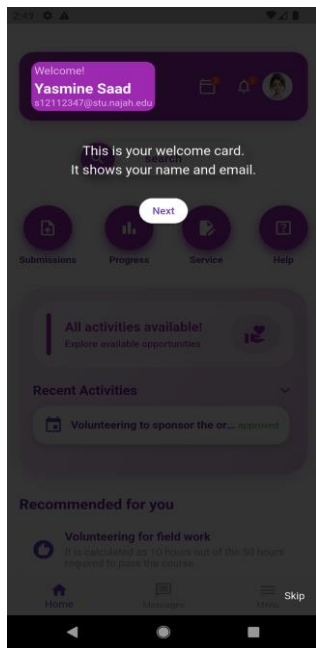


Figure3. 69: Student Guide – Welcome Card

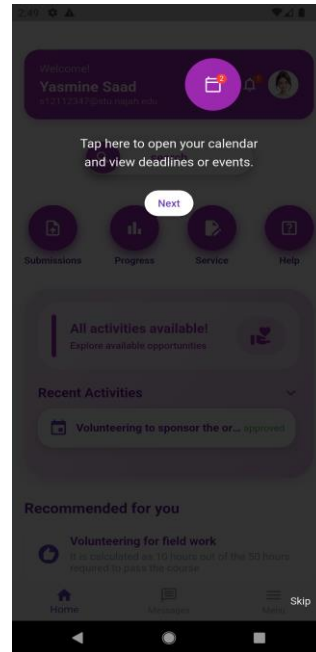


Figure3. 68: Navigation and Calendar Overview

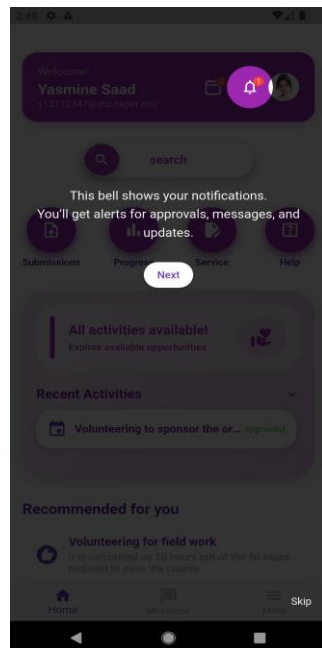


Figure3. 66: Notifications Overview

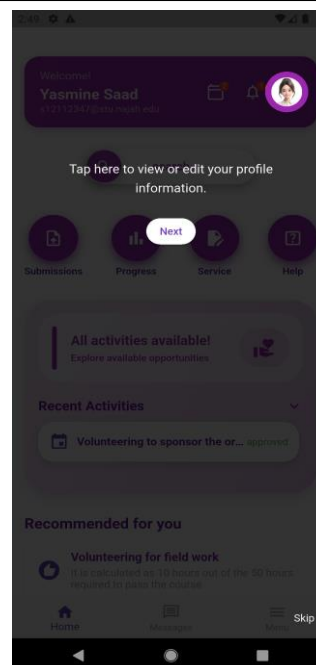


Figure3. 67: Profile Navigation.

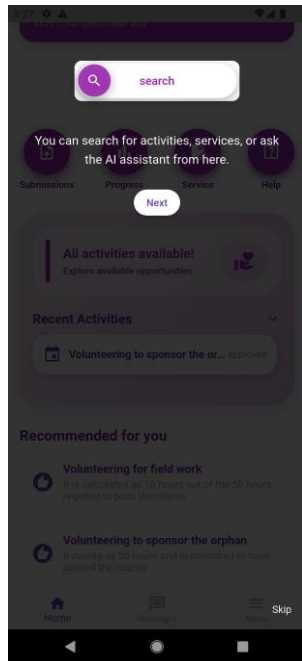


Figure3. 70: Search Feature

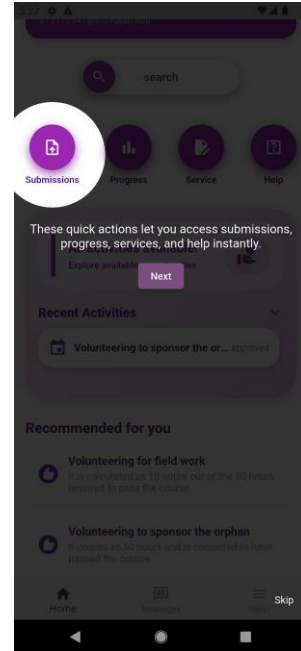


Figure3. 74: Submit Forms.

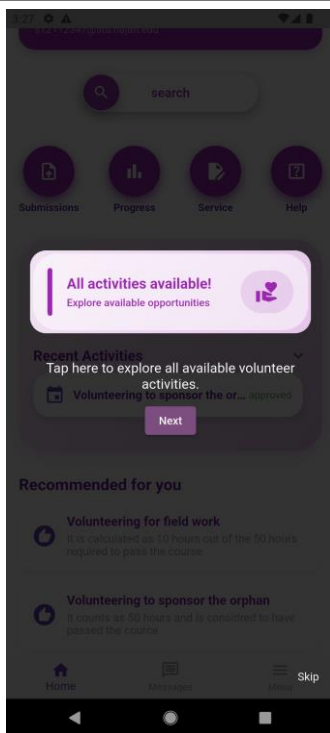


Figure3. 71: Available Activities Overview

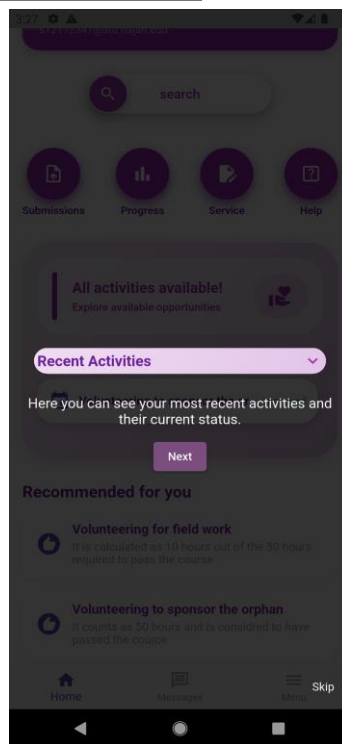


Figure3. 73: Recent Activities

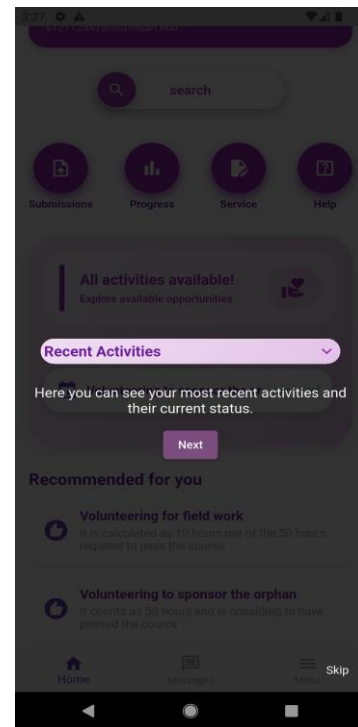


Figure3. 72: Activity Status Tracking.

Activities Management Feature:

The UniServe system provides a comprehensive activities management feature that enables **Community Service Centers** to create and publish available volunteering activities. Through this feature, centers can add activity details such as title, description, location, media, and schedule, making the activities visible to students in an organized and clear manner.

Students can browse all available activities through their application interface, view activity details, and submit volunteering requests for activities that match their interests. This process facilitates student participation and simplifies the selection of suitable community service opportunities.

Additionally, Community Service Centers are granted full control over their published activities, allowing them to **update or delete activities** when necessary. This ensures that all listed activities remain accurate, up to date, and aligned with current community service needs.

In the Center Side:

Add New Activity:

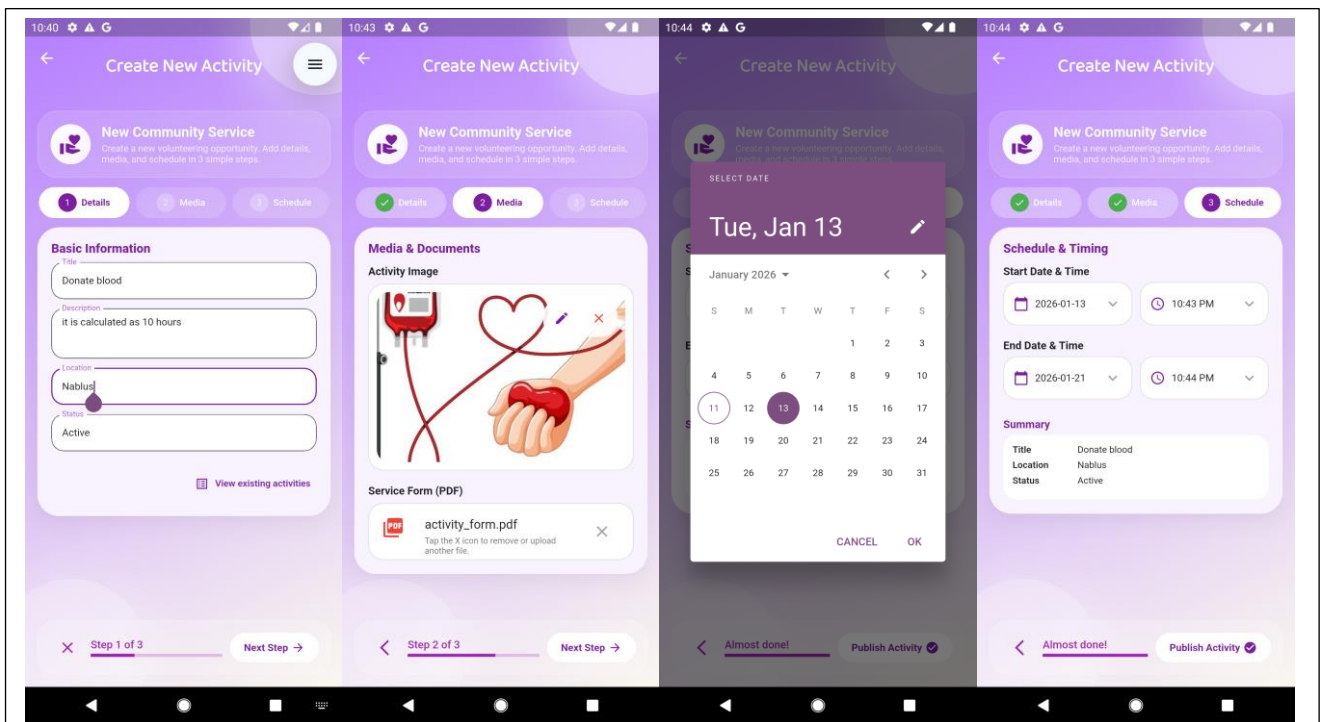


Figure3. 75: Add New Activity.

Viewing Activities:

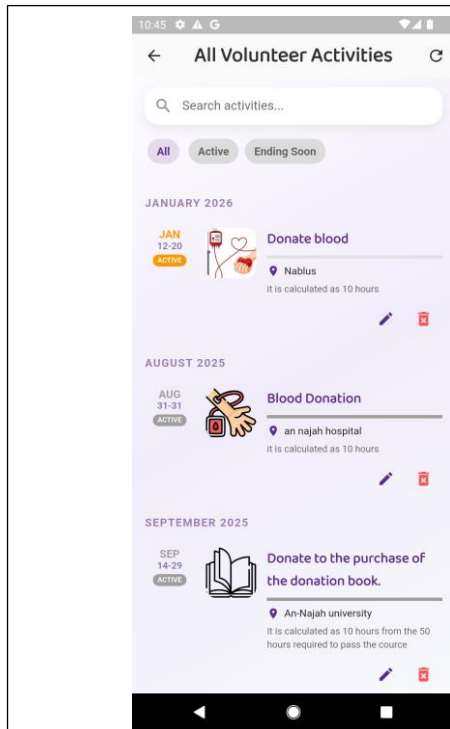


Figure3. 77: Center View Activities Page.

Update Specific Activity:

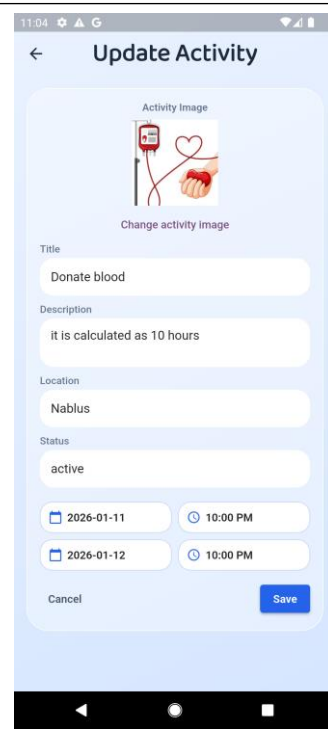


Figure3. 76: Center Update Activity Page.

Viewing a specific activity:

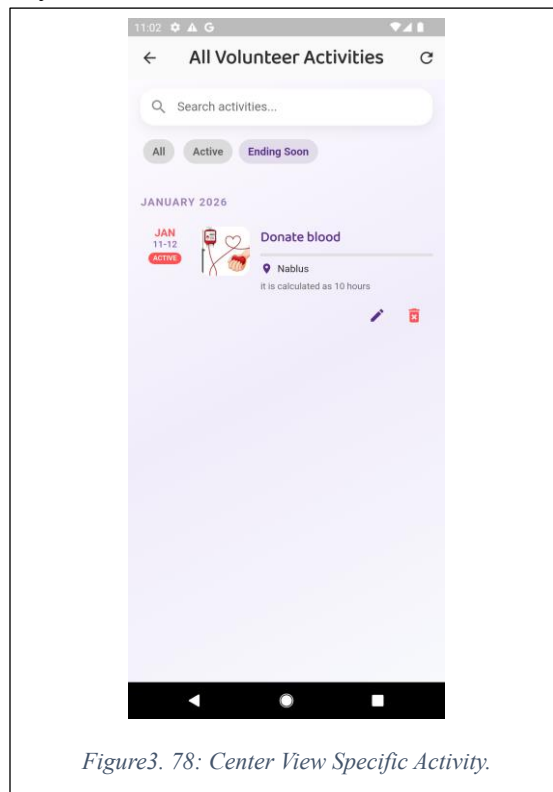


Figure3. 78: Center View Specific Activity.

In the Student Side:

The student views the activities posted by the service center and requests to volunteer in specific activities:

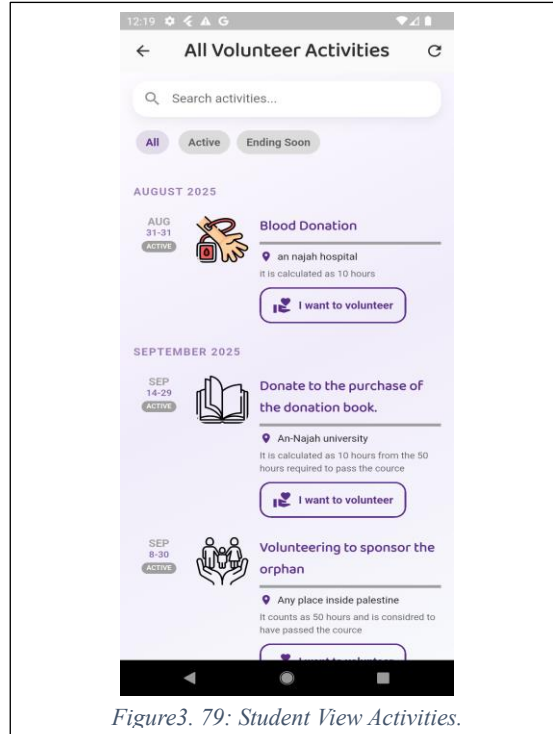


Figure3. 79: Student View Activities.

When a student applies to volunteer in a specific activity, the request is automatically sent to the corresponding Service Center. The request appears in the center's requests list, allowing administrators to review and manage it. In addition, real-time notifications are generated to inform the Service Center about new volunteer requests, ensuring timely review and response.

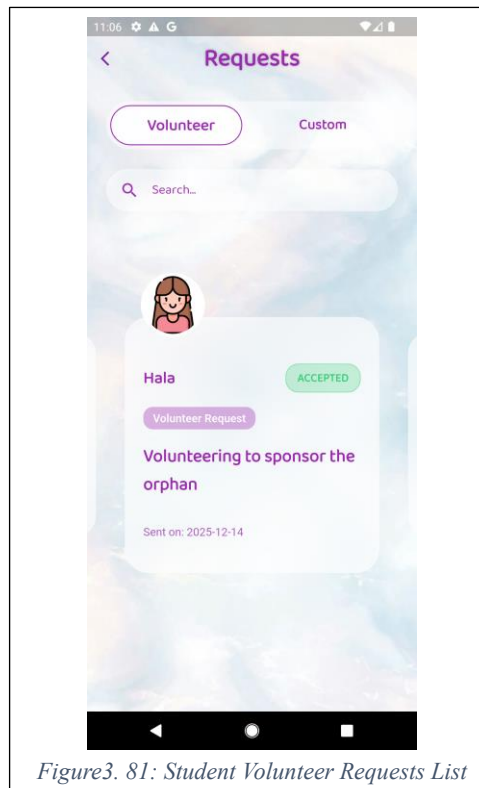


Figure3. 81: Student Volunteer Requests List

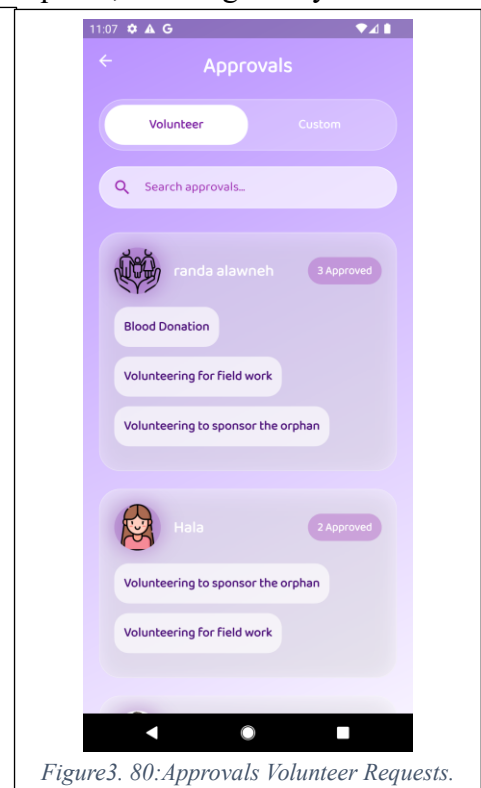


Figure3. 80:Approvals Volunteer Requests.

Custom Requests Feature:

UniServe provides a feature to allows students to submit custom community service activities they wish to volunteer in. These requests are reviewed by the Service Center, which can either approve or reject them. Throughout this process, communication between the student and the Service Center is facilitated through in-app notifications.

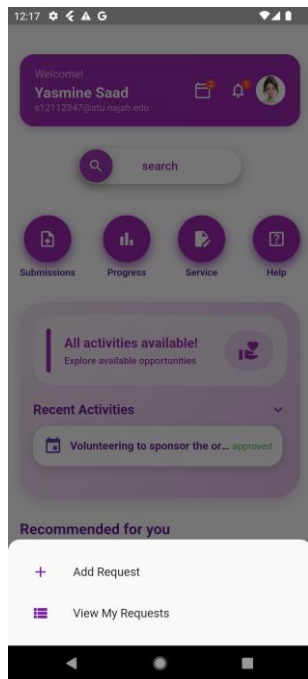


Figure3. 83: Student Custom Service Requests Interface

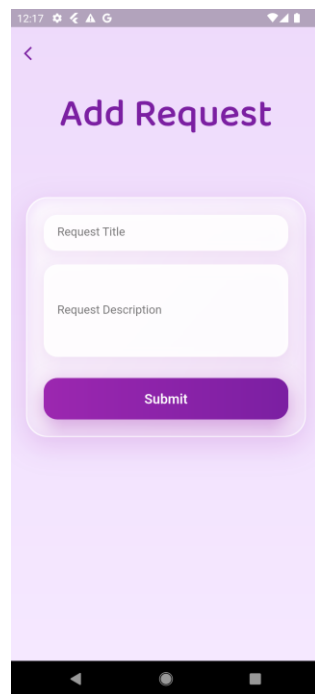


Figure3. 82: Add Custom Requests.

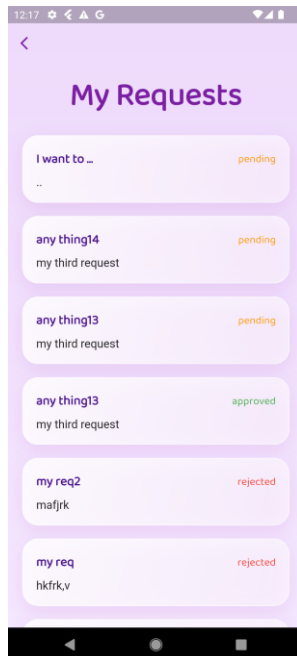
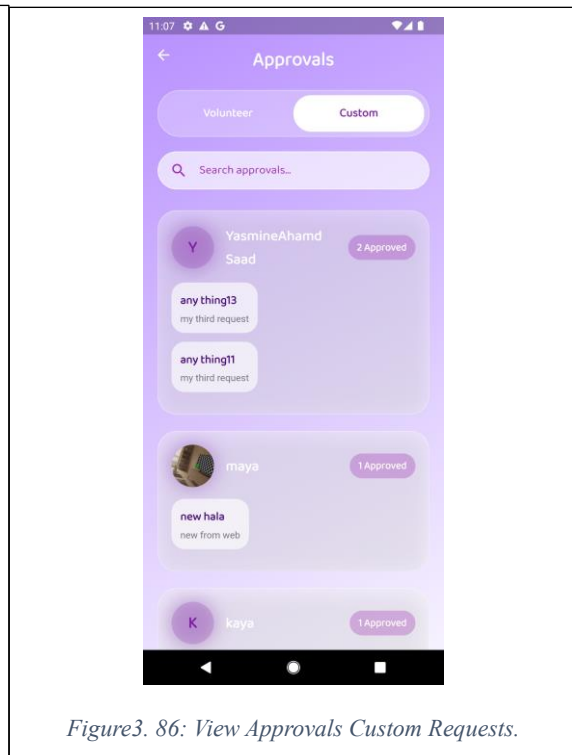
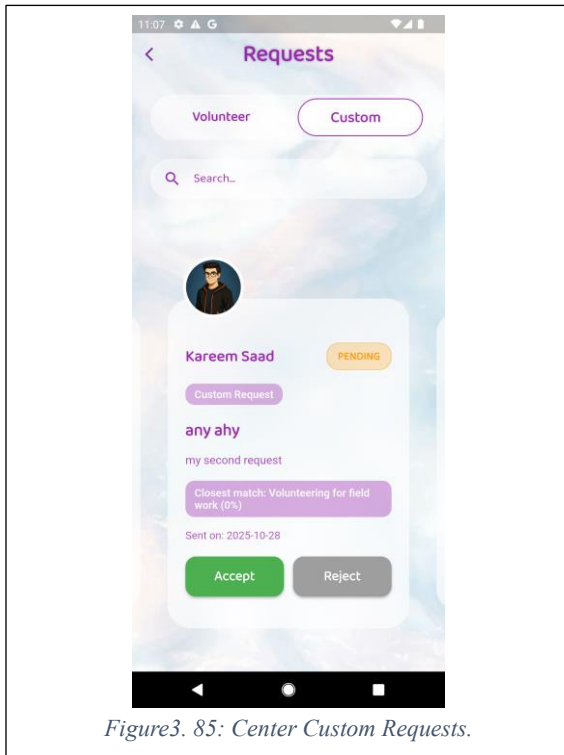


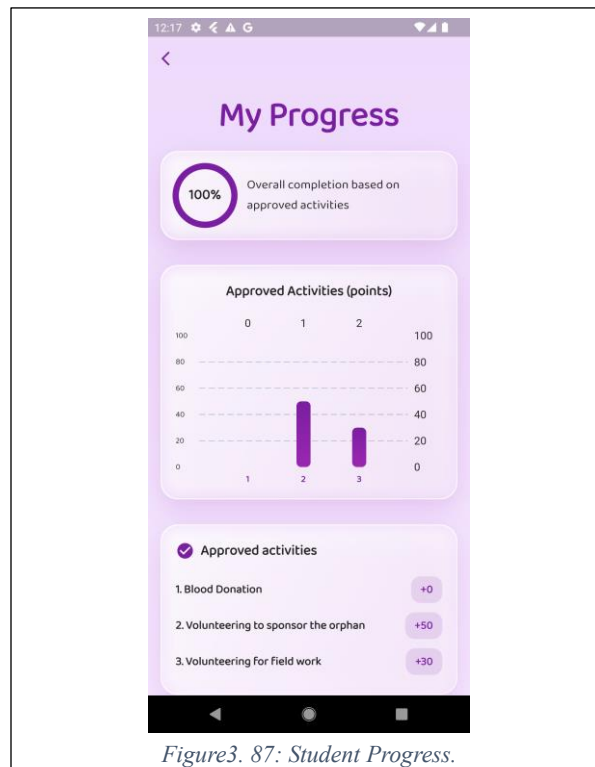
Figure3. 84: Student View Its Requests

In the center Side:



Student Progress:

The My Progress page allows students to track their completed volunteer hours and monitor the remaining hours needed after their activities are approved.



AI Features:

AI Features in the System

- Student Side - Smart Recommendations

On the student side, the system uses Artificial Intelligence to provide personalized activity recommendations.

The AI analyzes the student's interests, preferences, and previous volunteering history, then suggests suitable volunteering activities that match their profile.

This helps students easily discover activities that align with their skills and interests, making the volunteering experience more engaging and effective.

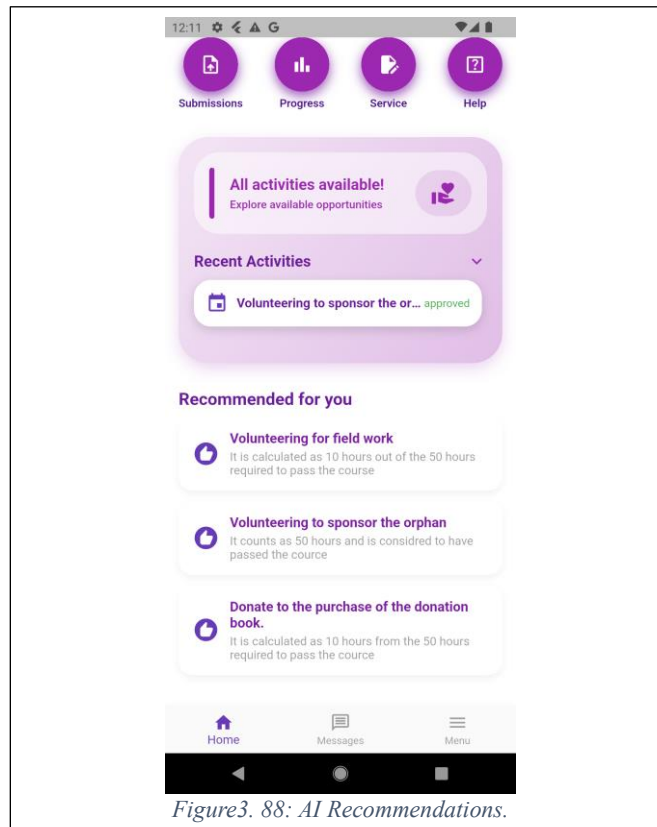


Figure3. 88: AI Recommendations.

