إقليم كوردستان - العراق
جامعة جيهان - اربيل
كلية العلوم التطبيقية
قسم علوم الحاسوب

هەرێمی کوردستان- عێراق
زانکۆی جیهان- هەولێر
کۆلێژی زانست کارپێکراو
بەشی زانستی کۆمپیوتەر

Kurdistan Region- Iraq
Cihan University- Erbil
College of Applied Sciences
Department of Computer Science

Higher Education Through Interactive In Mobile Application: A Case Study In Cihan University

A graduation project submitted to
Department of Computer Science/College of Sciences
In Partial Fulfillment of the Requirements
For the Degree of Bachelor of Science

in Computer Science

By
Hussain Ayad
Sara Tahsin
Aya Abdulsalam

Supervised by:

Asst. Lect. Alyaa Asaad

(2024-2025)

" بِسْم اللَّه الرَّحْمن الرَّحِيم"

﴿دَعْوَاهُمْ فِيهَا سُبْحَانَكَ اللَّهُمَّ وَتَحِيَّتُهُمْ فِيهَا سَلَامٌ ۚ وَآخِرُ دَعْوَاهُمْ أَنِ الْحَمْدُ لِلَّهِ رَبِّ الْعَالَمِينَ﴾

"صَدَقَ اللَّهُ العَظِيمَ "

"الحمد لله الذي ما تم جهد ولا ختم سعي الا بفضله الحمد لله حبآ وشكرا وامتنان على البدء والختام لم تكن الرحلة قصيره ولا الطريق محفوفا بالتسهيلات لكن شاء الله ان تفعلها

ف الحمد لله الذي يسر البدايات وبلغنا النهايات بفضله وكرمه"

Examination Committee Certificate

Examination Committee

Head of Committee

Signature:

Name:

Date:

Member of Committee

Signature:

Name: Date:

Supervisor

Signature:

Name:

Date:

Head of Department

Signature:

Name: Date:

Acknowledgment

We sincerely express our gratitude to our supervisor, Asst. Lect. Alyaa Asaad, for her unwavering support and invaluable guidance throughout every stage of this project. Her expertise and insightful feedback have been instrumental in shaping our ideas and enhancing the quality of our work.

We are also deeply thankful to our families and friends for their constant encouragement and support in countless ways. Whether by listening to our ideas, offering constructive advice, or providing motivation during challenging moments, their presence has been a pillar of strength in this journey.

Our appreciation extends to the dedicated faculty and staff of the Computer Science Department at Cihan University - Erbil. Their commitment to fostering a dynamic and engaging learning environment, along with the opportunities for collaboration and access to valuable resources, has significantly enriched our academic experience.

Lastly, we are grateful to everyone who contributed to our research by sharing their knowledge and insights. Your input has been invaluable in shaping our understanding and refining our conclusions.

Thank you all for your support and contributions to this project.

Warm Regards,

Hussain , Sara & Aya

Abstract

The rapid advancement of technology has significantly transformed various sectors, including education. Among these innovations, Augmented Reality (AR) stands out as a cutting-edge tool with immense potential in university guidance and navigation. Our project leverages AR technology to create an interactive, user-friendly mobile application that simplifies campus navigation for students, staff, and visitors.

By simply pointing their smartphone cameras at university corridors or buildings, users can access real-time, augmented information—such as detailed directions, classroom locations, faculty details, and campus statistics—overlaid onto their screens. This immersive experience is particularly beneficial for newcomers and visitors, helping them effortlessly locate their desired destinations while enhancing engagement through dynamic digital guidance.

Additionally, the application integrates seamless payment functionalities, allowing students to pay tuition installments via Visa cards, FIB, or cash, with instant receipt generation.

Ultimately, this project aims to revolutionize campus navigation by merging augmented reality with practical university services, delivering a smooth, efficient, and futuristic experience for all users.

Key Words: Augmented Reality (AR), Campus Navigation, AI-Powered Chatbot Assistant, Interactive Maps, Navigation & Routing, Fee Payment System, Location-Based Services ,User Reviews.

Contents

Contents

Table of Figures

Chapter One: Introduction

1.1 Background

Campus tours are among the opportunities for colleges and universities to show their environment, facilities and achievement to the visitors or prospective students. Along with the advancement technology of mobile devices, smartphone and tablet have created new possibilities for the institutions to create mobile application which is integrated with augmented reality to engage campus visitors in new and more intuitive ways. Augmented reality can superimpose a virtual object or information into real world environments. Many universities are utilizing this technology as a mobile campus touring application to make the visitors become familiar with the campus through the self-guided devices [1].

Augmented reality (AR) enables an interactive experience with the real world where real world objects are enhanced with computer-generated perceptual information. Twenty-five years have passed since the first AR application designed exclusively to be used in educational settings. Since then, this technology has been successfully implemented to enrich educational contexts providing learning gains, motivation, enjoyment, autonomy, among other benefits. This study provides an overview of AR technology in education from its origins to the present. Consequently, based on the analysis of its evolution, the study defines three generations of AR in education. Moreover, the study identifies some major challenges from previous AR applications and, finally, it poses some insights to address these challenges to enhance the benefits of AR for education [2].

Augmented Reality (AR) technology has become increasingly popular due to its potential use in an indoor environment. AR technology enables virtual information, such as navigation instructions, to be merged into the actual environment via a mobile screen. Using an AR-based Indoor Navigation speeds uptime while also being interactive in searching for a particular building location. Every year when new semester students enroll in the university, some students will have difficulty finding a particular location on the campus. The most searched for building upon arrival at the university is the student halls of residence. While searching for it, students waste time asking others for information or looking for a nearby campus map [3].

Often pose intricate navigation challenges to both newcomers and regular visitors. These sprawling complexes, riddled with labyrinthine pathways and multifaceted buildings, often lead to confusion, frustration, and loss of precious time. Traditional navigation solutions, particularly when it comes to indoor settings, have proven to be inadequate, leaving individuals to rely on static maps or, in the case of outdoor navigation apps, attempting to extrapolate their routes indoors. In this context, we present an innovative solution, an augmented reality (AR) mobile application [4].

AR technology is used to replace the traditional map to provide an immersive virtual navigation direction [5].

that is accessible via mobile devices that uses AR to overlay the information as images of the searched location on campus, especially for indoor navigations for lecture halls, tutorial rooms, laboratories, and offices which rarely are covered by the normal maps. Based on the respondents' evaluation, [6].

1.2 Augmented Reality

Augmented reality, in computer programming, a process of combining or “augmenting” video or photographic displays by overlaying the images with useful computer-generated data. The earliest applications of augmented reality were almost certainly the “heads-up-displays” (HUDs) used in military airplanes and tanks, in which instrument panel-type information is projected onto the same cockpit canopy or viewfinder through which a crew member sees the external surroundings. Faster computer processors have made it feasible to combine such data displays with real-