- Service Center Side – Intelligent Request Evaluation:

On the service center side, AI is integrated into the **new volunteering requests** submitted by students.

When a student submits a new request, the AI **compares the request with the existing activities** available at the service center and calculates a **similarity percentage** between them.

This similarity score helps the service center **make better decisions** by providing smart insights to support accepting or rejecting the request more efficiently and accurately.

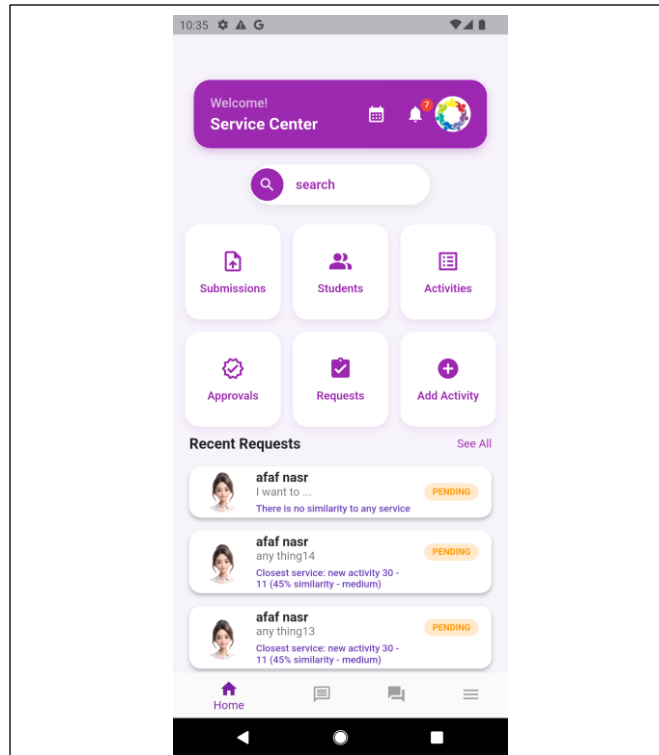


Figure3. 89: AI-Powered Request Analysis Home Page.

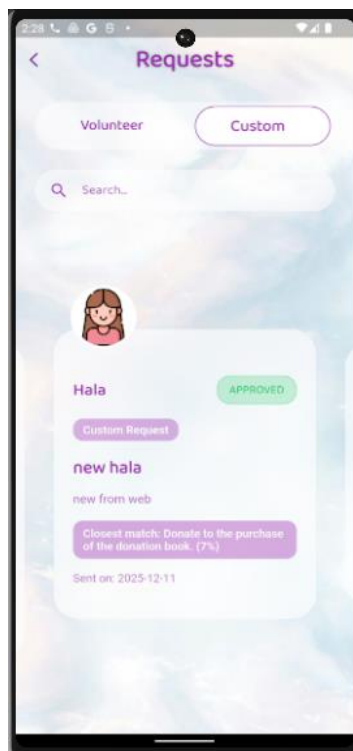


Figure3. 90: AI-Powered Request Analysis Requests Page.

Forms Submissions:

Once a student's volunteer request is approved by the Service Center, the student is required to submit the official community service form. This form must be signed and verified by the organization where the volunteering activity was completed. The submitted form is then reviewed by the Service Center to confirm completion and validate the recorded volunteer hours.

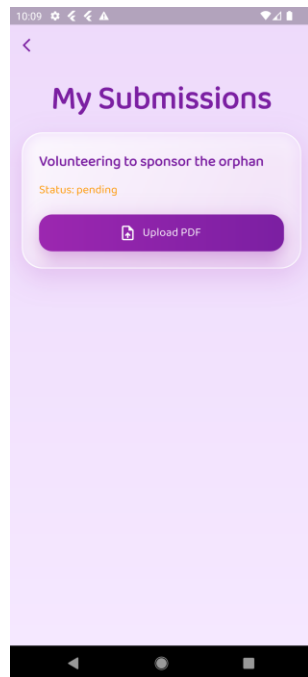


Figure3. 92: Student Upload File.

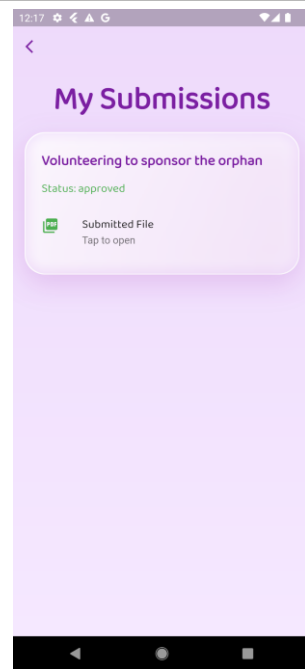


Figure3. 91: Student Submitted Form

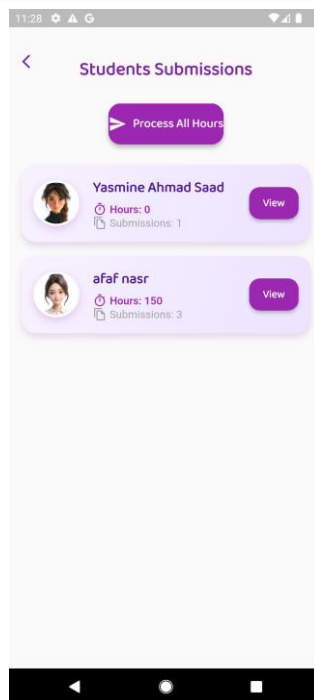


Figure3. 93: Center View All Submissions.

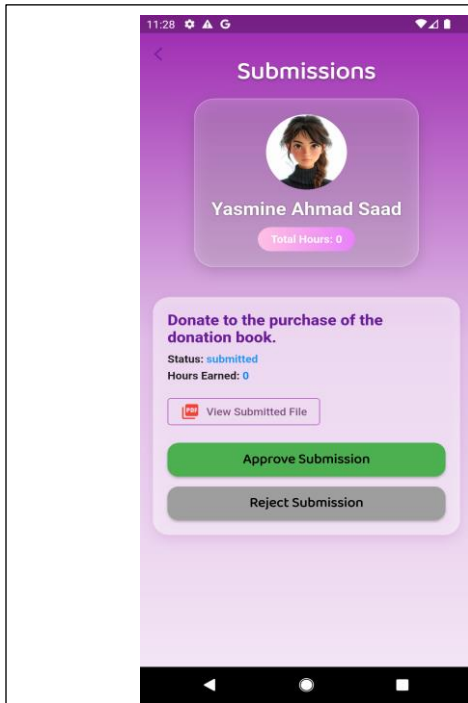


Figure3. 94: Center View Specific Student Submission.

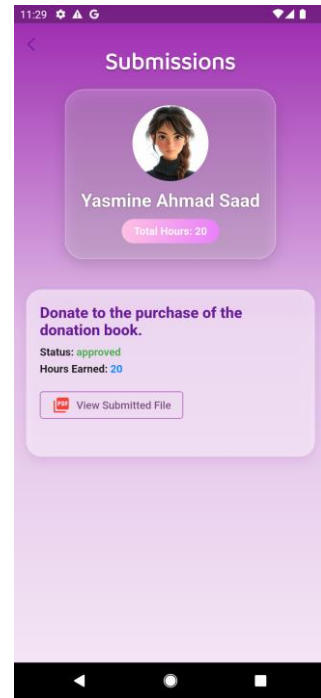


Figure3. 95: Center Approved Submissions.

Search and AI Assistance Feature:

The system provides an advanced search feature that allows students to search for activities and services available within the application. In addition, an integrated AI assistant enables students to ask questions and receive system-based responses, helping them quickly find relevant information and guidance related to community service activities.

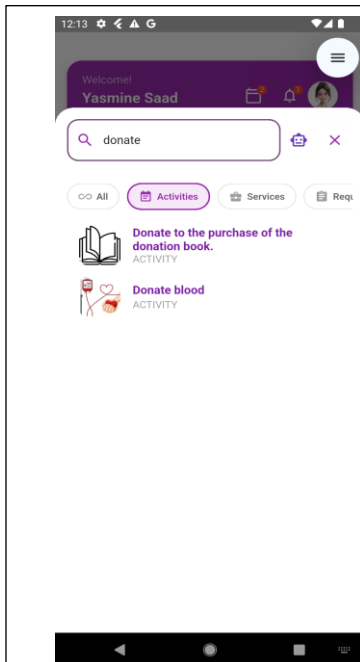


Figure3. 98: Activities Search Interface.

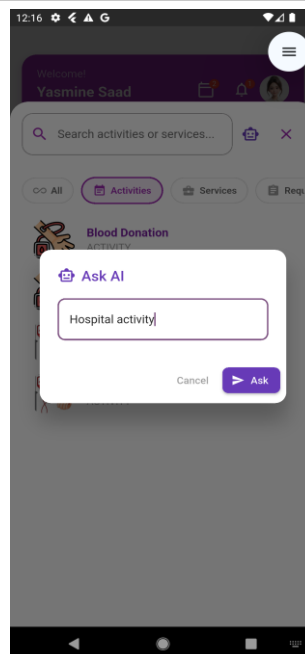


Figure3. 97: Ask AI Interface.

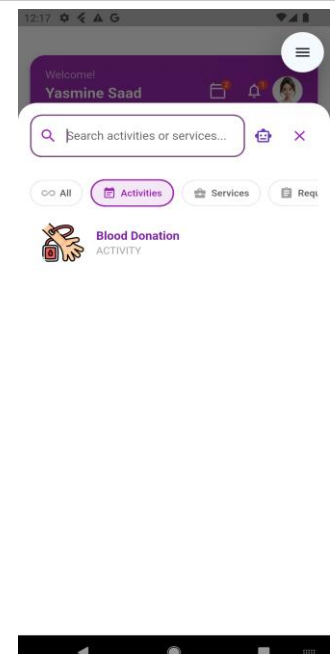


Figure3. 96: Search Results Display.

In the Center Side: The system provides a search feature that allows Service Centers to search for students within the platform. This helps centers quickly locate specific students, view their profiles, and manage communication or service-related processes efficiently.

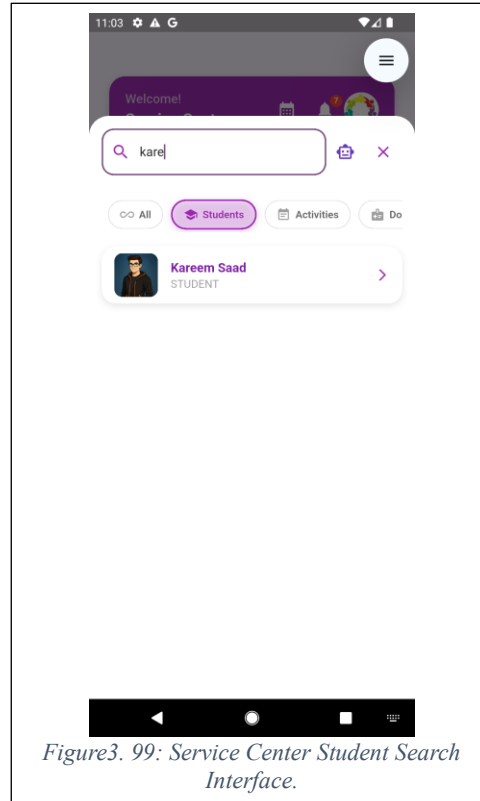


Figure3. 99: Service Center Student Search Interface.

FAQs & Student Questions Feature:

The FAQs & Questions feature provides a direct communication channel between students and the Service Center. Students can view frequently asked questions, submit new inquiries, and track their questions, while the Service Center can respond, manage FAQs, and provide official answers to ensure clarity and support.

In the Center Side:

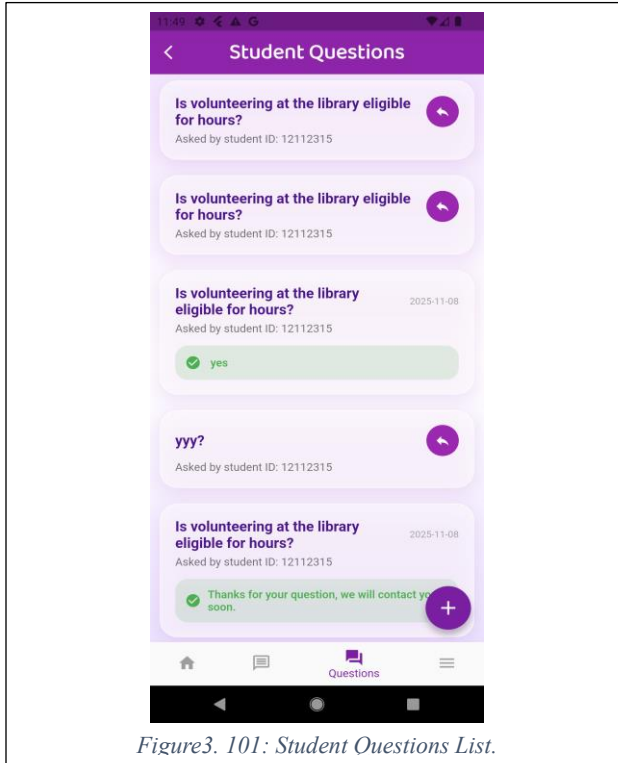


Figure3. 101: Student Questions List.

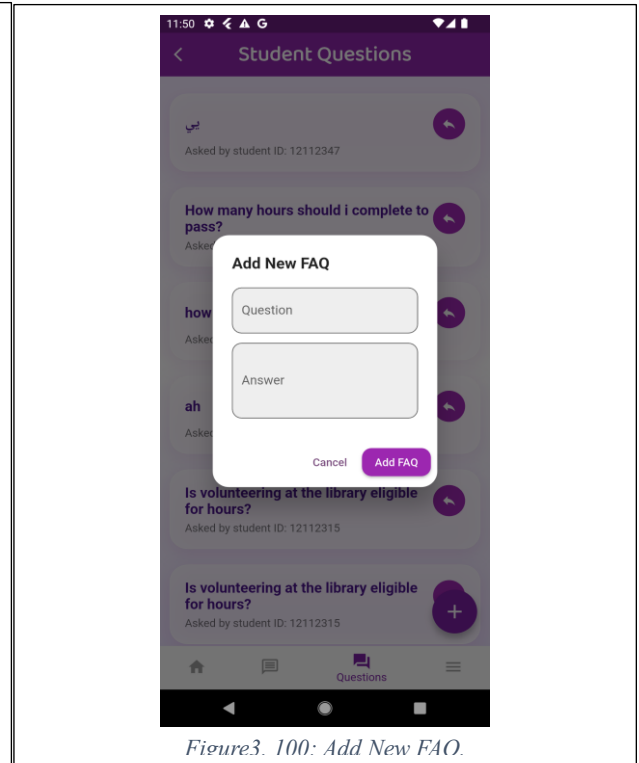


Figure3. 100: Add New FAQ.

In the Student Side:

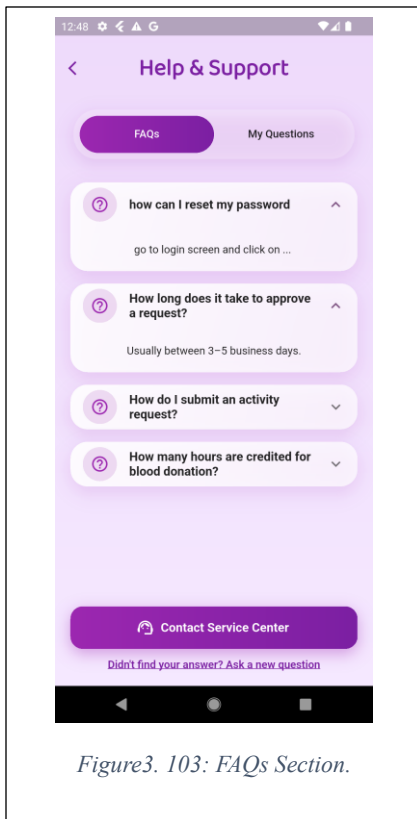


Figure3. 103: FAQs Section.

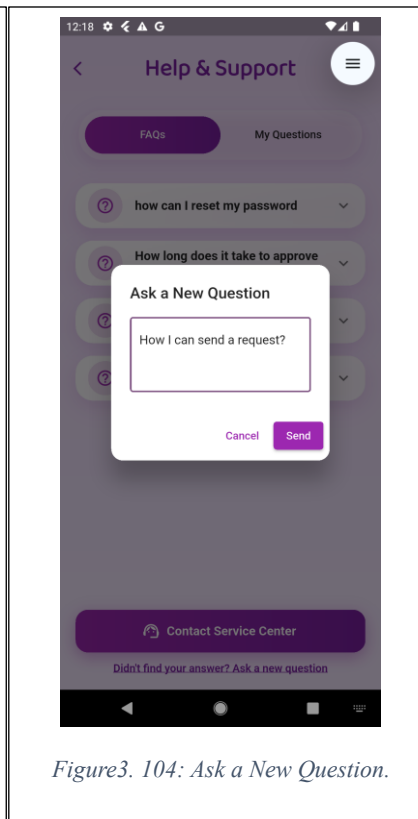
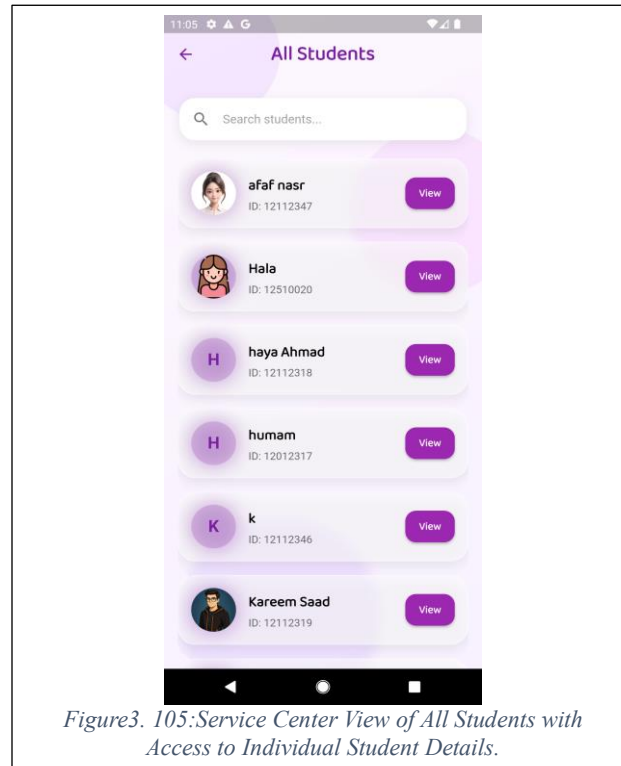


Figure3. 104: Ask a New Question.



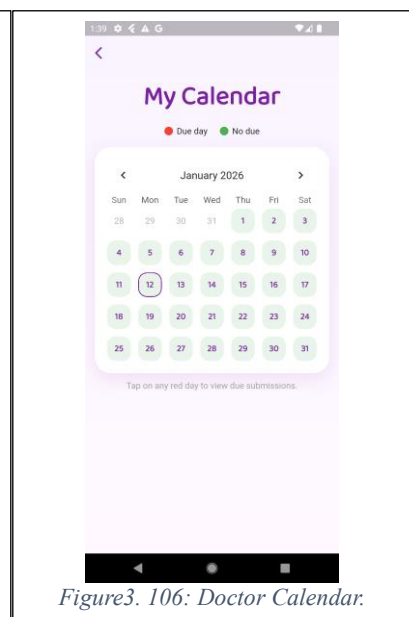
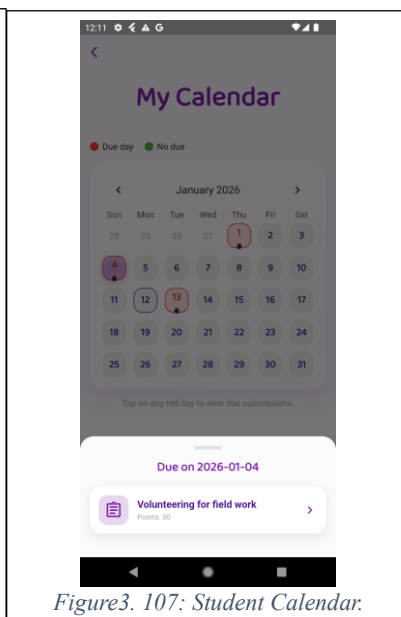
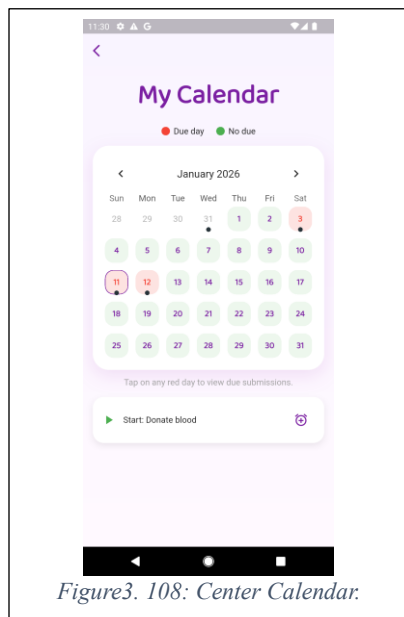
Figure3. 102: Students Questions.

The Service Center can **view a complete list of all registered students** and easily search through them. By selecting any student, the Service Center can access detailed information related to that student, enabling effective monitoring and management.



Calendar Feature:

The Calendar feature allows students, doctors, and service centers to track important activity dates such as start dates, end dates, and submission deadlines. It helps users stay organized and ensures timely completion of tasks and requirements related to community service activities.



Doctor Feature: Student Evaluation and Monitoring:

- The system automatically evaluates students based on their completed volunteering hours and verified submissions received from the Service Center.
- The doctor can view all assigned students and their progress, while **the final pass/fail decision is generated by the system** according to predefined criteria.
- The Service Center is responsible for tracking student activities, verifying submissions, and communicating the results to the system, which then updates each student's status accordingly.

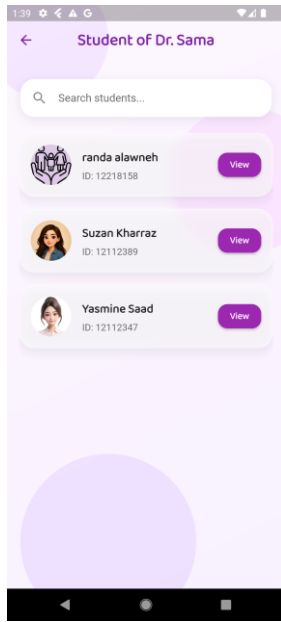


Figure3. 110: Doctor's Students List.

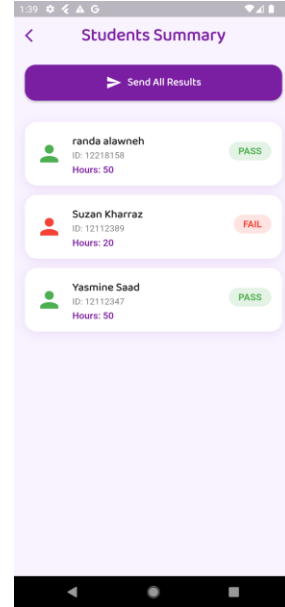


Figure3. 109: Students Summary.

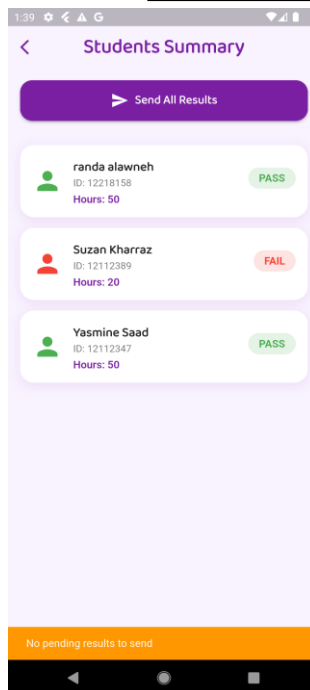


Figure3. 111: Final Results Status.

Admin Role:

Roles & Permissions Management Feature:

- The system provides a fully dynamic Roles & Permissions management feature managed by the Service Center.
- The Service Center can create new roles, edit or delete existing ones, and assign specific permissions to each role.
- This allows flexible control over system access, ensuring that every user (Admin, Doctor, Student, or Service Center) only has the capabilities required for their responsibilities.

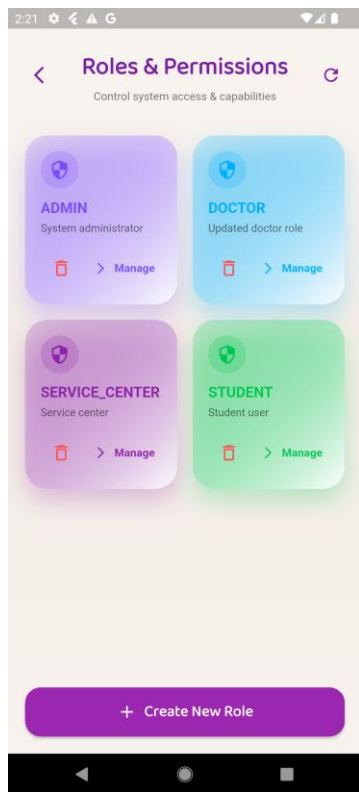


Figure3. 113: Roles & Permissions Overview.

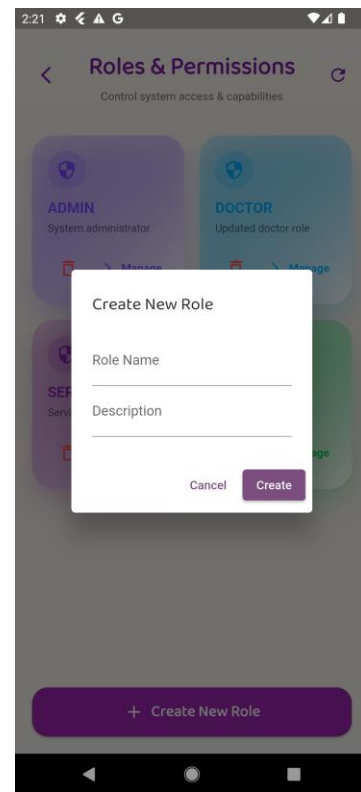


Figure3. 112: Create a new role dynamically.

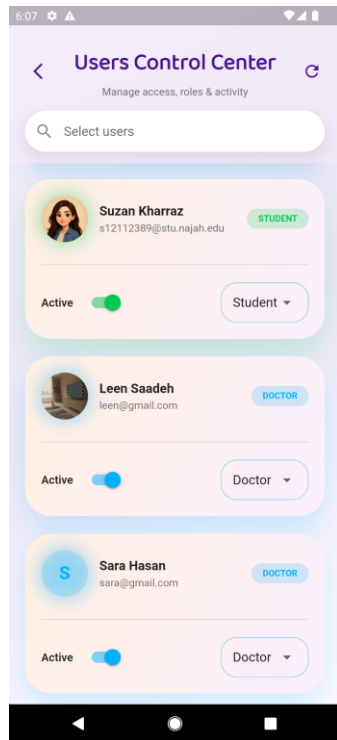


Figure3. 115: Assign roles, activate or deactivate users, and manage access.

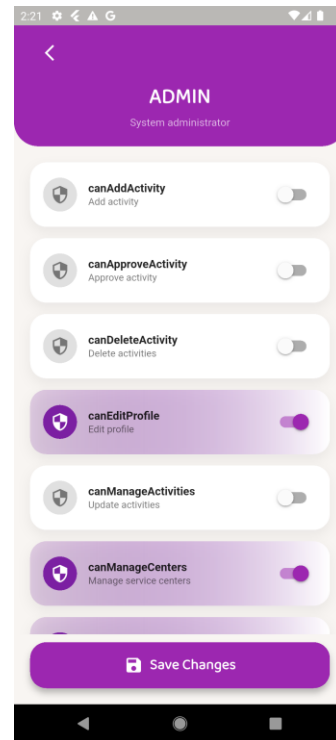


Figure3. 114: Admin Permissions Management.

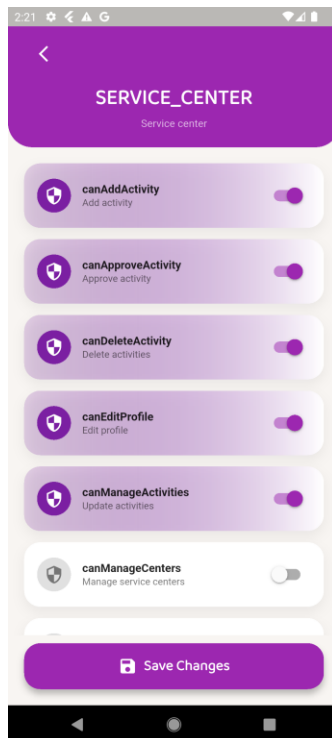


Figure3. 116: Service Center Permissions Management.

Confirm Student Feature:

The admin can import an Excel file containing the names and details of students officially registered in the course. This process is used to verify students and grant them approval, allowing only eligible students to sign in and access the application securely.

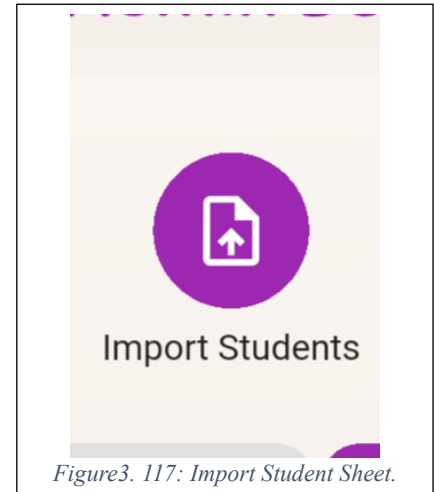


Figure3. 117: Import Student Sheet.

Assigning Students to Doctors Feature:

The admin has full control over managing both doctors and students within the system. This feature allows the admin to assign students to their corresponding doctors, ensuring proper supervision and organization, especially when multiple doctors are teaching the same course. By managing these assignments, the system guarantees that each student is linked to the correct doctor for follow-up, evaluation, and coordination throughout the course.

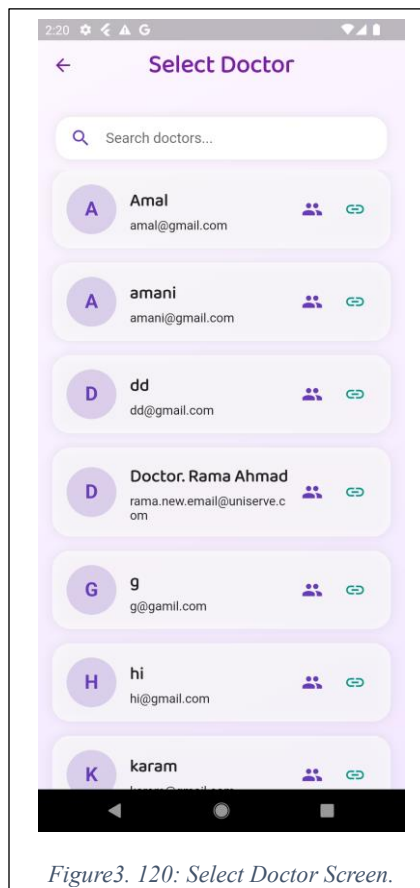


Figure3. 120: Select Doctor Screen.

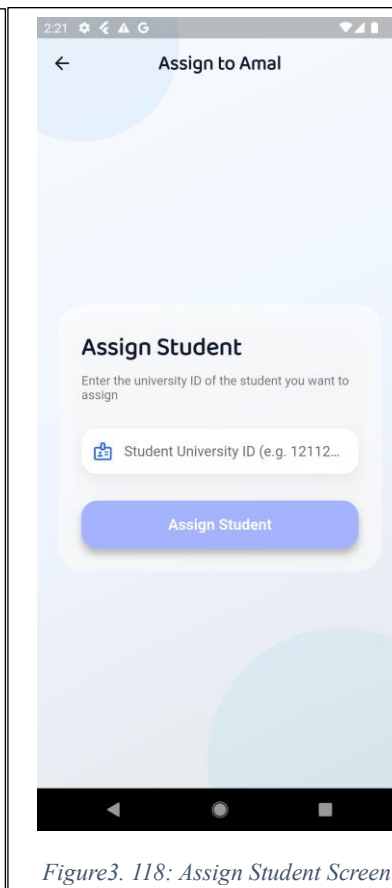


Figure3. 118: Assign Student Screen.

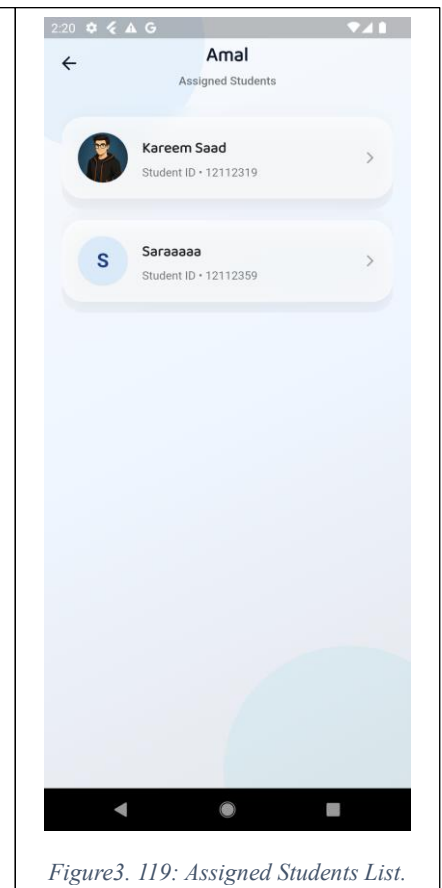


Figure3. 119: Assigned Students List.

3.2.2Website:

Web Platform Support

The system is fully supported on the web and provides the same features and functionalities available on the mobile application. All roles (Admin, Service Center, Doctor, and Student) can use the web version seamlessly, with a consistent user experience, complete workflows, and real-time synchronization.

This ensures flexibility and accessibility across devices without any limitation in performance or functionality.

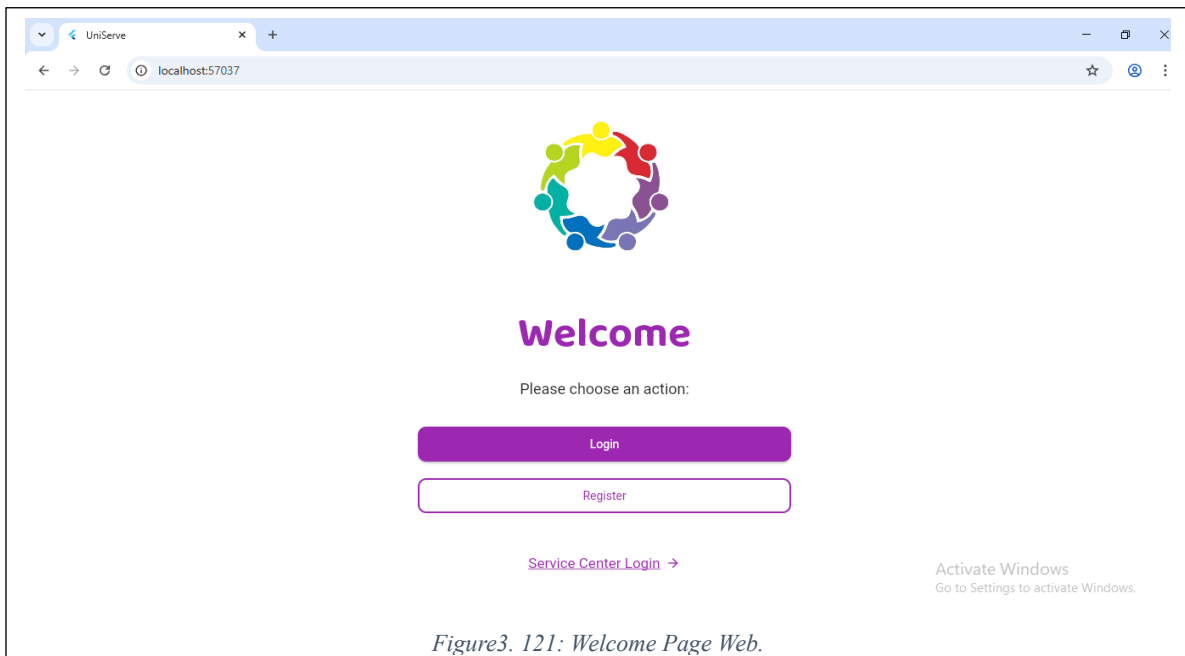


Figure3. 121: Welcome Page Web.

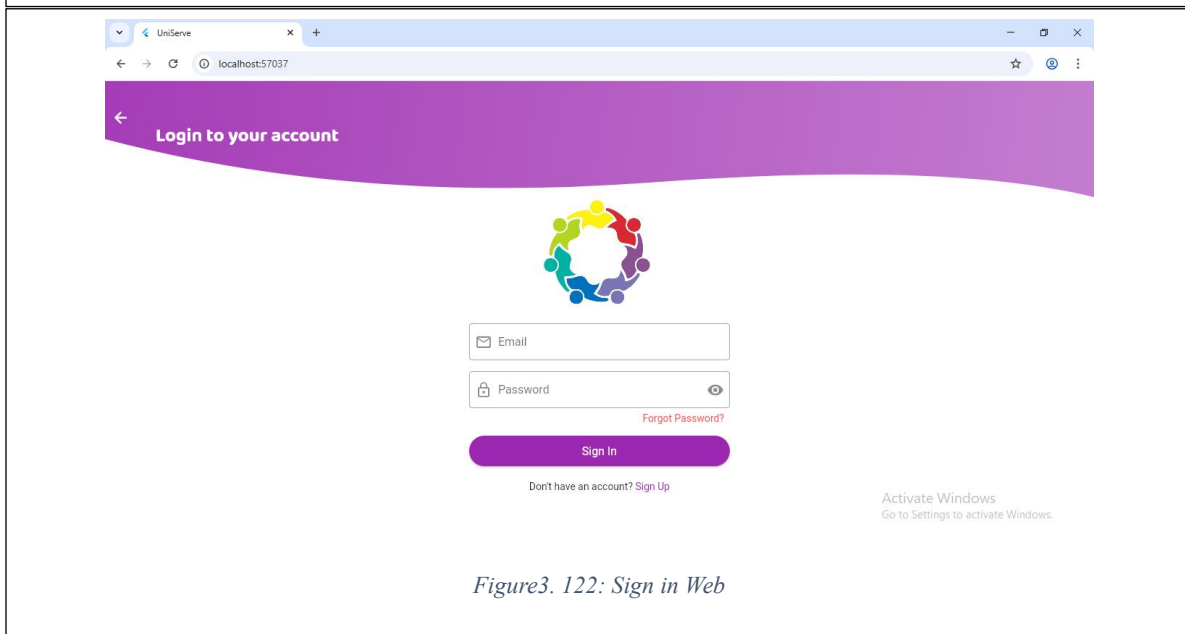
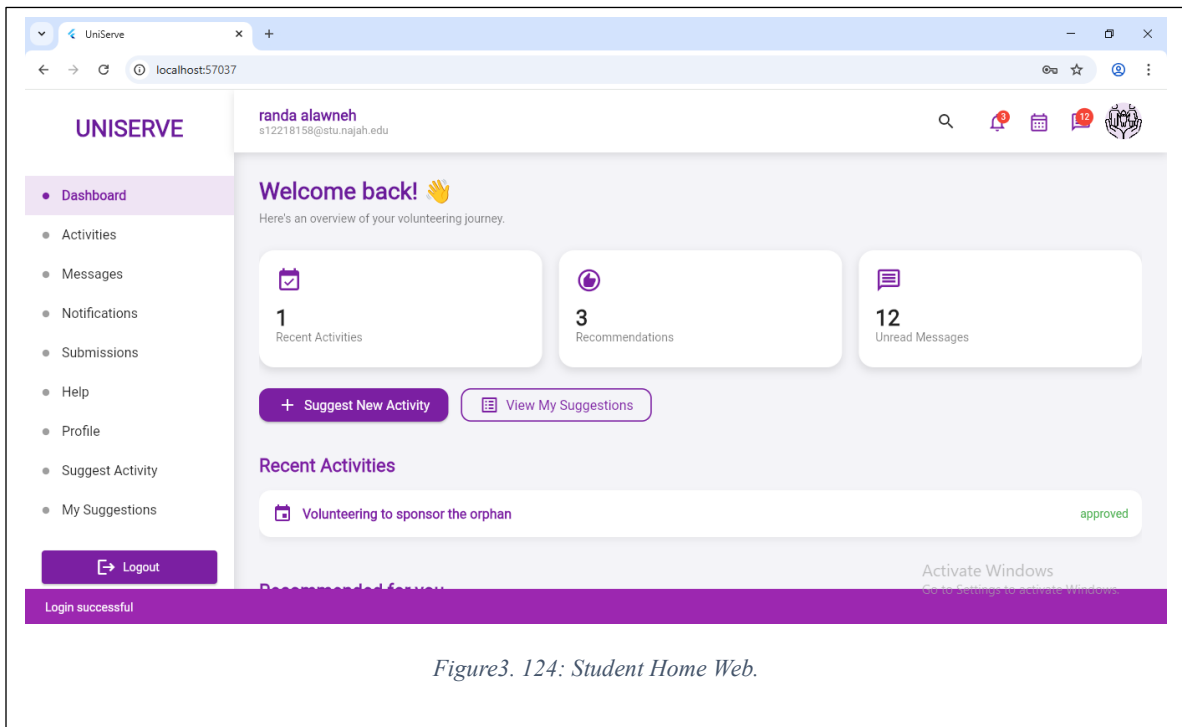
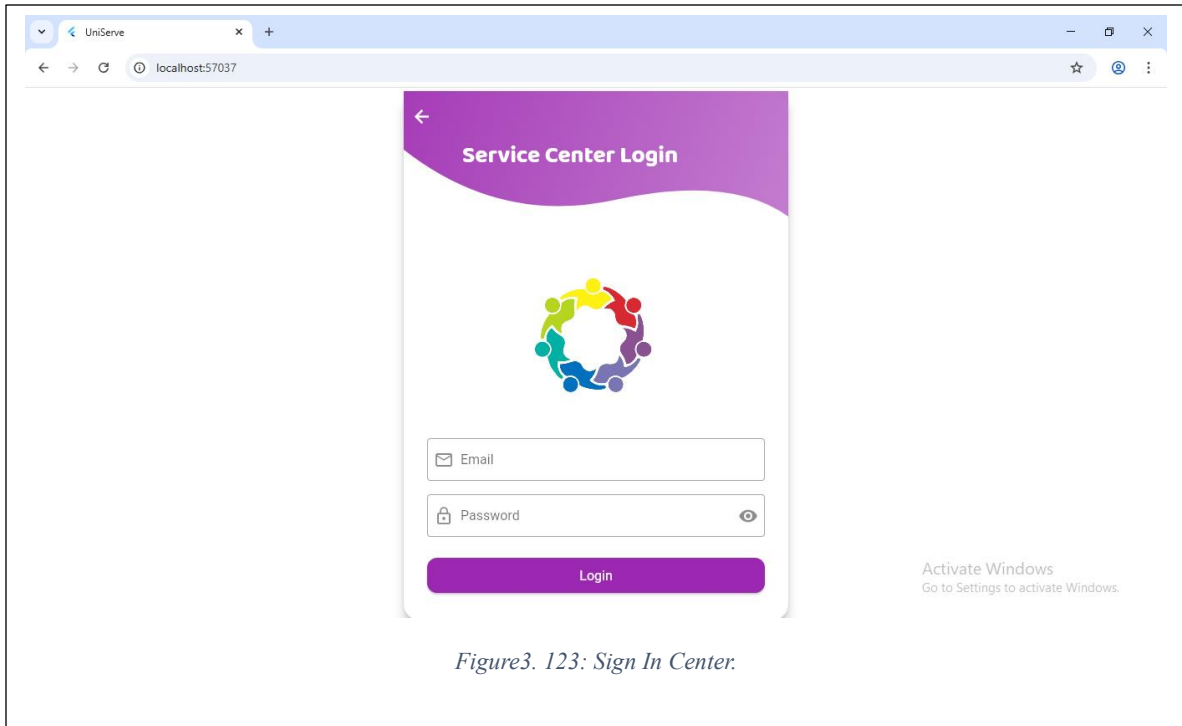
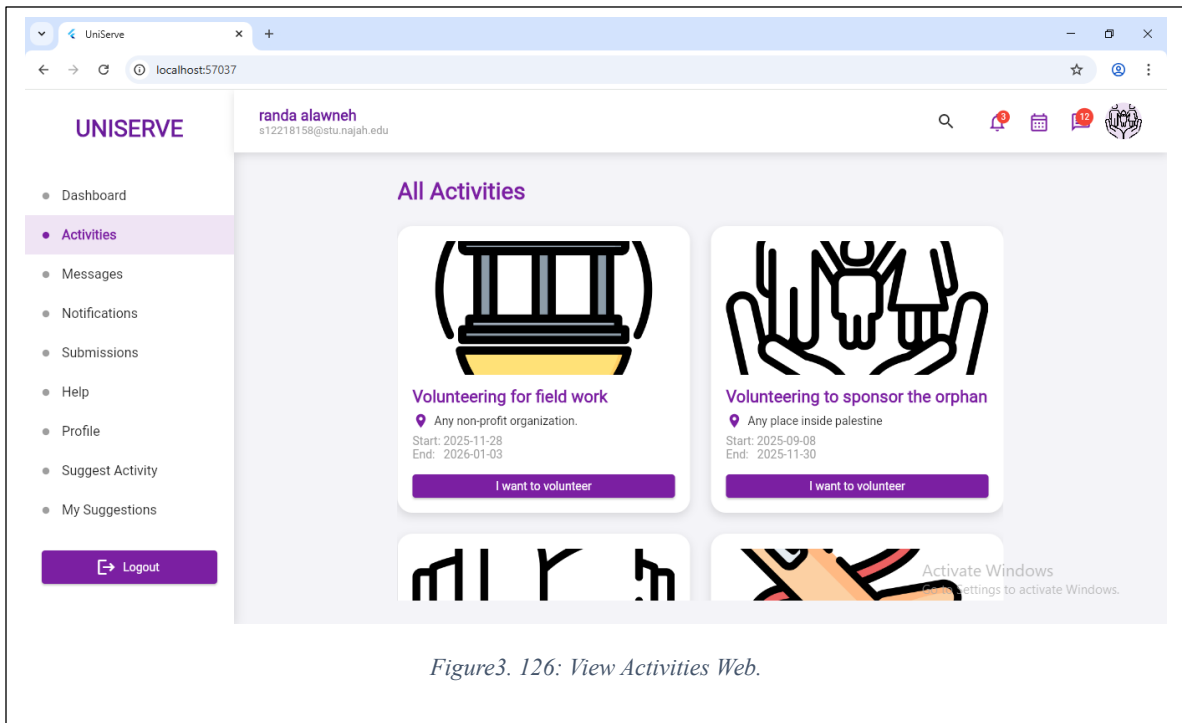
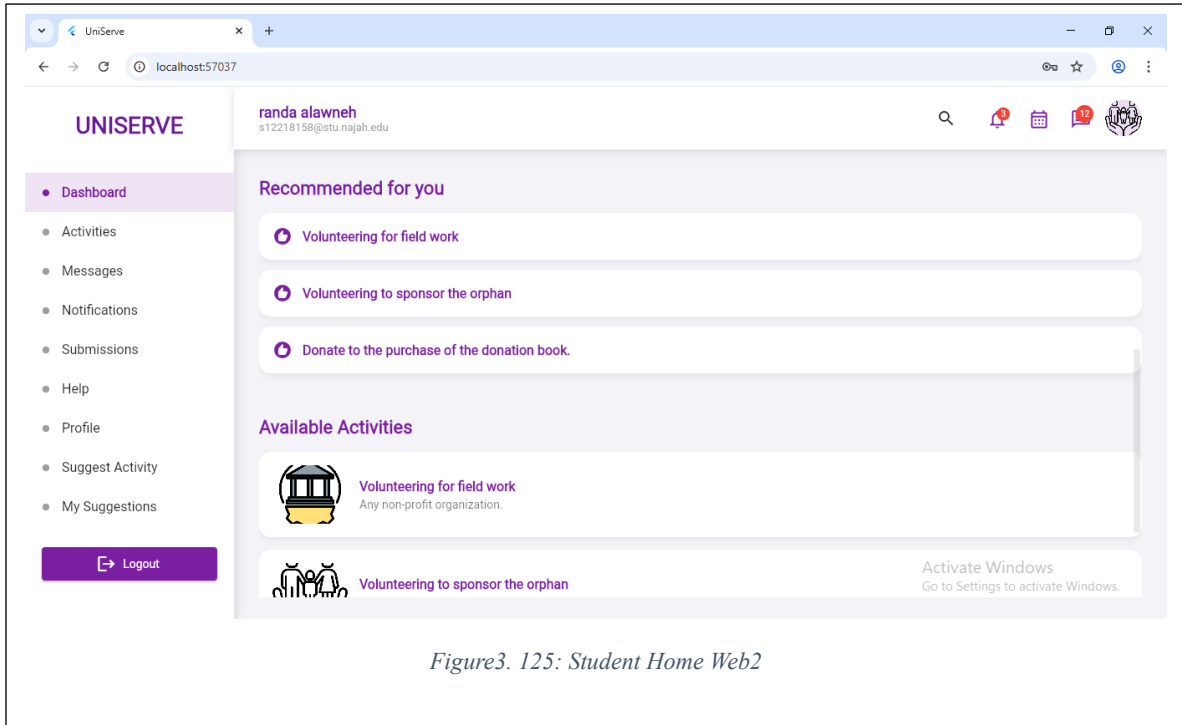
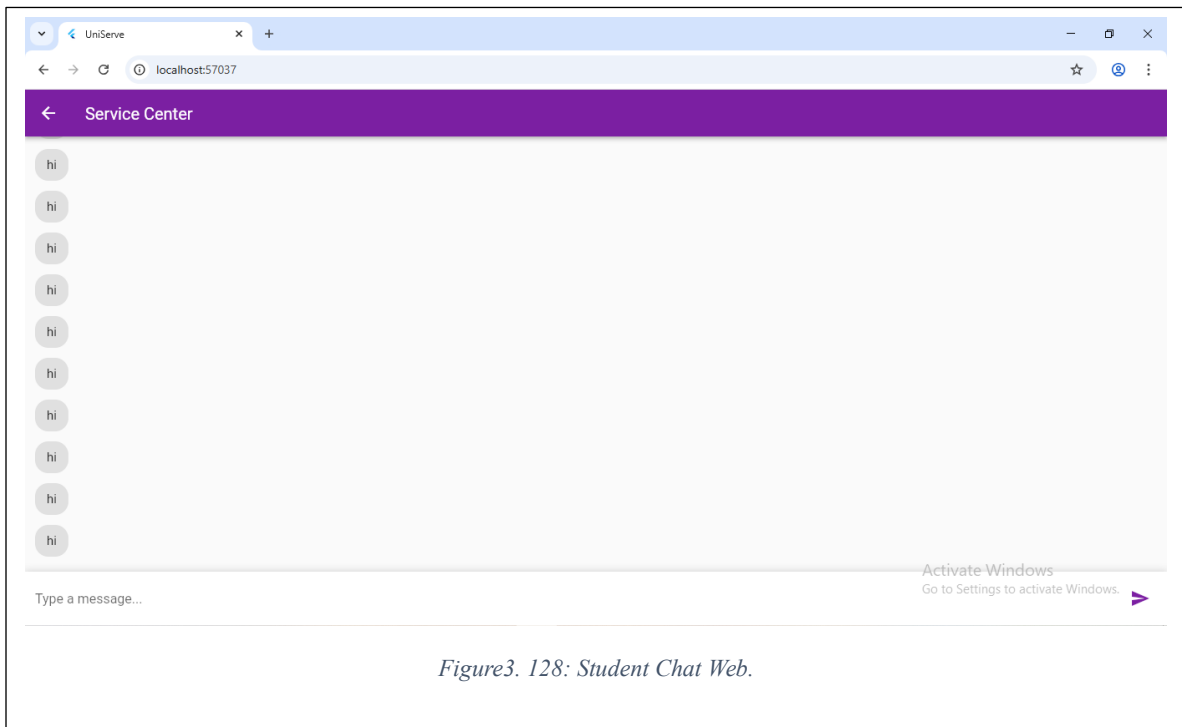
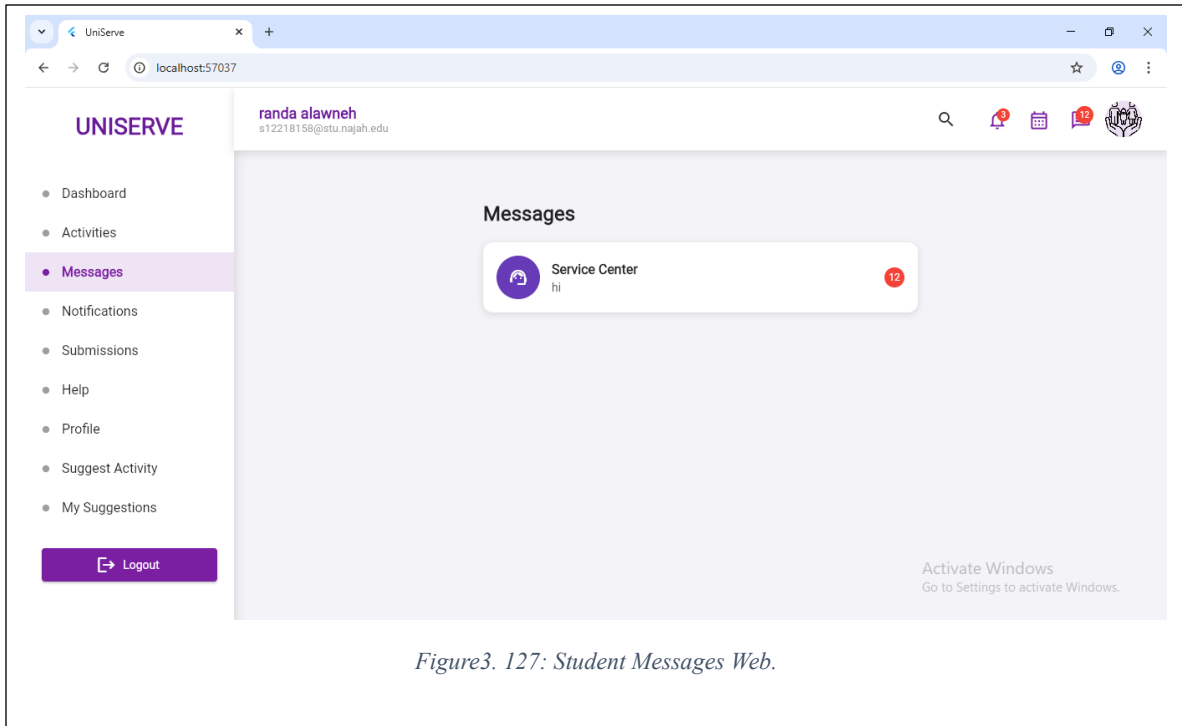


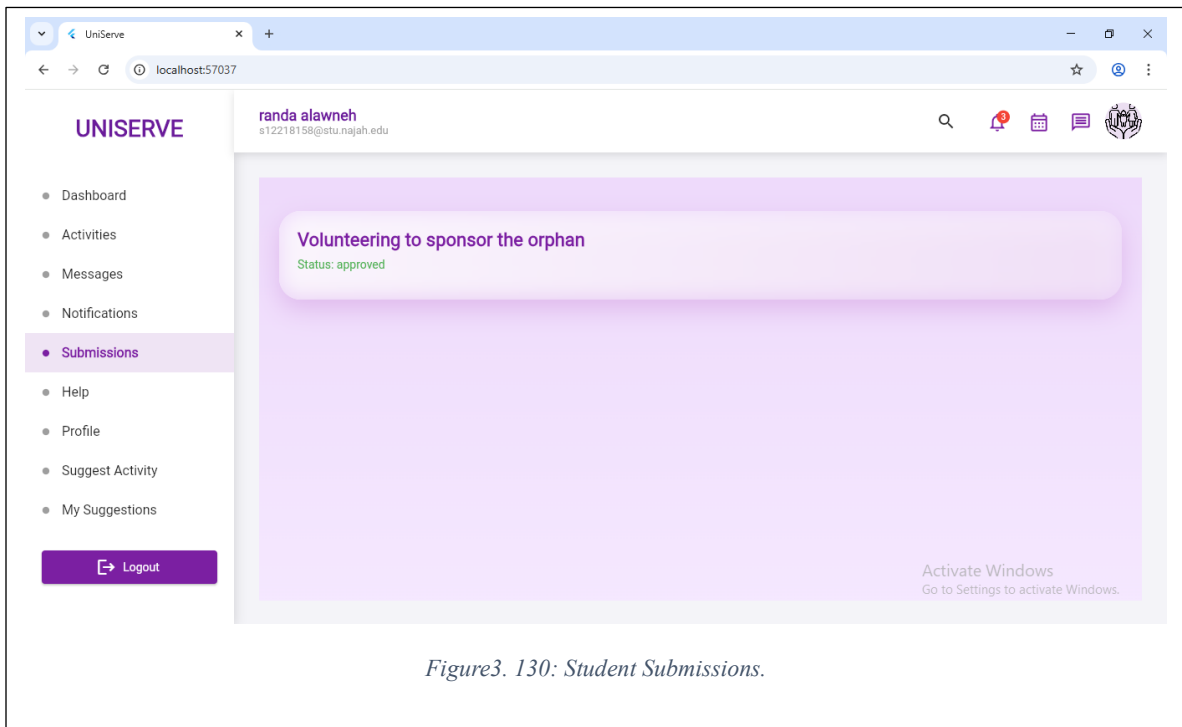
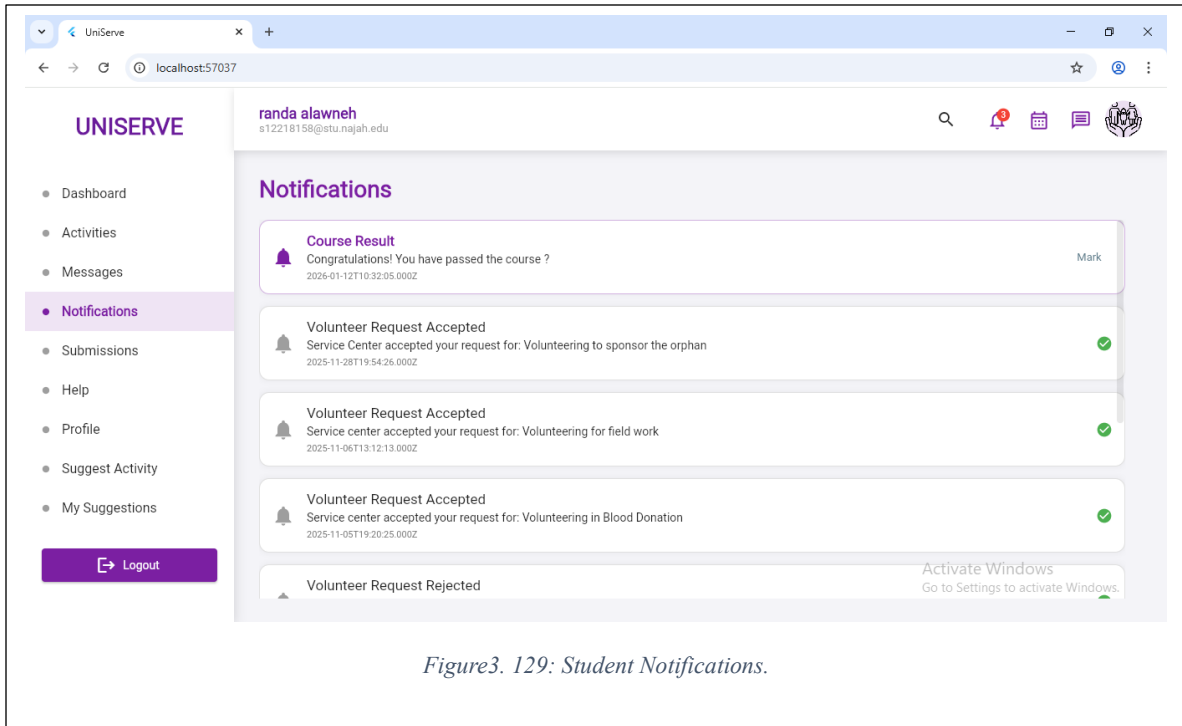
Figure3. 122: Sign in Web

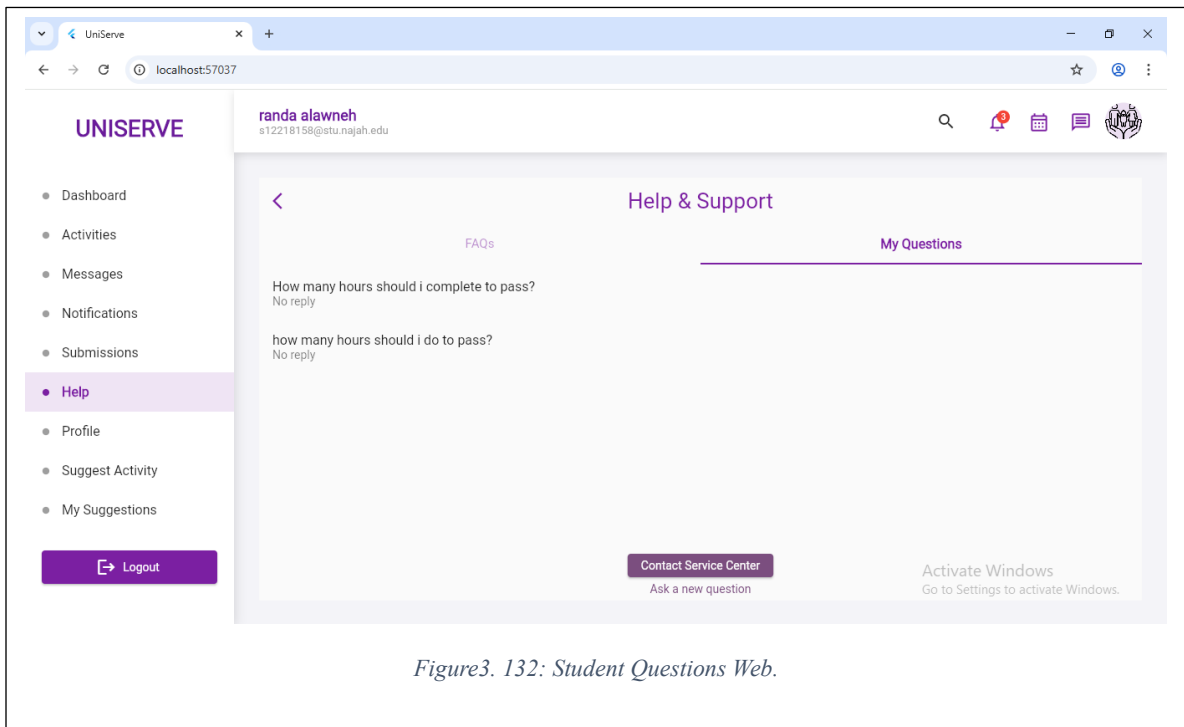
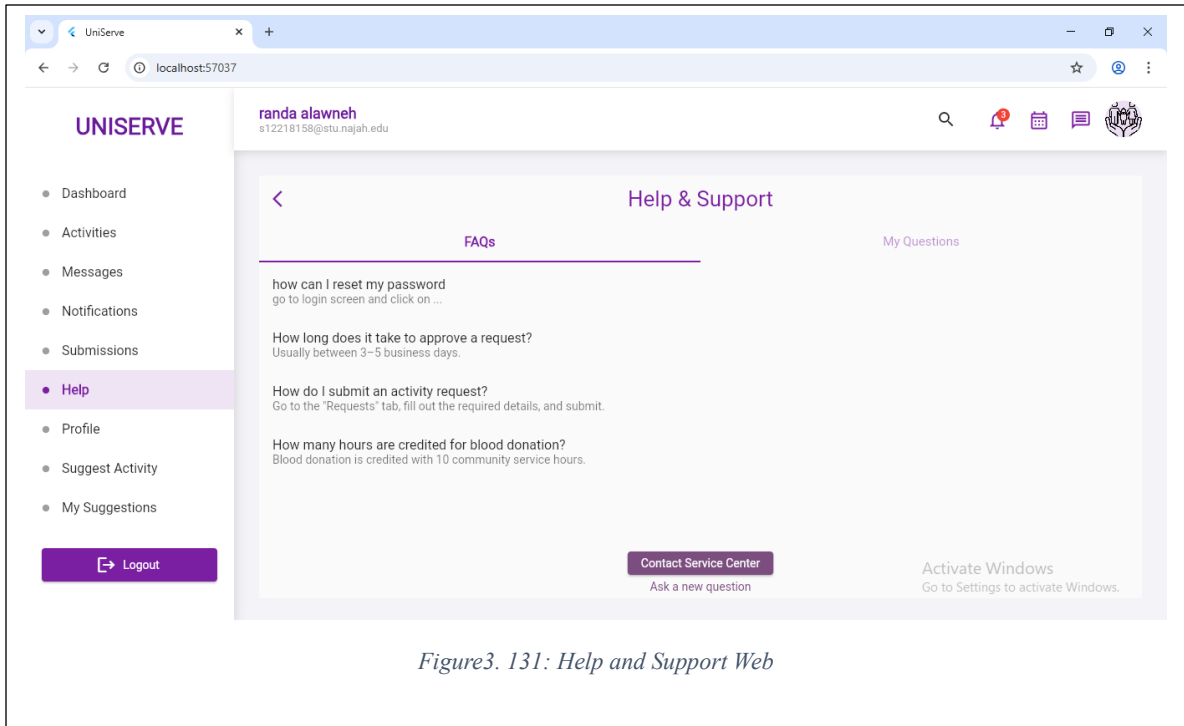


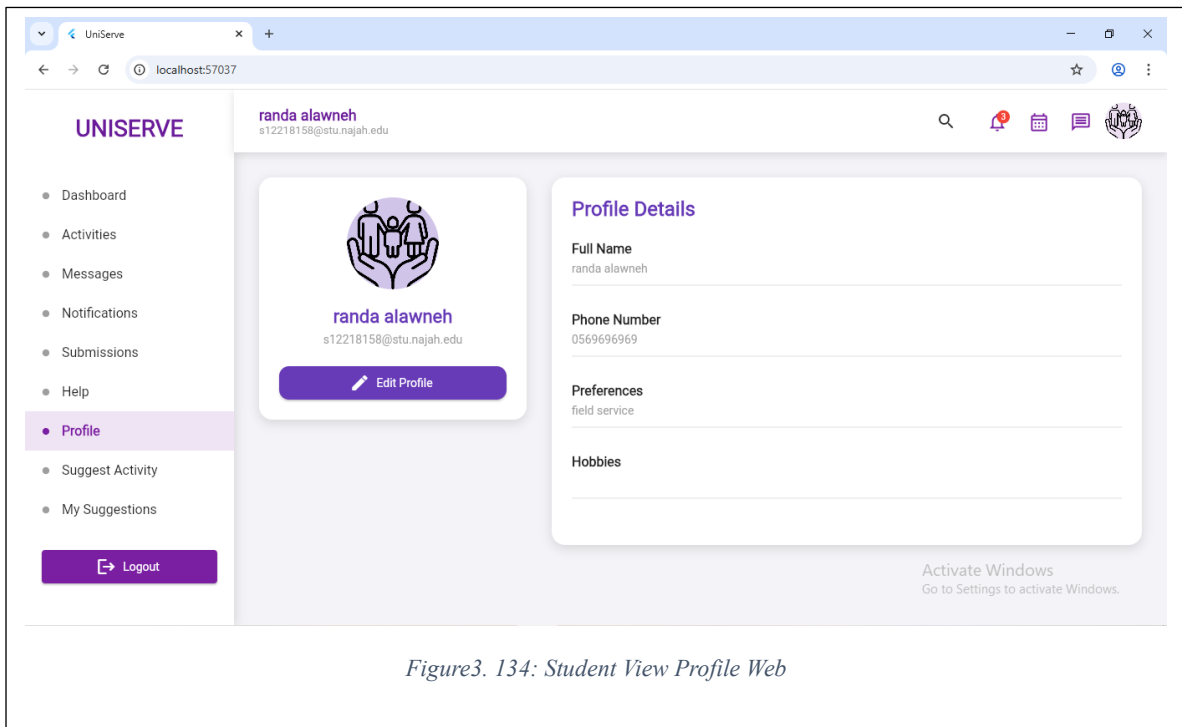
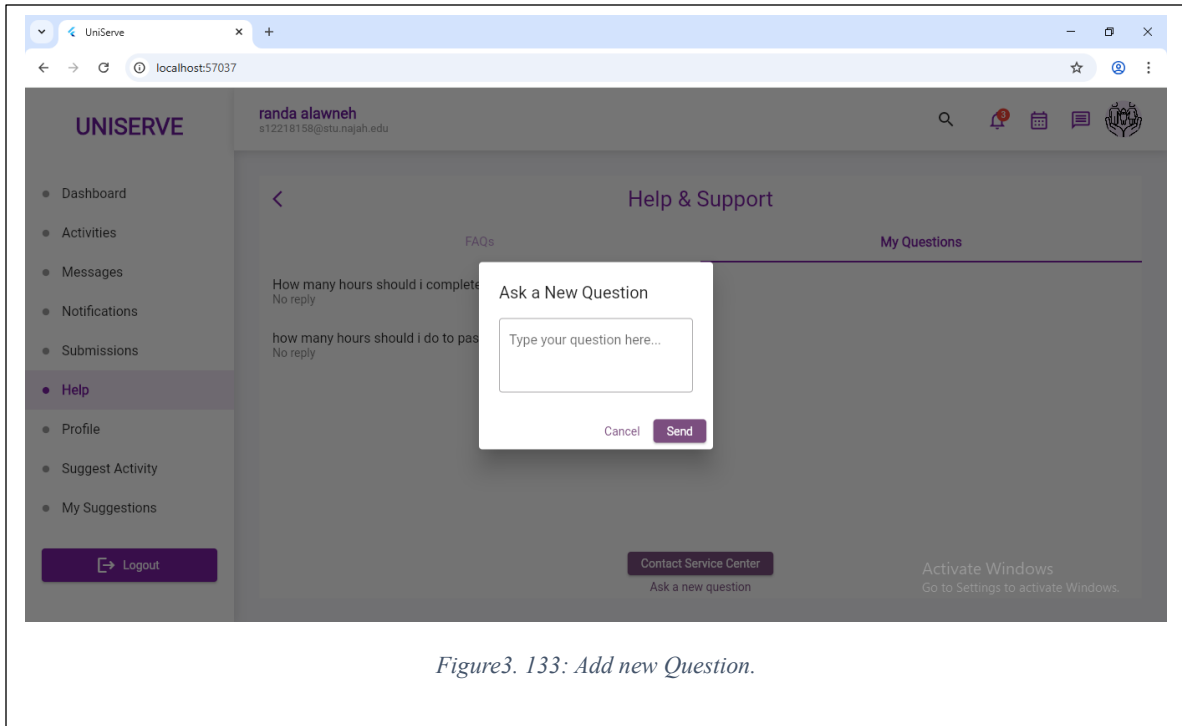
Student Role:

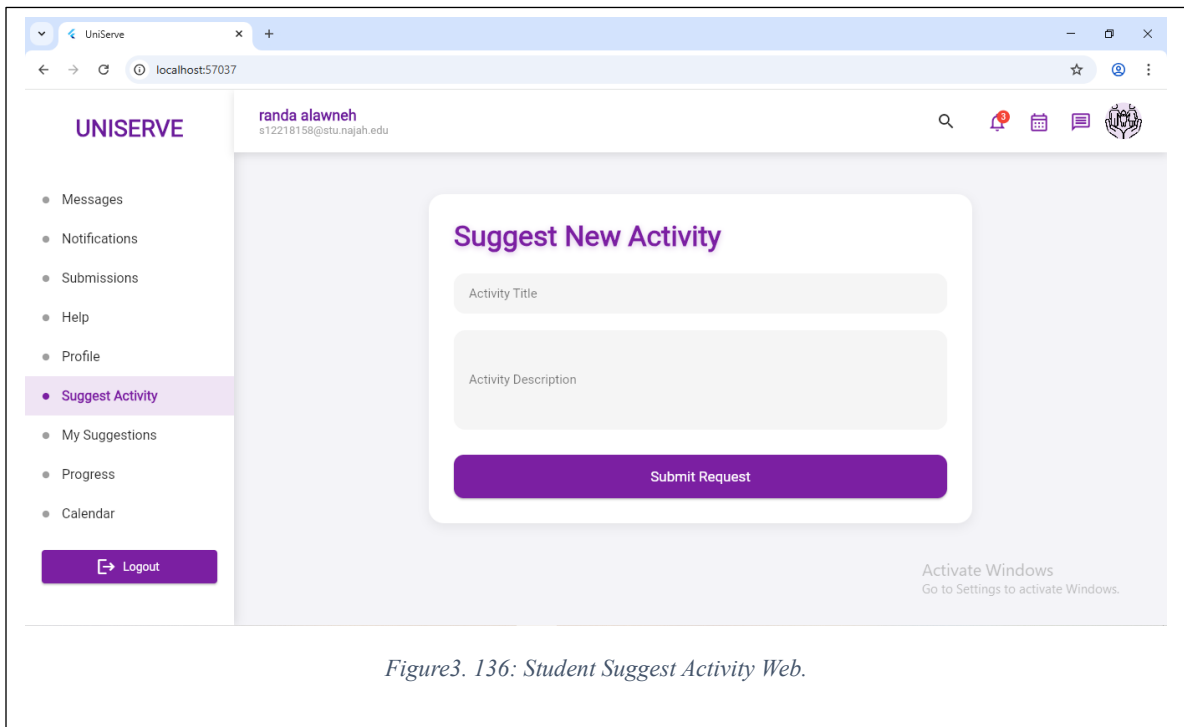
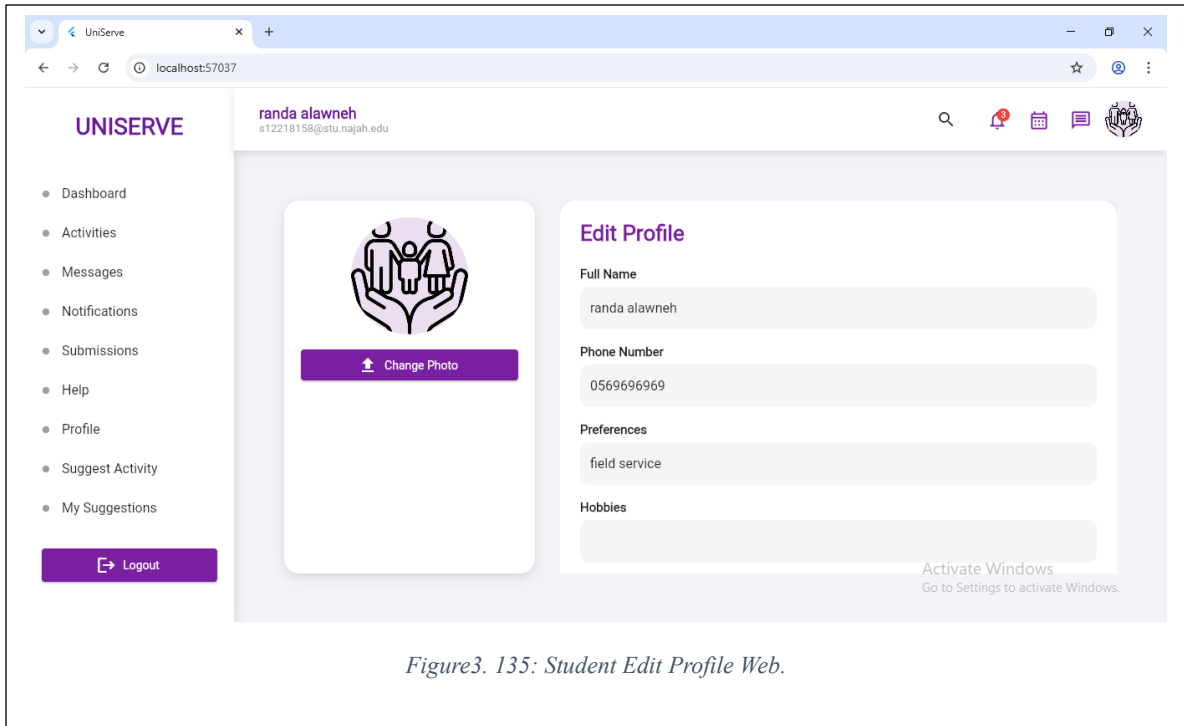


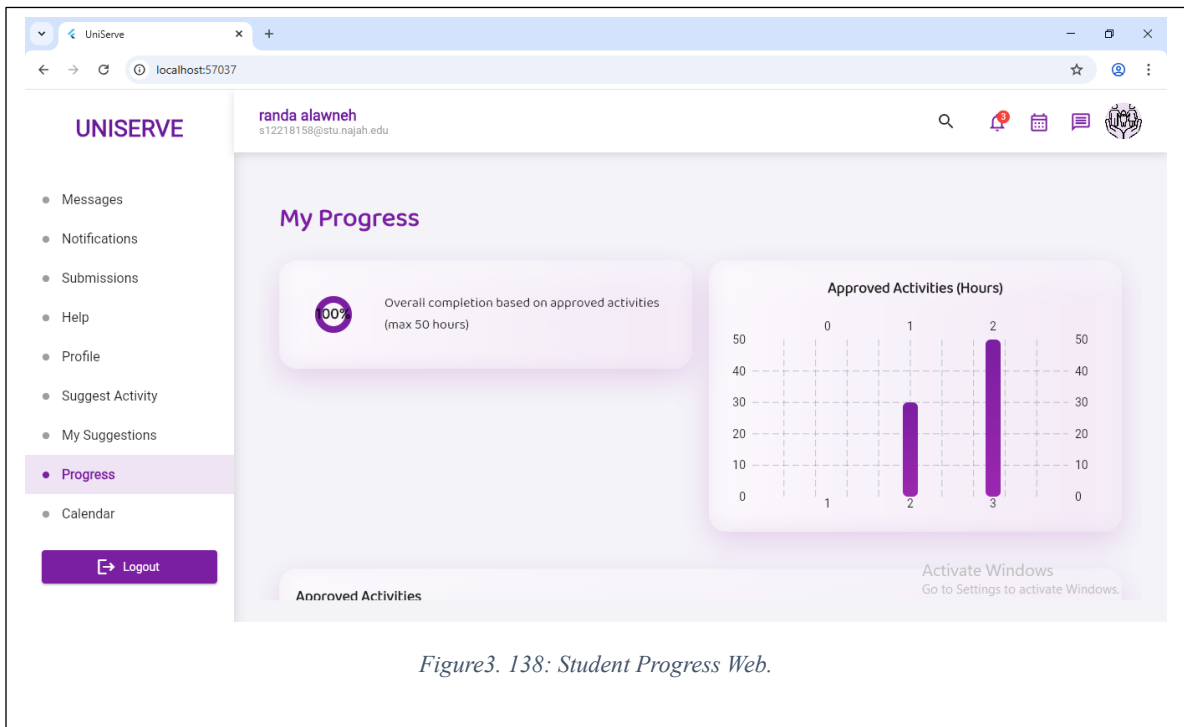
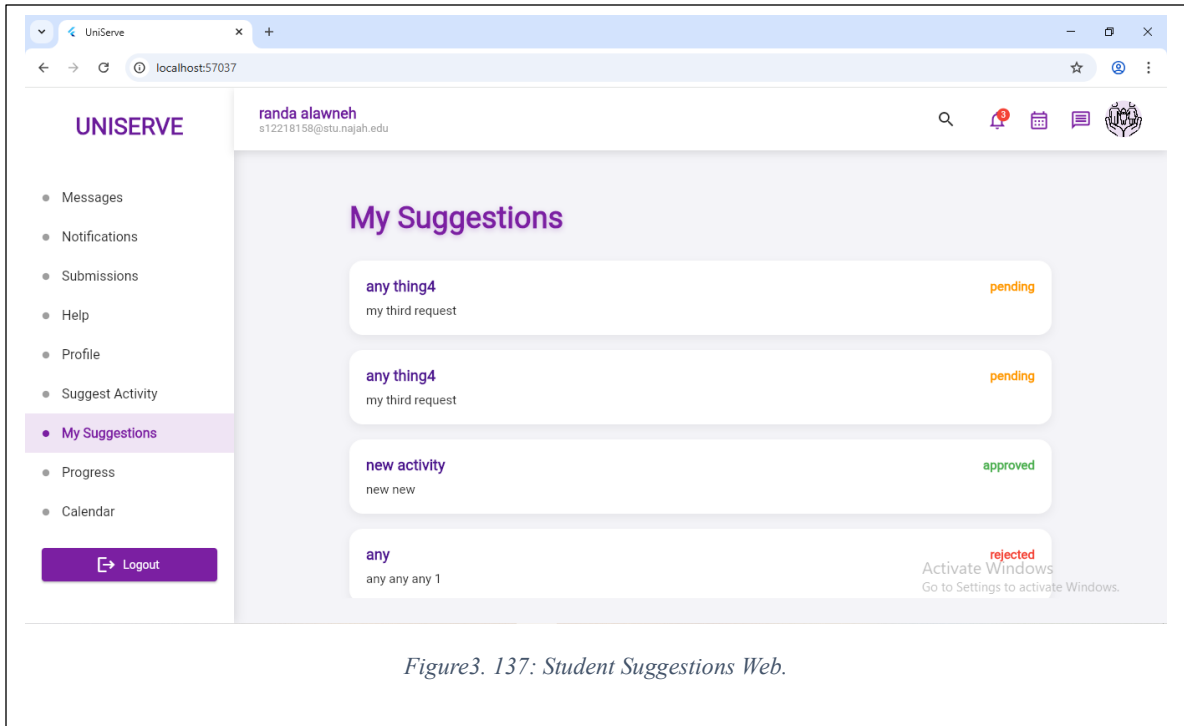


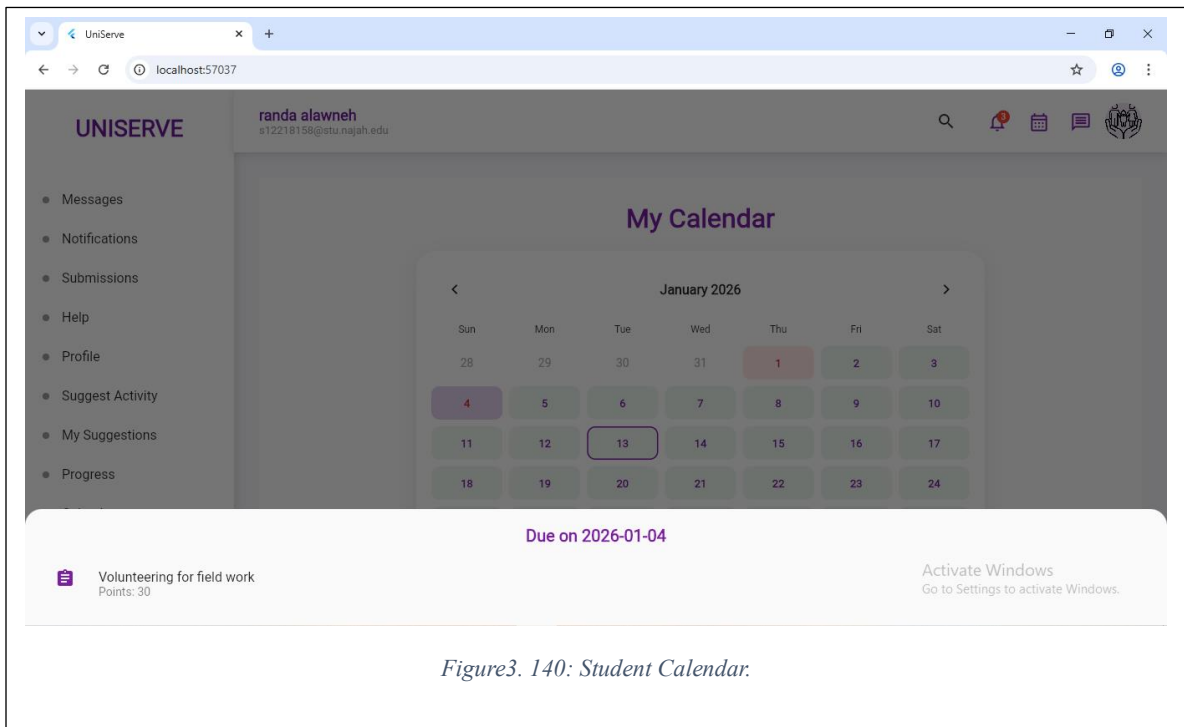
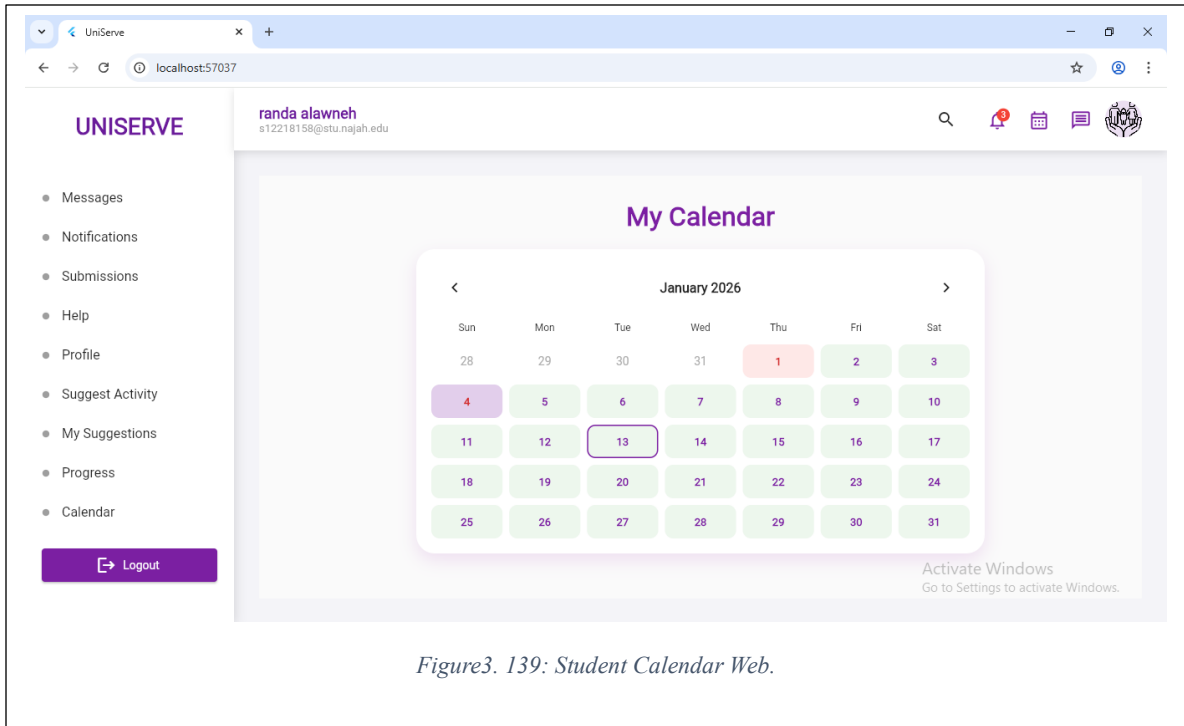




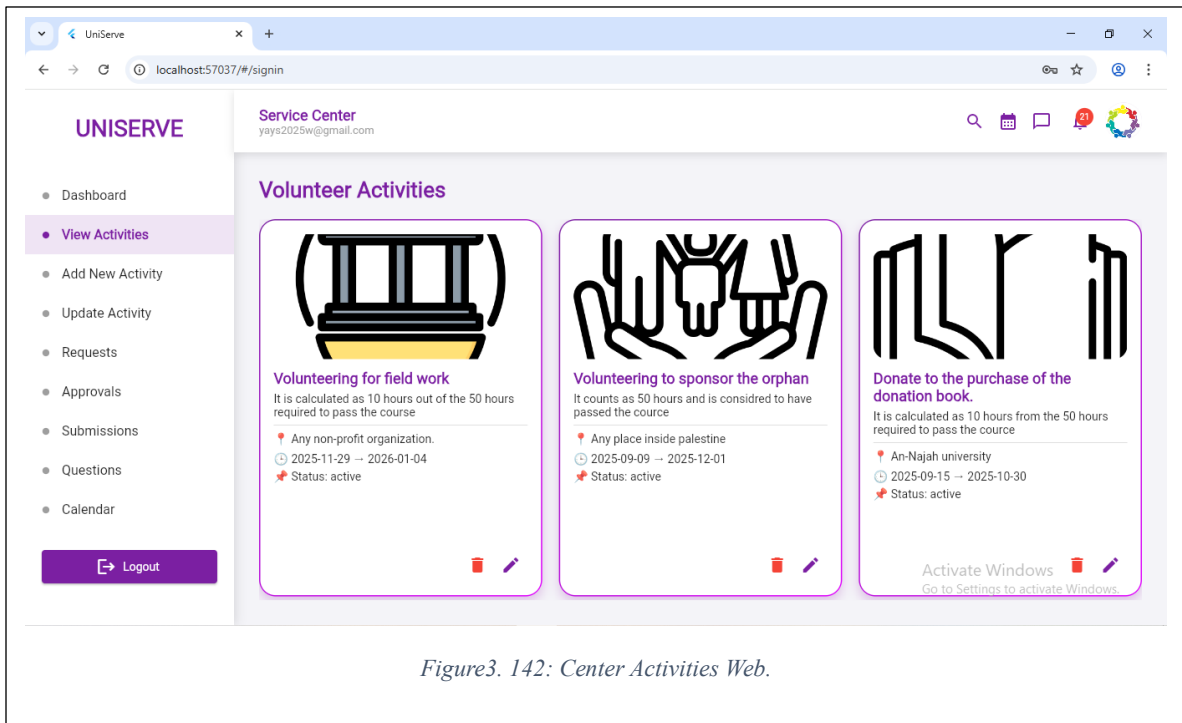
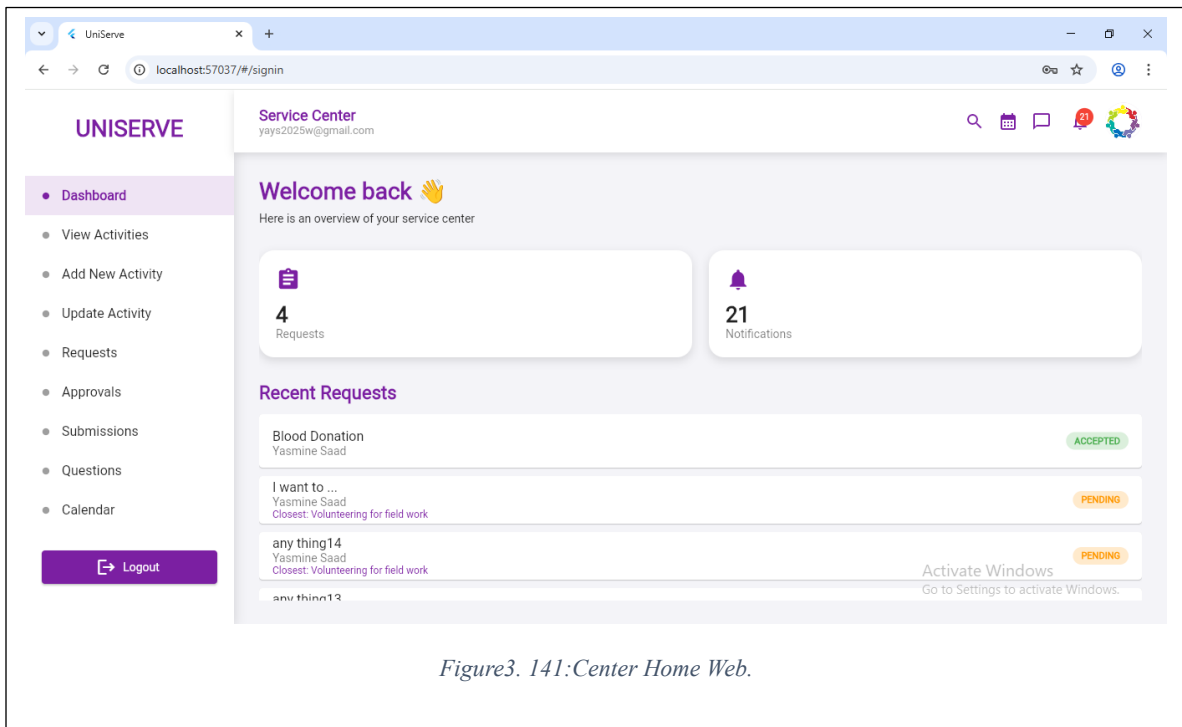








Center Role:



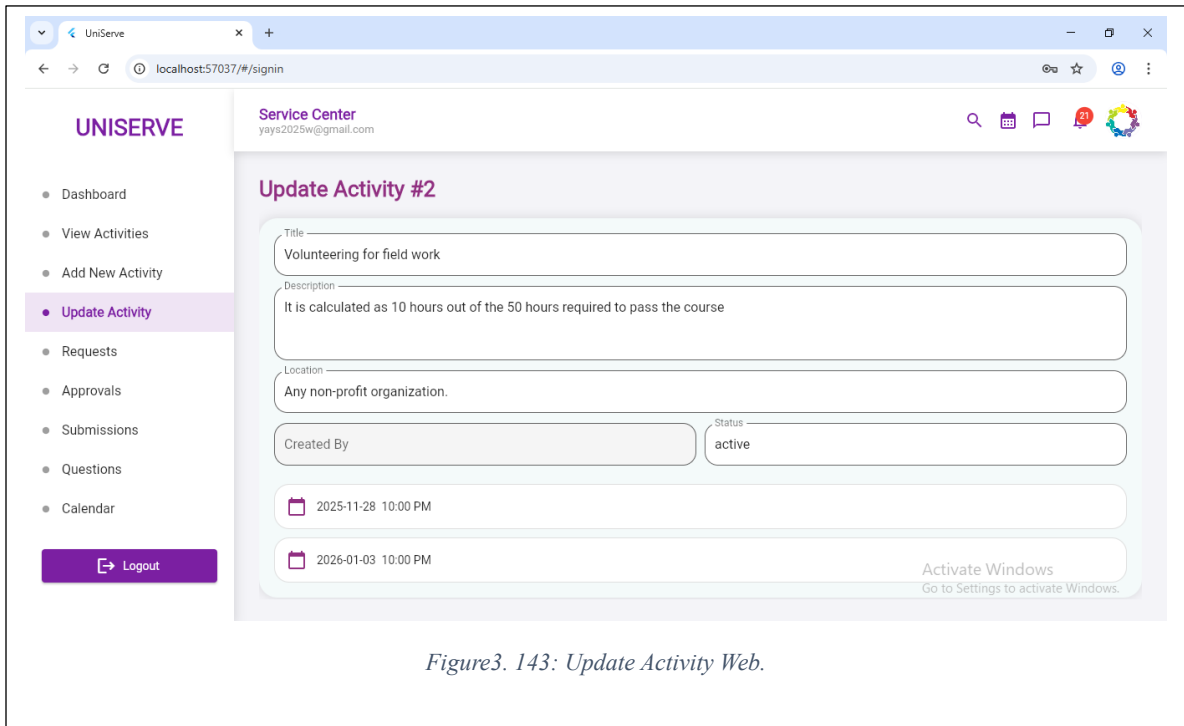


Figure3. 143: Update Activity Web.

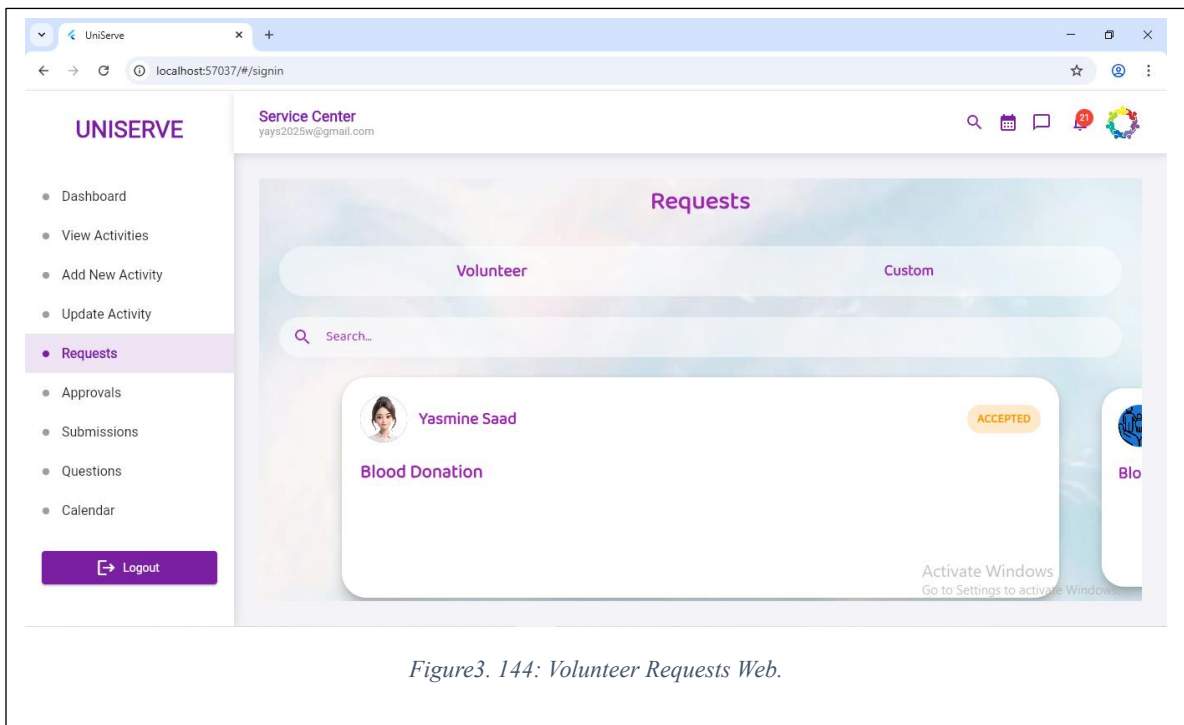


Figure3. 144: Volunteer Requests Web.

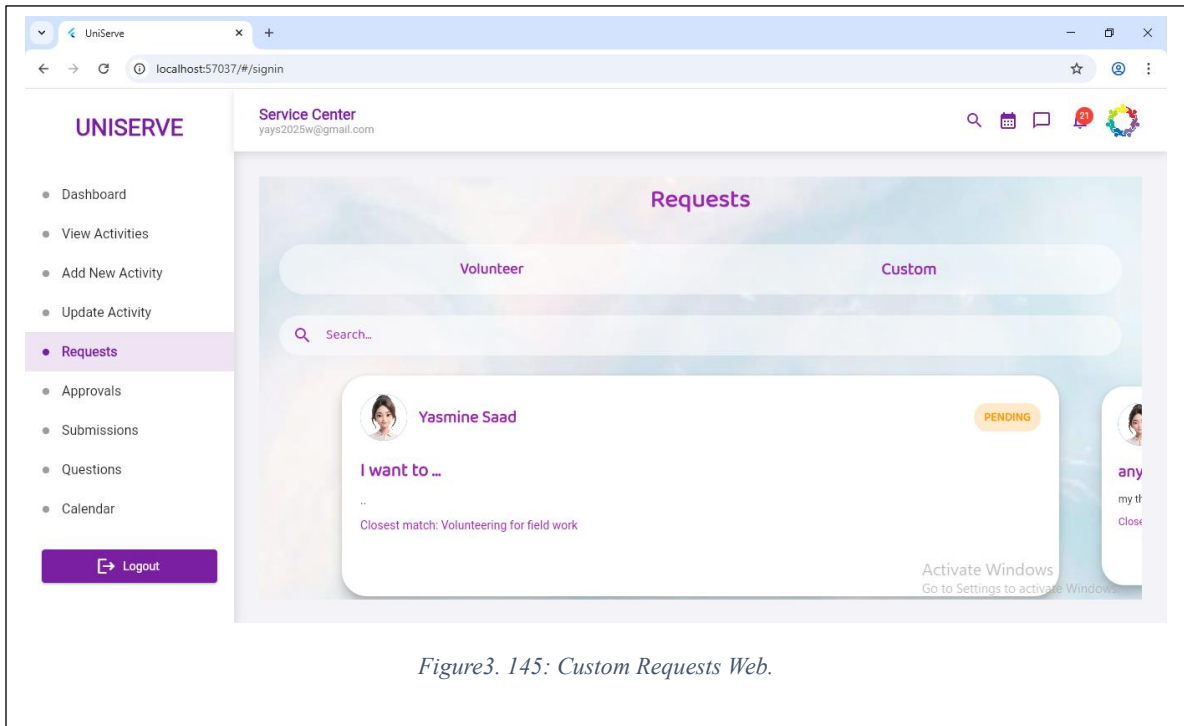


Figure3. 145: Custom Requests Web.

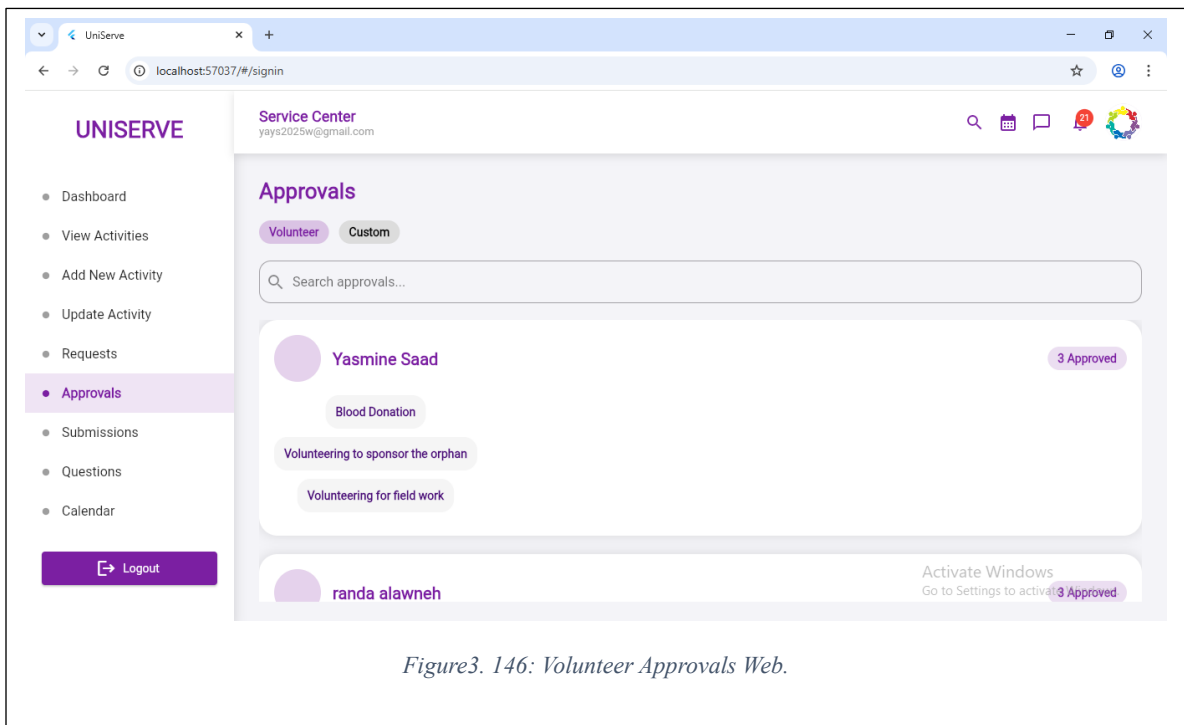
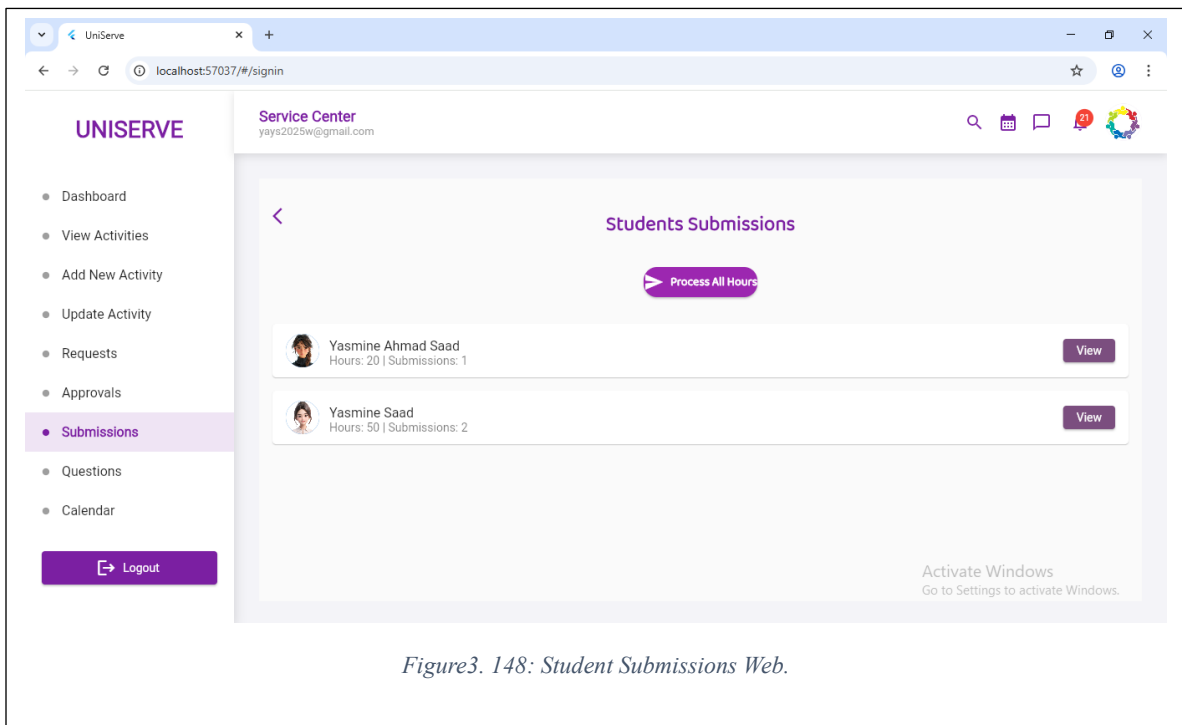
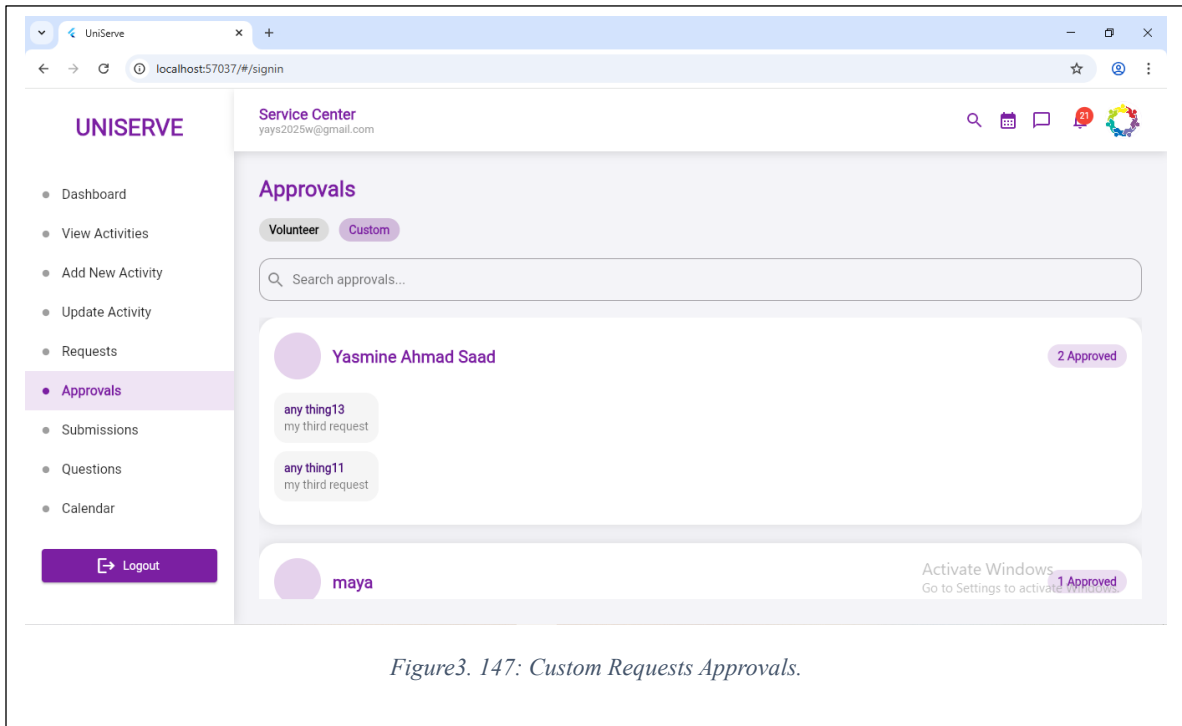
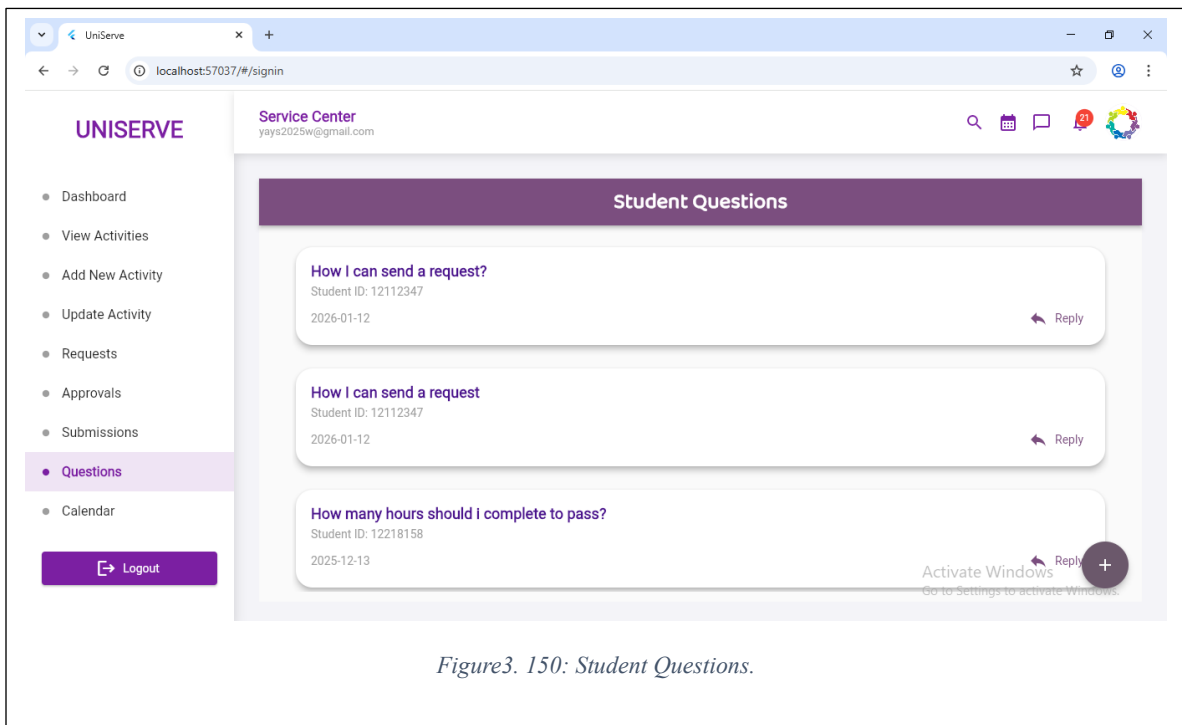
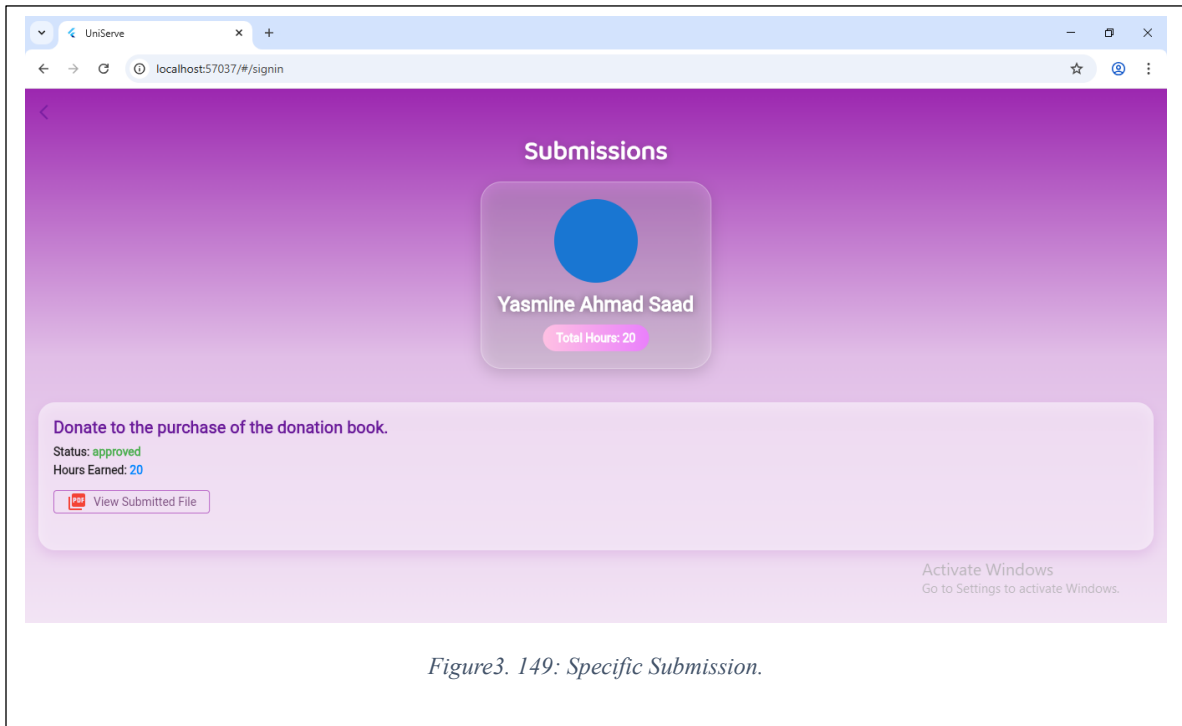


Figure3. 146: Volunteer Approvals Web.





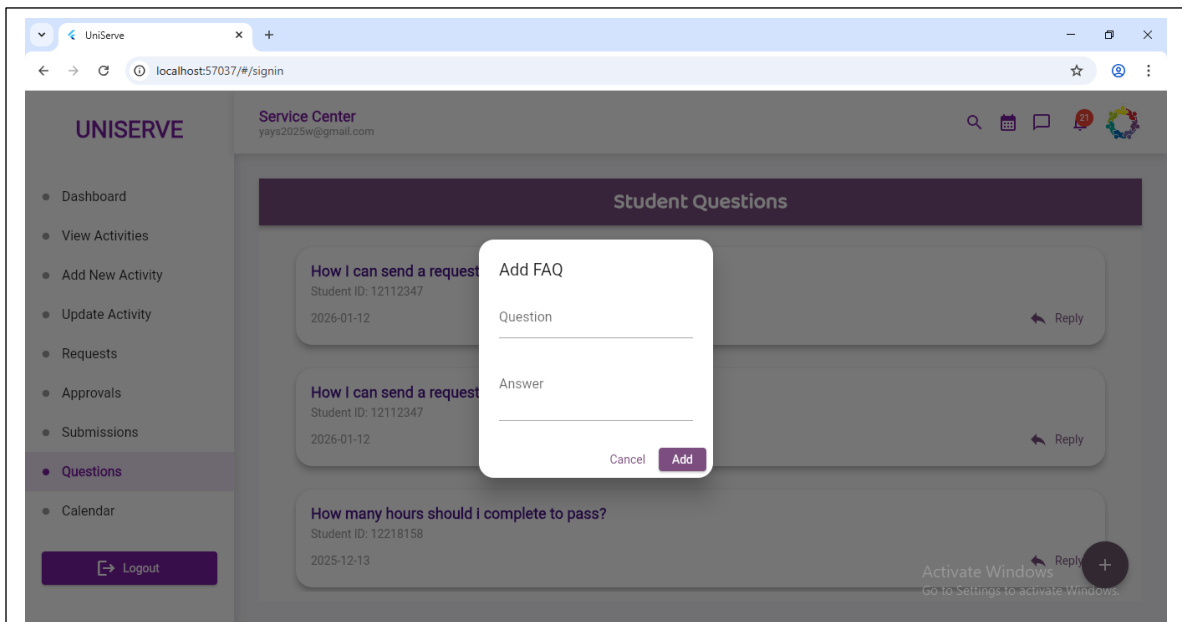


Figure3. 151: Add New FAQ.

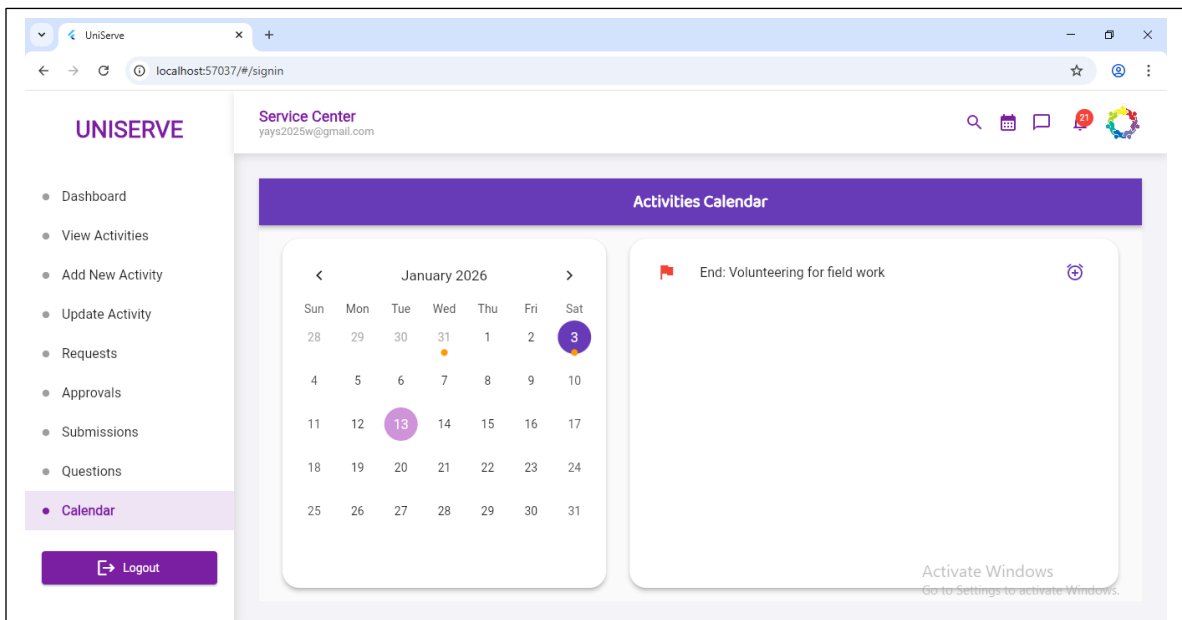


Figure3. 152: Center Calendar Web.

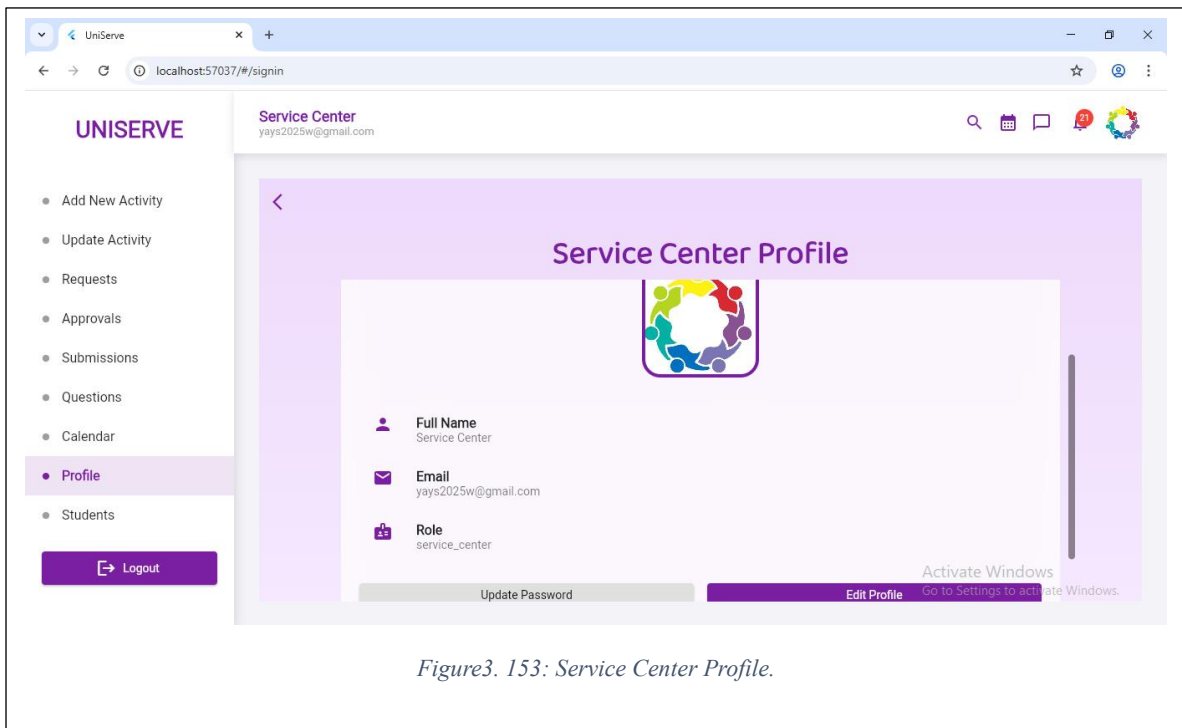


Figure3. 153: Service Center Profile.

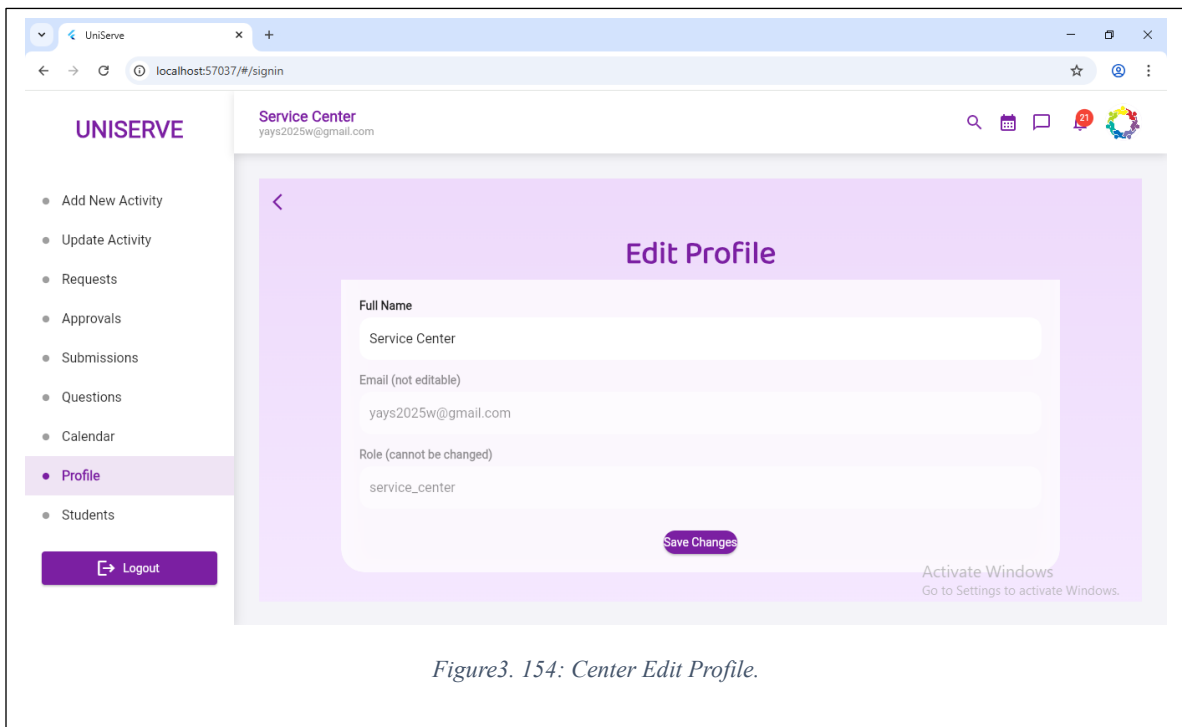
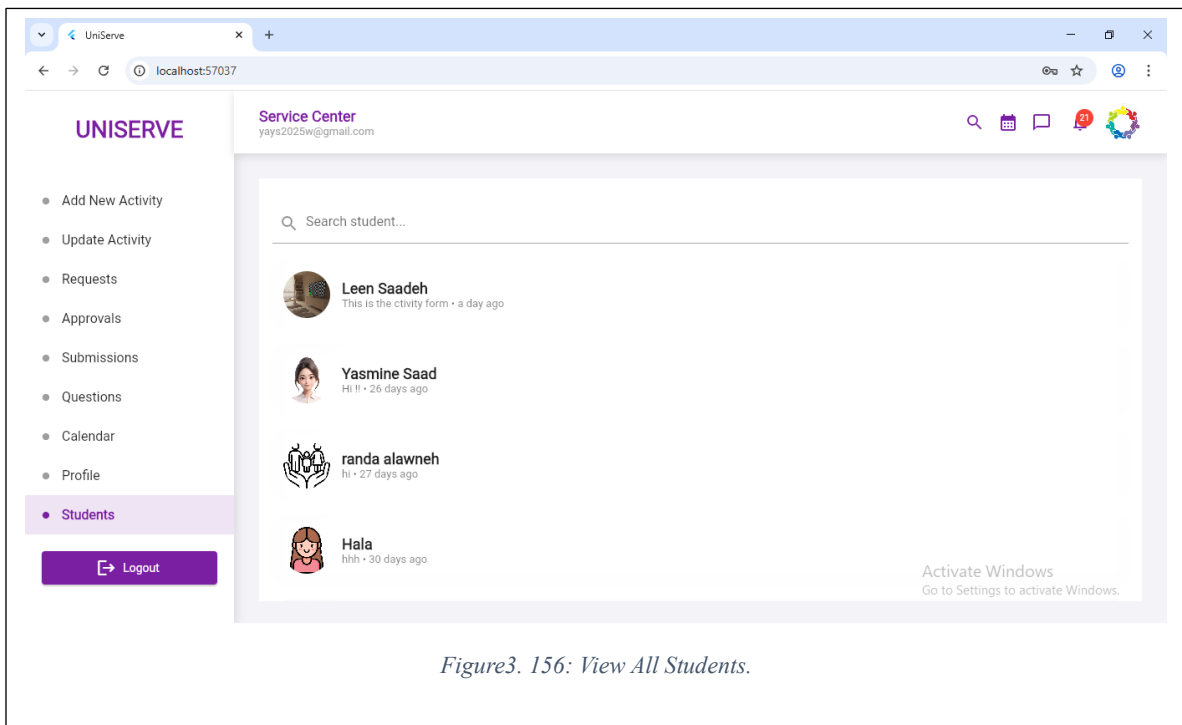
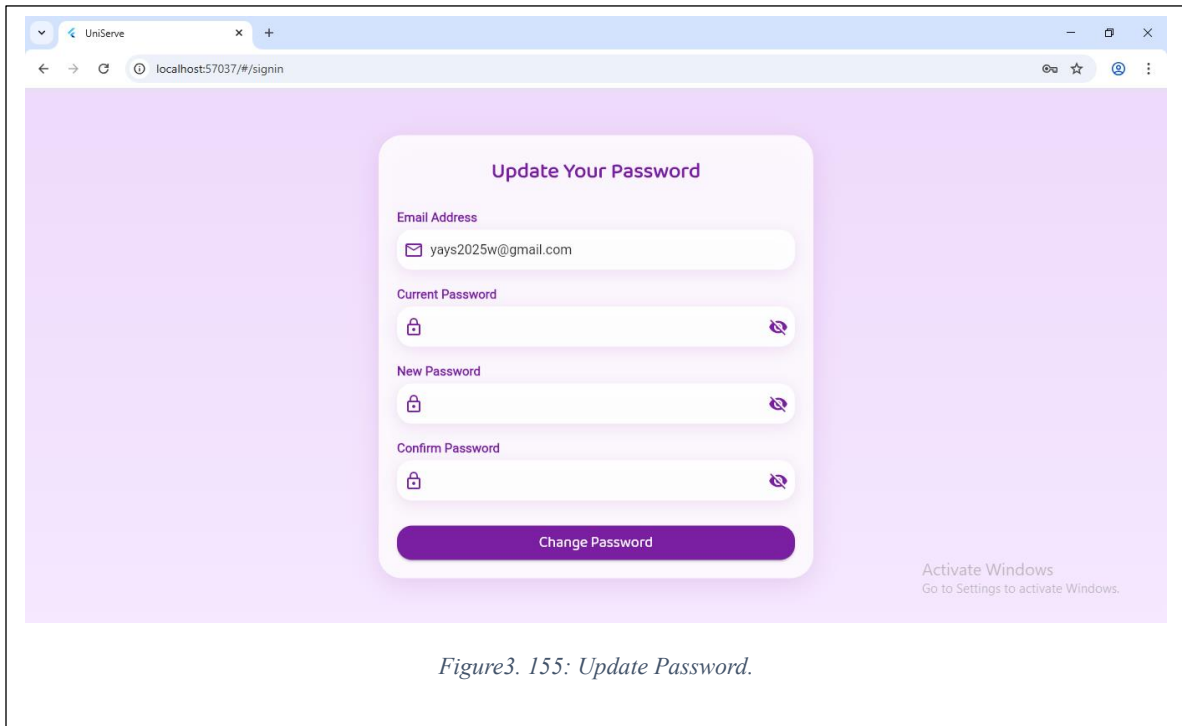
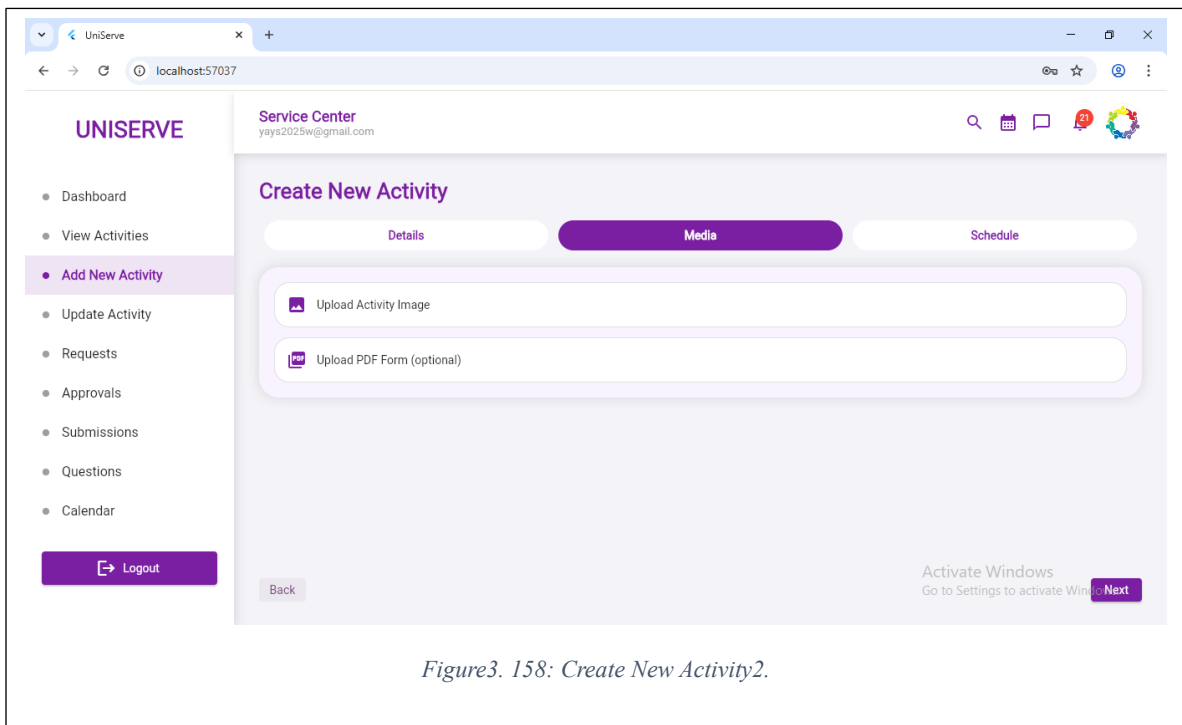
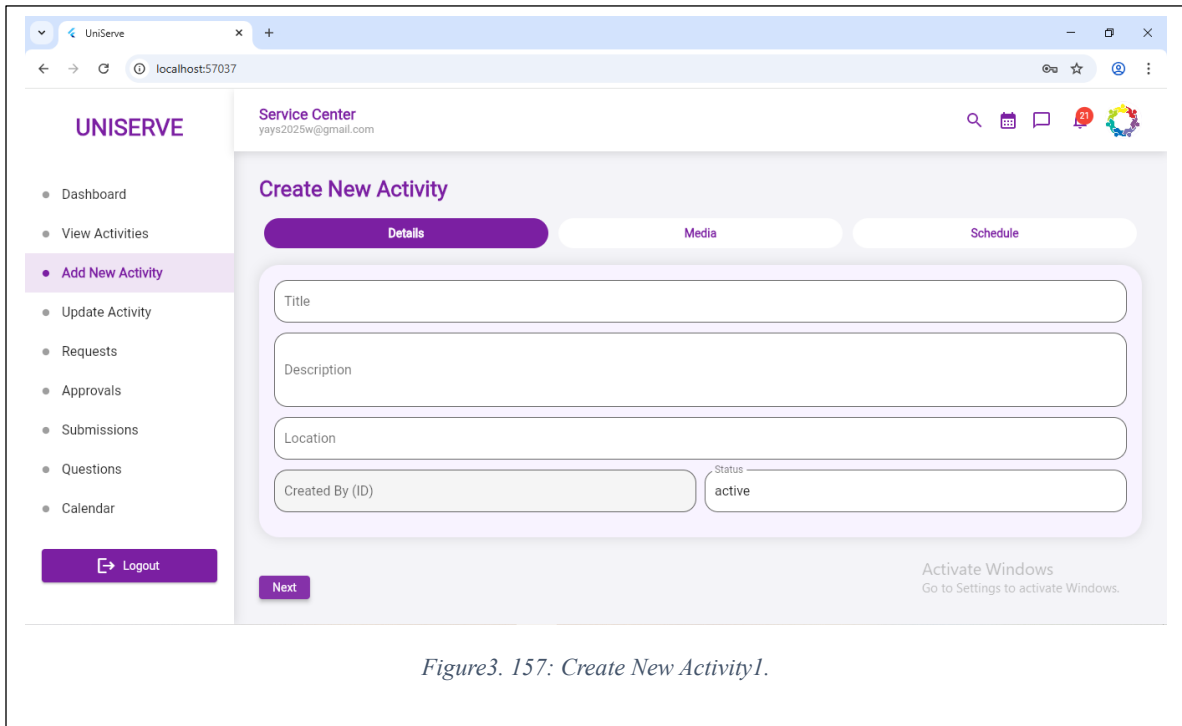
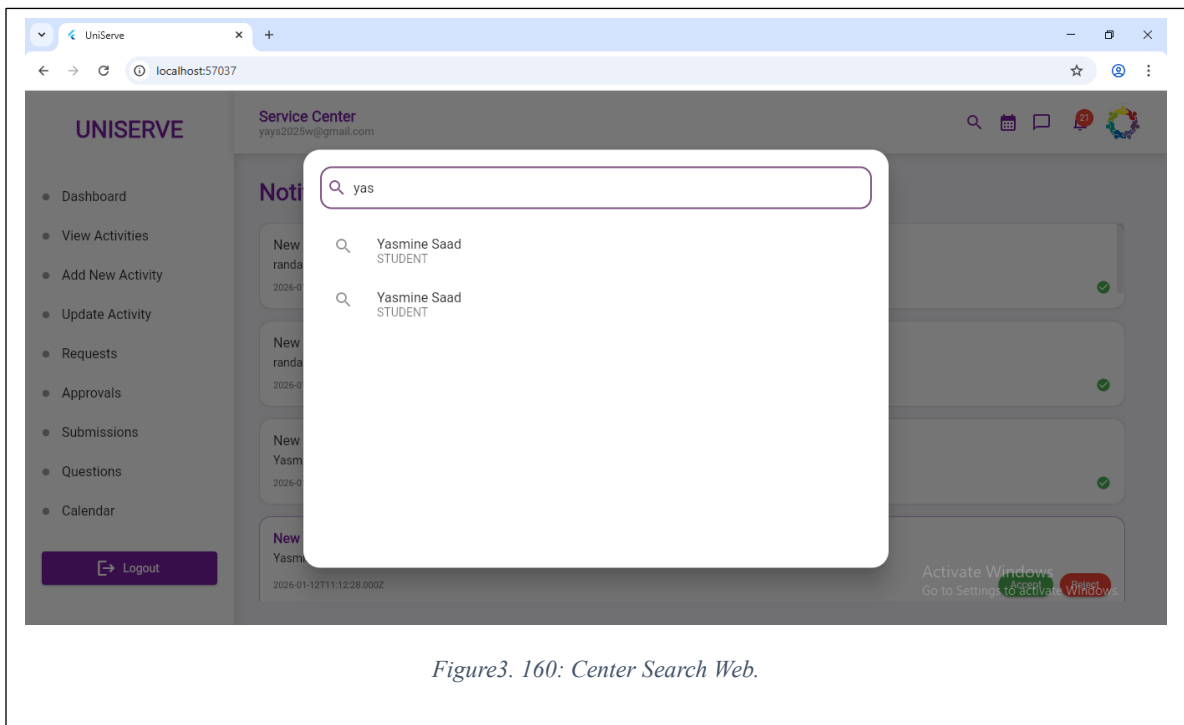
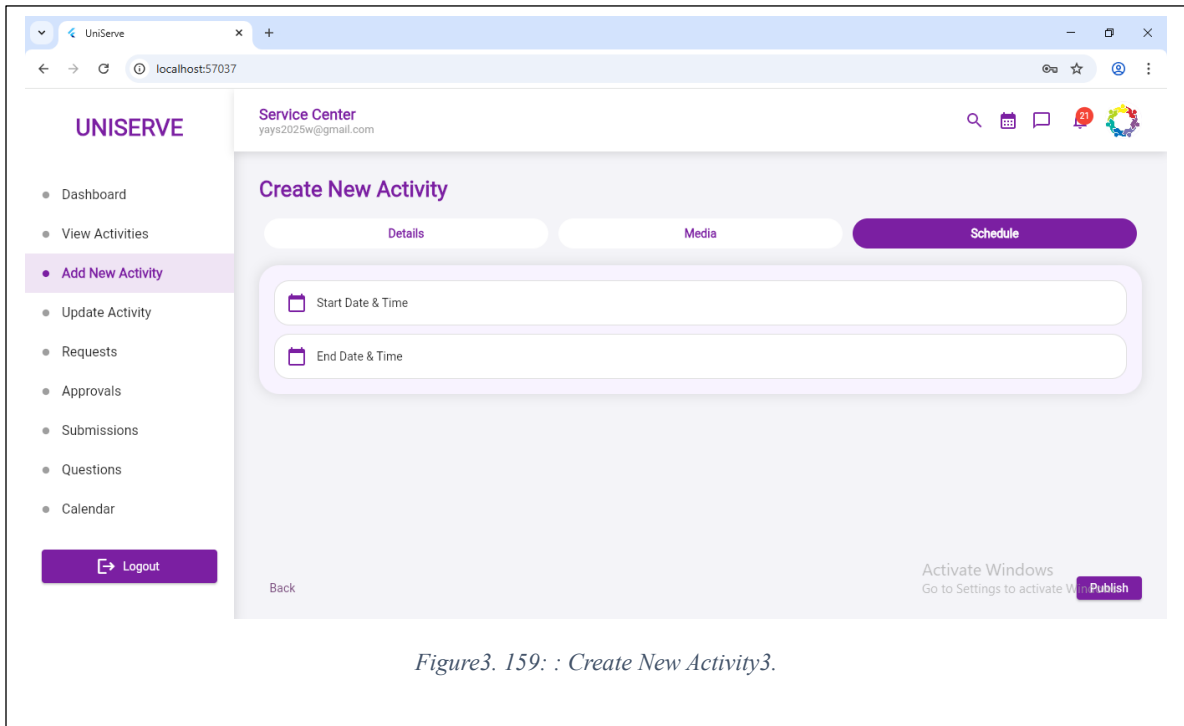


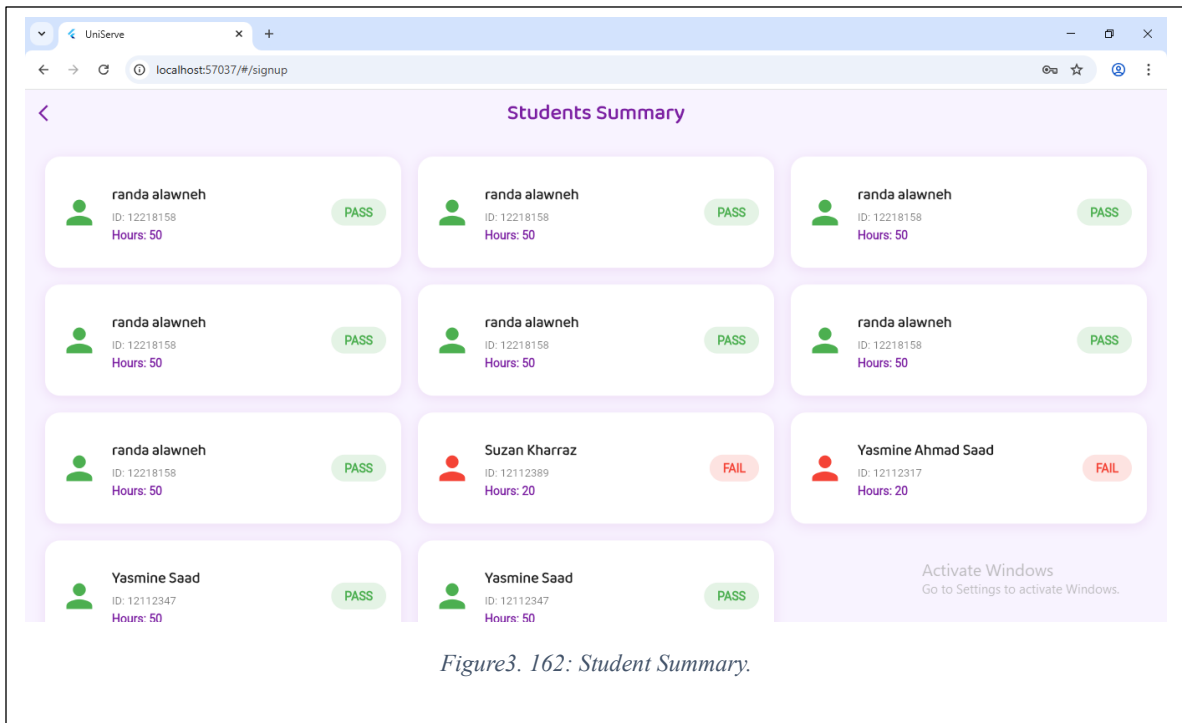
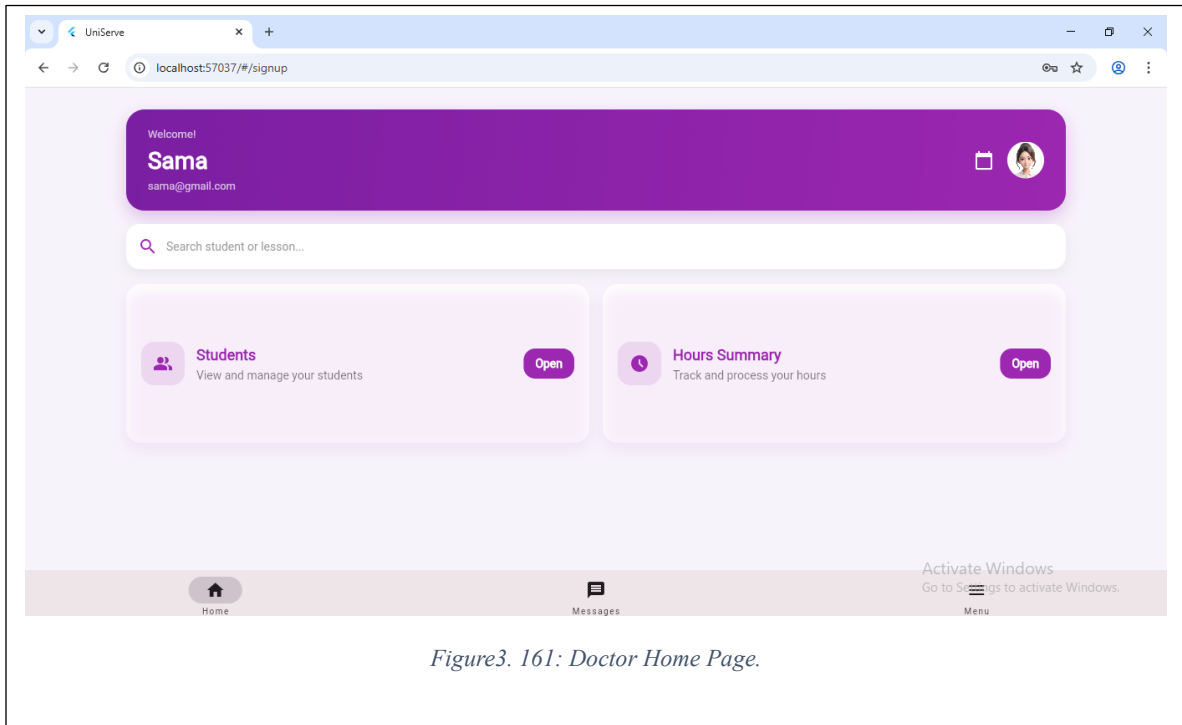
Figure3. 154: Center Edit Profile.

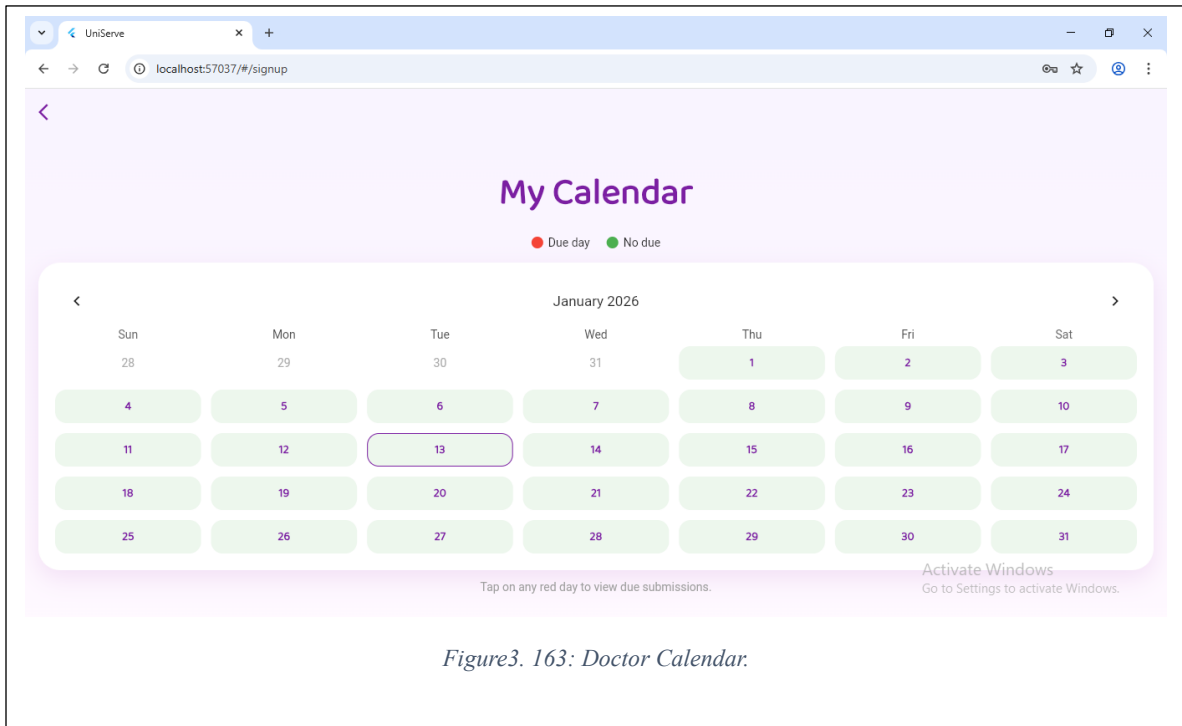




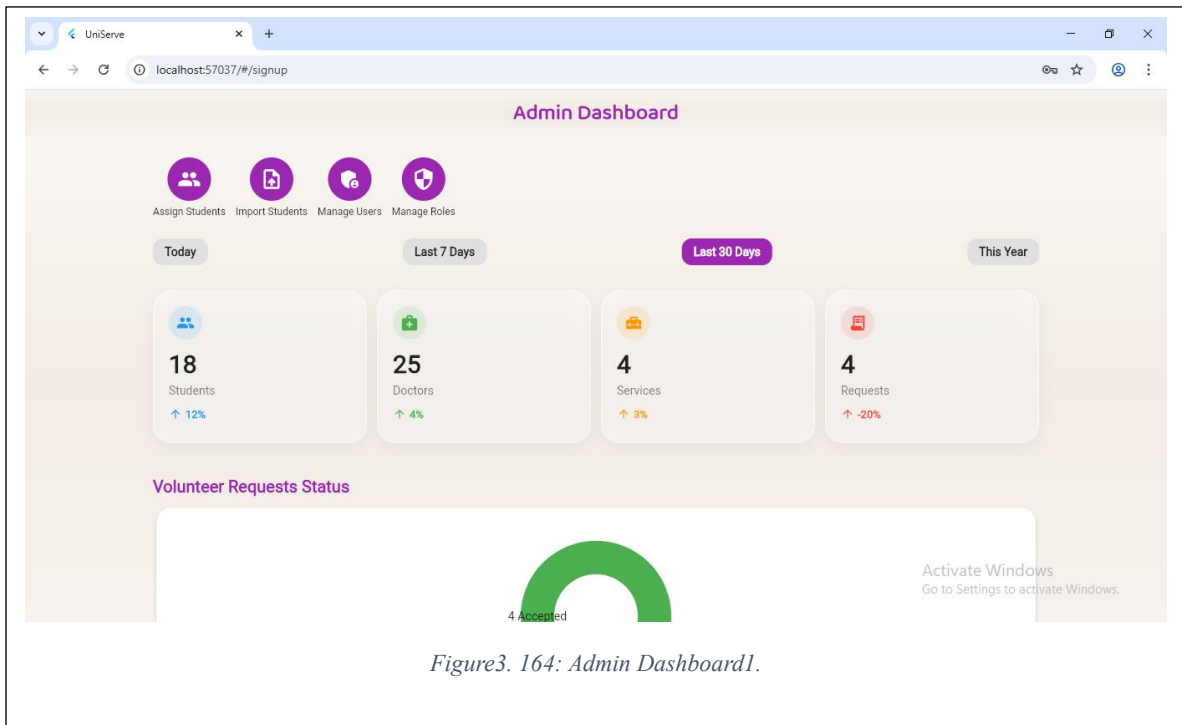


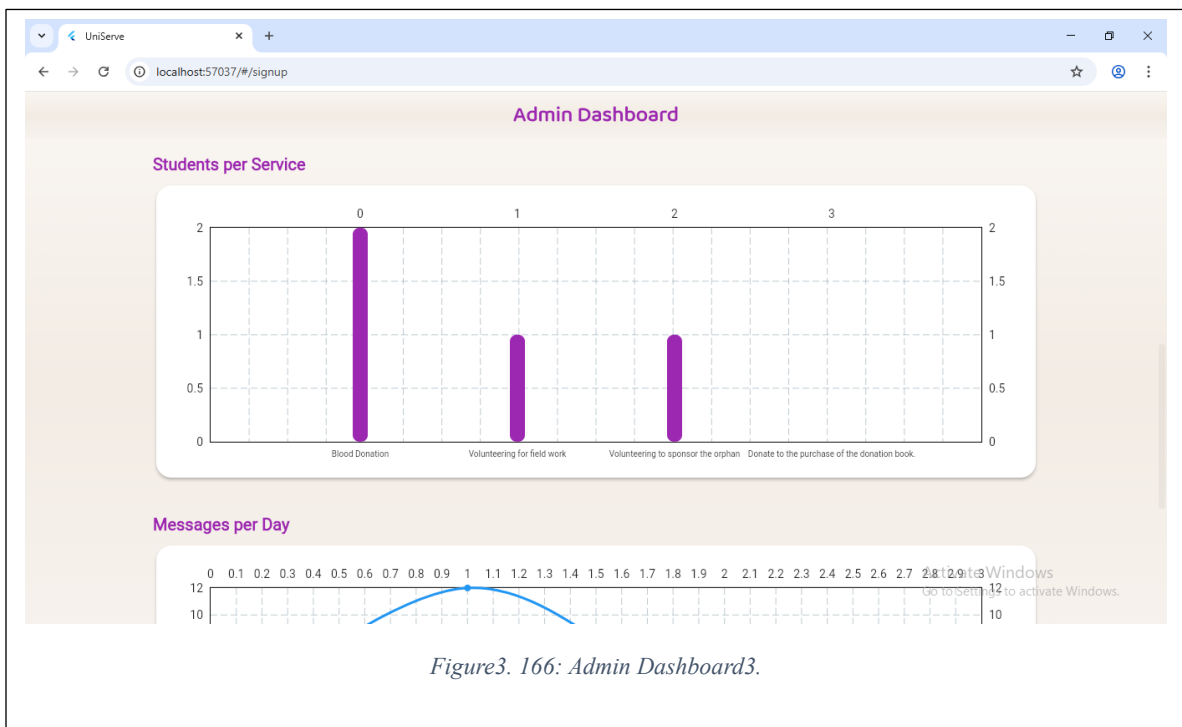
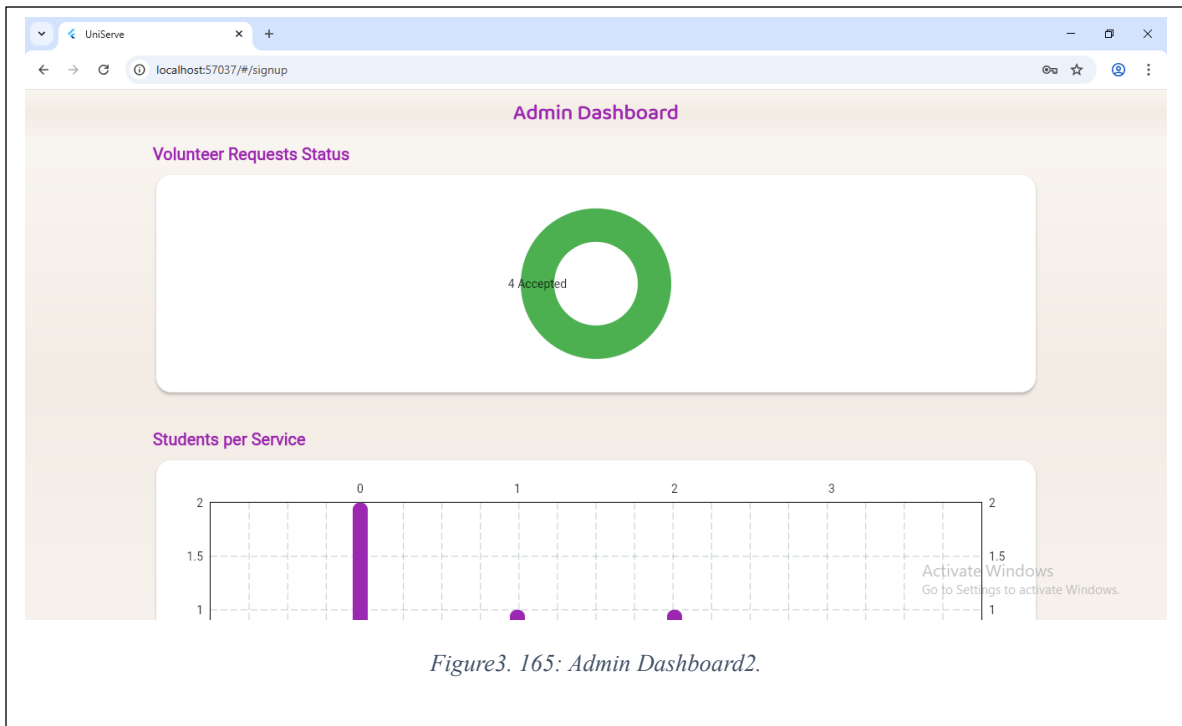
Doctor Role:

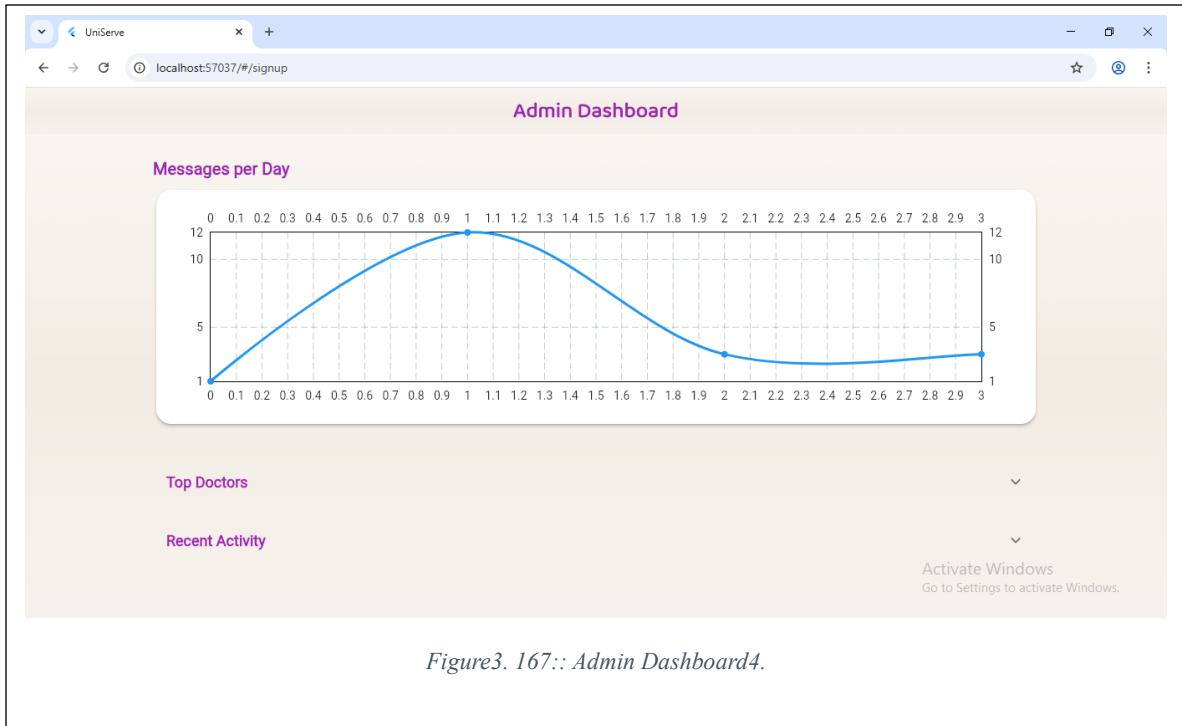


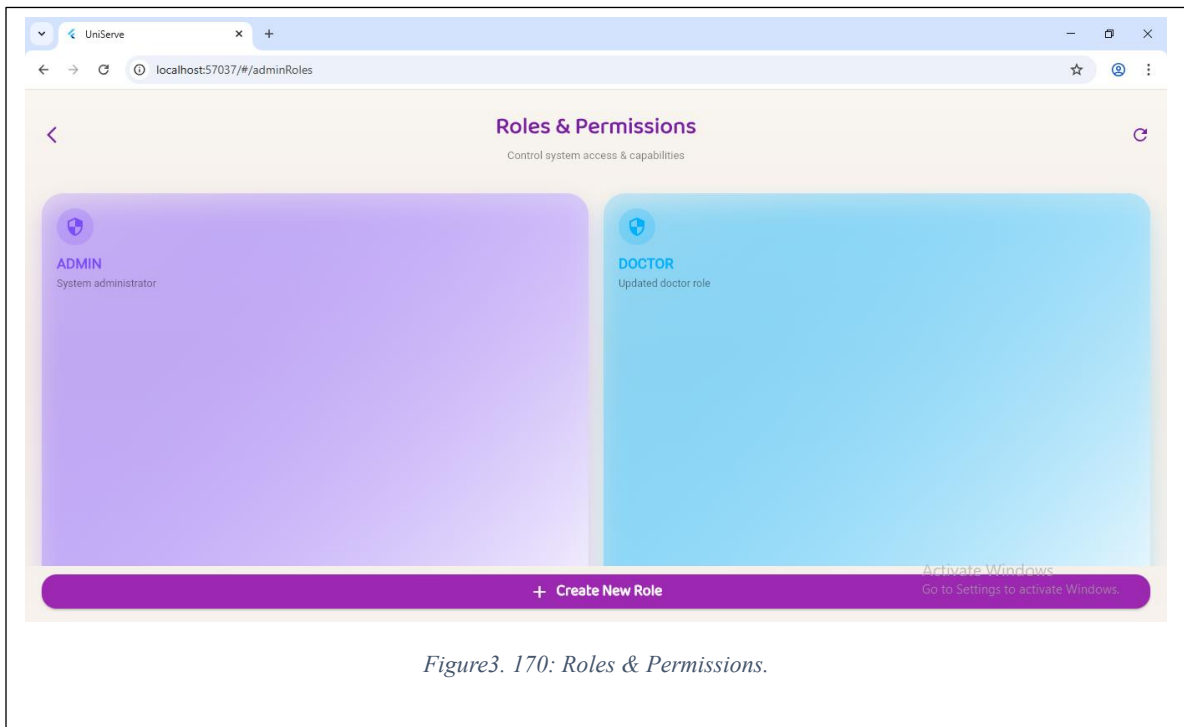
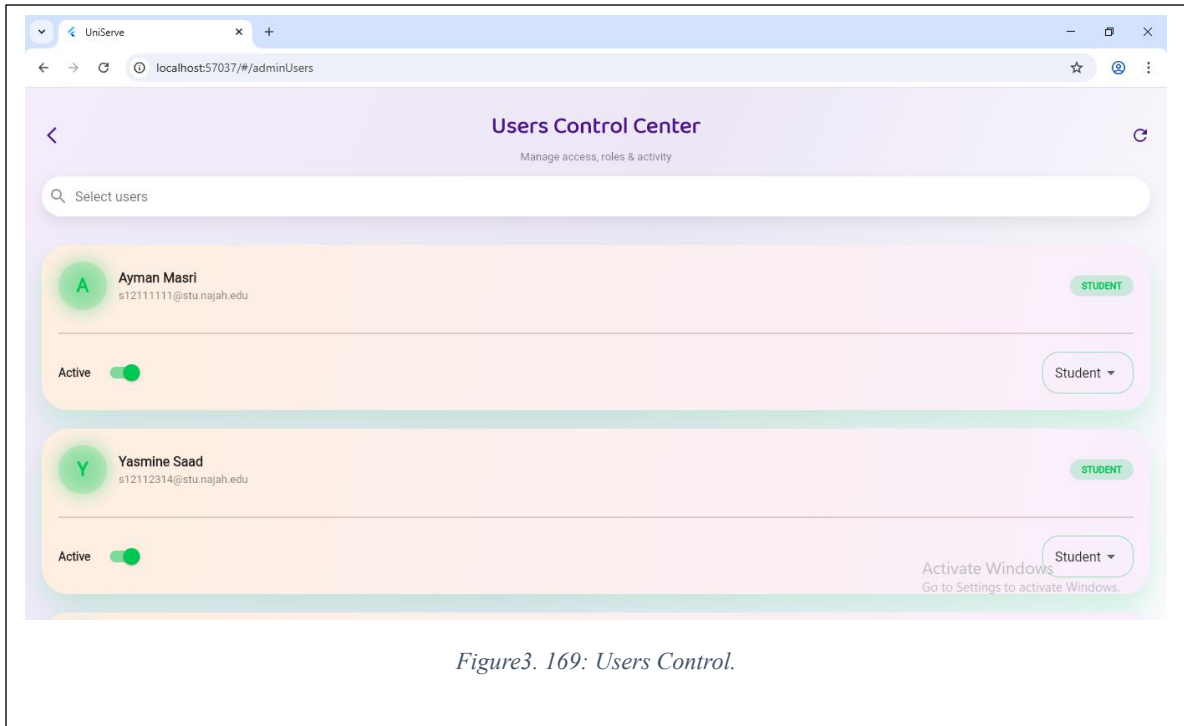


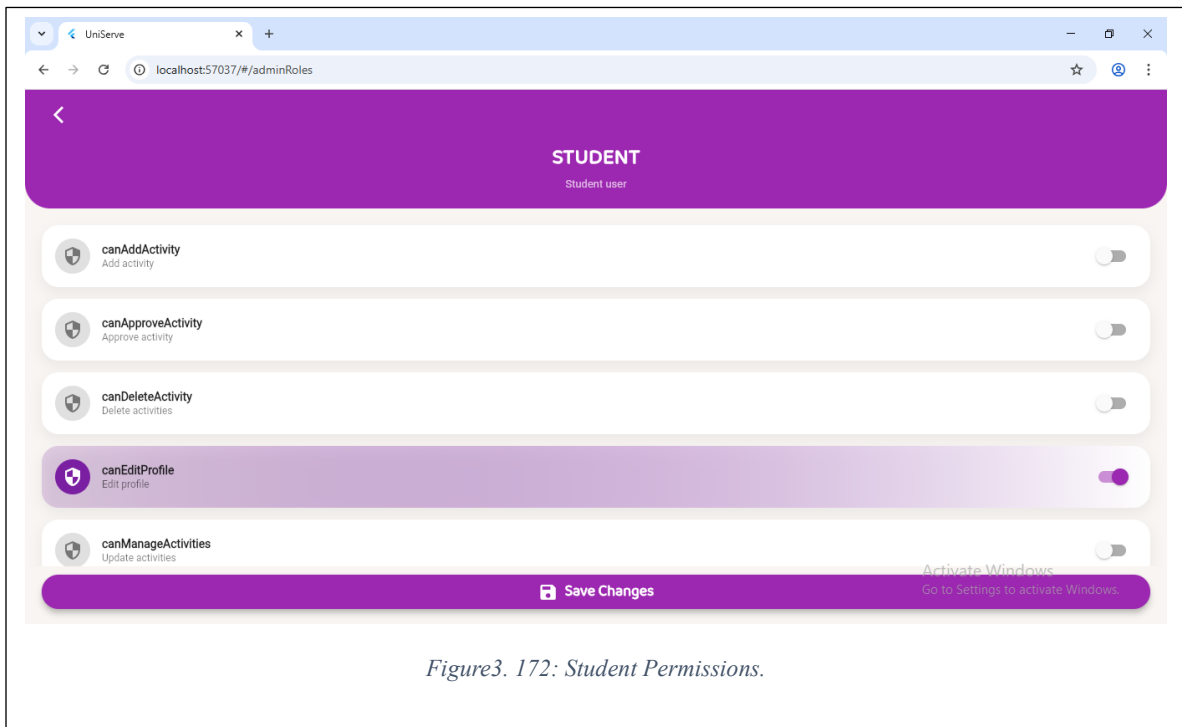
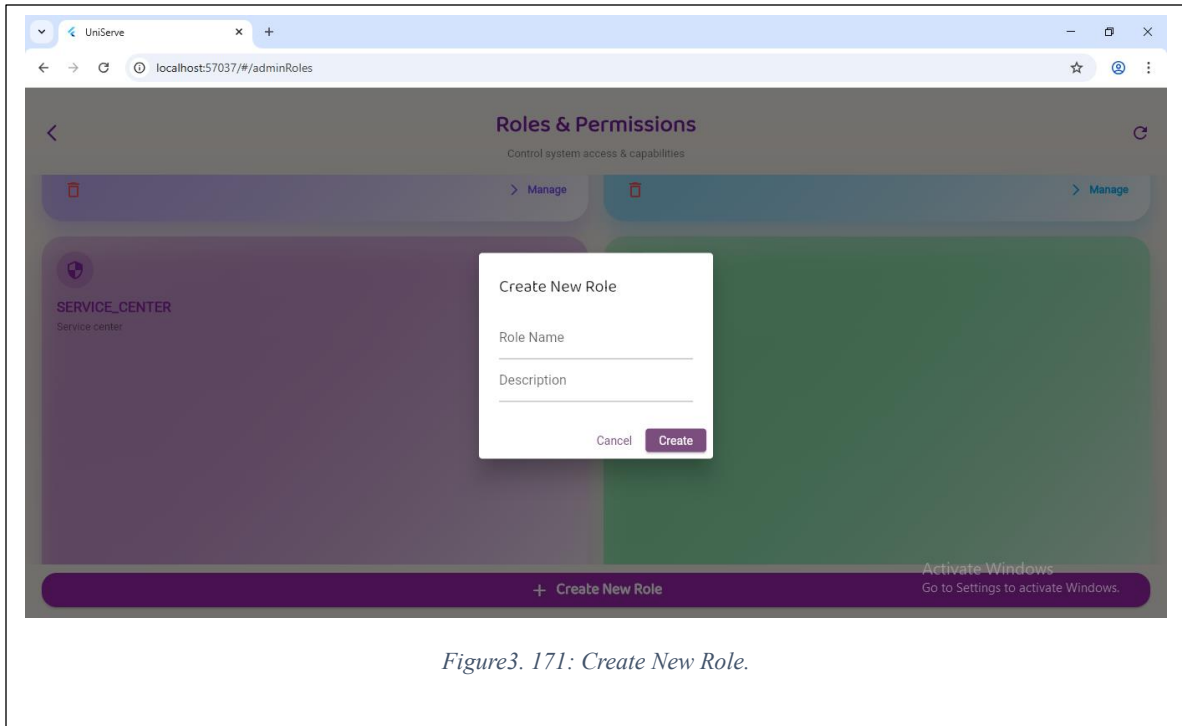
Admin Role:

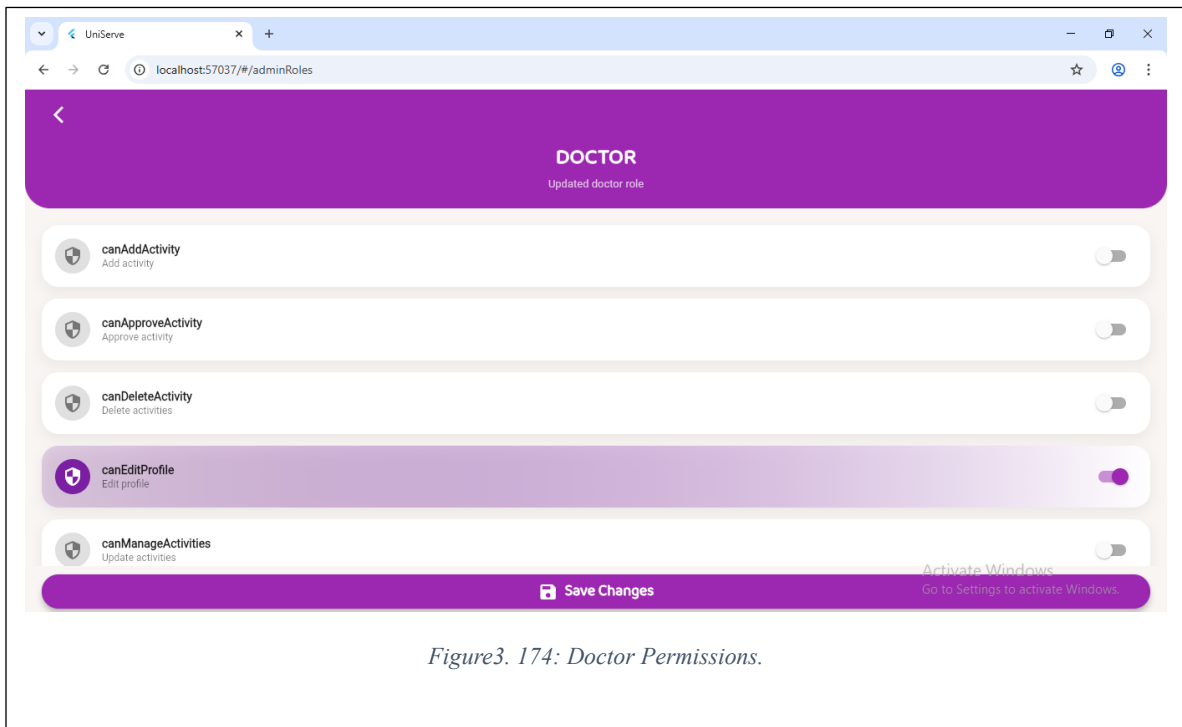
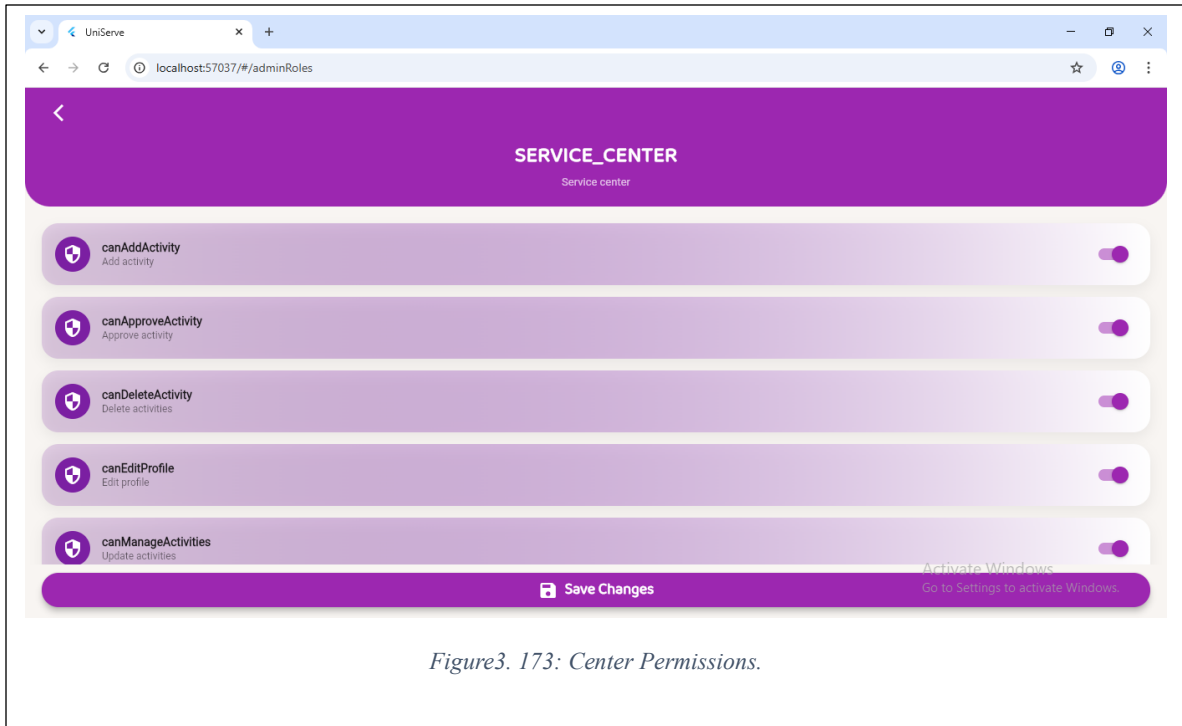












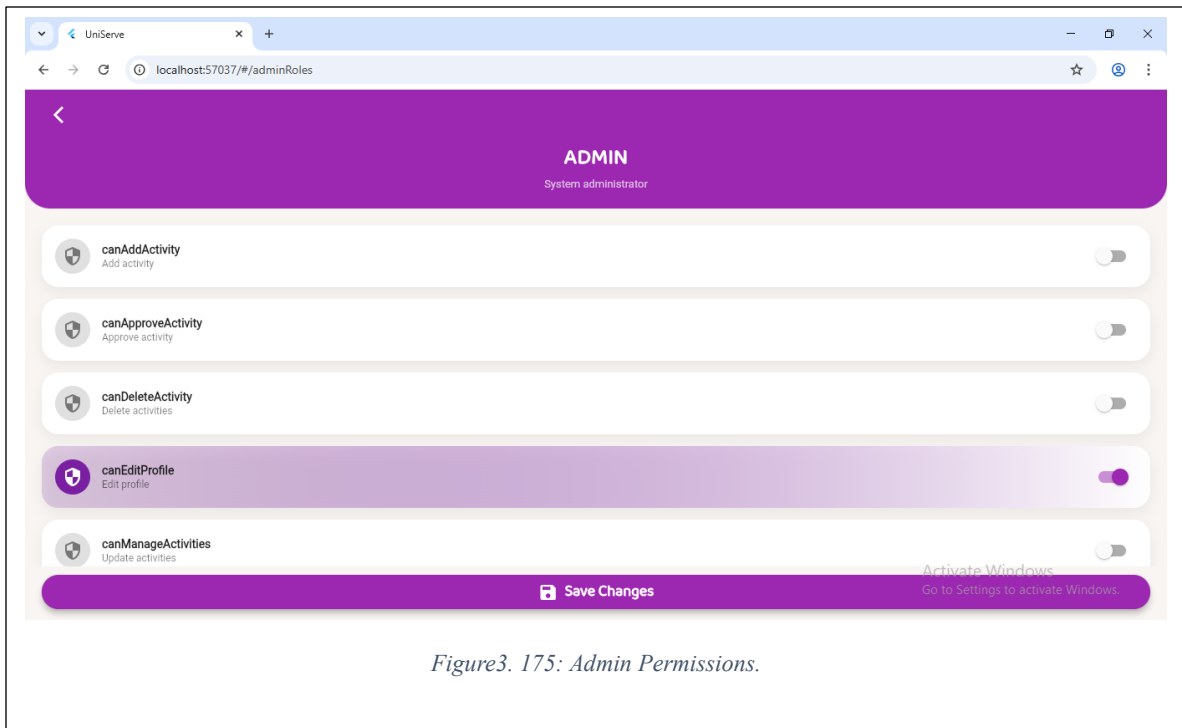


Figure3. 175: Admin Permissions.

4 Results and Analysis

This chapter presents the results obtained from the design, implementation, and testing of the UniServe system. The collected data focused on evaluating system functionality, role-based access control, user interaction, and the effectiveness of AI-assisted features. The results were analyzed to assess whether the system met its functional and non-functional requirements.

4.1 System Functionality Results:

The UniServe system was successfully implemented as a multi-platform application, operating consistently on both mobile and web environments. All core features were tested across different user roles, including **students, service centers, doctors, and administrators**. The results showed that each role was able to access only the functionalities assigned to it through the dynamic roles and permissions module.

Students were able to register, browse activities, submit volunteering requests, track progress, view their calendar, and communicate through notifications and messages. Service centers successfully managed activities, reviewed student requests, tracked submissions, and communicated with both students and doctors. Doctors were able to view their assigned students and review system-generated summaries of student progress. Administrators successfully managed users, roles, permissions, and assignments between students and doctors.

The system operated without functional discrepancies between the mobile and web versions, confirming full feature parity across platforms.

4.2 Progress Tracking and Evaluation Results:

Student progress was calculated automatically based on approved volunteering activities and submitted forms verified by service centers. The system accurately computed total completed hours and remaining required hours. Based on predefined thresholds, students were automatically classified as Pass or Fail by the system without manual intervention from doctors.

The results indicated that automated evaluation reduced human error and ensured consistent grading criteria. Doctors received final summaries rather than raw data, improving clarity and decision transparency.

4.3 AI-Based Recommendation Results (Student Side):

The AI recommendation module analyzed student interests and preferences to suggest suitable volunteering activities. During testing, students were presented with personalized activity recommendations that aligned with their selected interests and previous engagement patterns.

The results showed that recommended activities were relevant and context-aware, improving the user experience and reducing the time required for students to find suitable volunteering opportunities.

4.4 AI-Supported Request Analysis Results (Service Center Side):

For custom volunteering requests submitted by students, the AI component compared the new requests with existing service center activities using similarity analysis. The system generated a similarity percentage and identified the closest matching activity when applicable.

The results demonstrated that this feature effectively supported service centers in decision-making. Requests with higher similarity scores were easier to evaluate, while low-similarity cases were clearly flagged. This reduced ambiguity and improved consistency in request approvals and rejections.

4.5 Roles and Permissions Analysis:

The dynamic roles and permissions module allowed administrators and service centers to create, modify, or delete roles and assign permissions without modifying the system code. Testing confirmed that permission changes were reflected immediately across the system.

This flexibility improved system scalability and adaptability, allowing the platform to support future institutional requirements with minimal effort.

4.6 Error Estimation and System Reliability:

Minor delays were observed during AI similarity calculations, particularly when processing long textual requests; however, these delays remained within acceptable limits and did not affect overall system usability. User input validation minimized data entry errors, and automated status updates reduced inconsistencies in student records.

Overall, the system demonstrated high reliability, with consistent behavior across repeated test cases and user roles.

4.7 Summary of Results:

The results confirm that UniServe successfully fulfills its objectives of managing volunteering activities, automating progress evaluation, enabling effective communication between stakeholders, and leveraging AI to enhance decision-making. The integration of automation and AI significantly improved efficiency, accuracy, and user experience compared to traditional manual systems.

5 Discussion

This chapter evaluates the effectiveness of the UniServe system in addressing the challenges of managing community service activities. It discusses the project's contributions, limitations, and implications, and relates the results to current practices in volunteering management systems.

5.1 Problem Resolution:

The project aimed to overcome the limitations of traditional manual processes used in managing the Community Service course, including poor organization, limited progress tracking, inefficient communication, and lack of evaluation transparency.

Based on system design and testing, UniServe demonstrates that a centralized digital solution can effectively automate volunteering workflows, improve coordination among stakeholders, and enhance overall organization. Although the system has not yet been deployed in a real environment, the results indicate that the proposed approach addresses the core problem it was designed to solve.

5.2 Project Contributions:

UniServe introduces an integrated volunteering management system specifically designed for academic community service programs. A key contribution is the dynamic roles and permissions mechanism, which enables flexible access control without structural modifications. In addition, the integration of AI features, including activity recommendations and similarity-based request analysis, extends traditional systems by providing intelligent decision-support capabilities.

5.3 Implications of the Results:

The findings highlight the importance of automation in improving efficiency, consistency, and objectivity in volunteering management. Automated progress calculation and system-based evaluation reduce human error, while AI-assisted features support users in decision-making and activity selection. These results align with current trends toward intelligent, data-driven academic systems.

5.4 Limitations

The main limitation of this work is the absence of real-world deployment and user evaluation. Furthermore, the AI models were developed using limited datasets, which may affect accuracy. Some processes still rely on manual input, offering opportunities for further automation.

5.5 Future Applications:

Future work includes deploying the system in an academic environment to assess usability and performance, enhancing AI models with larger datasets, and expanding the system for use in other volunteering and service-based institutions.

6 Conclusions and Recommendation

6.1 Conclusions:

This project successfully designed and implemented an integrated volunteering management system that connects students, service centers, doctors, and administrators within a unified digital platform. The proposed system replaces traditional manual processes with an automated, transparent, and role-based workflow.

Based on design implementation and system testing, the results indicate that automating volunteering processes can significantly improve efficiency, accuracy, and organization. The system enables students to explore activities, submit volunteering requests, track progress, and receive notifications. Service centers are able to manage activities and verify student submissions, while doctors can monitor student progress through system-generated summaries. Final pass/fail decisions are determined objectively by the system according to predefined rules.

A major strength of the system lies in its dynamic roles and permissions model, which allows flexible management of access rights without requiring structural changes. In addition, the integration of AI features demonstrates the potential to enhance user experience through personalized activity recommendations and similarity-based analysis to support service center decision-making.

The successful implementation of both mobile and web versions with identical functionality confirms the system's readiness for deployment and its ability to operate consistently across platforms. However, the system has not yet been deployed or evaluated in a real operational environment.

6.2 Recommendations for Improvement:

To enhance the system before real-world deployment, the following improvements are recommended:

- Further training and refinement of AI models to improve recommendation and similarity accuracy.
- Adding analytical dashboards to support monitoring and reporting.
- Allowing users to customize notification preferences.
- Strengthening security through multi-factor authentication and enhanced audit logging.

6.3 Lessons Learned:

This project provided practical experience in full-stack development, role-based access control, real-time communication, and AI integration. It also highlighted the importance of modular and flexible system design to support future expansion and maintenance.

6.4 Future Work and Open Problems:

Future work includes deploying the system in a real academic environment, integrating it with official university systems, and adding multilingual support. Open challenges involve validating AI performance with real user data and ensuring system scalability as the number of users and activities increases

7 References:

- Flutter Documentation. (n.d.). *Flutter documentation*. Retrieved January 2026, from <https://docs.flutter.dev>
- Node.js Documentation. (n.d.). *Node.js documentation*. Retrieved January 2026, from <https://nodejs.org/en/docs>
- MySQL Documentation. (n.d.). *MySQL documentation*. Retrieved January 2026, from <https://dev.mysql.com/doc/>
- Firebase Documentation. (n.d.). *Firebase documentation*. Retrieved January 2026, from <https://firebase.google.com/docs>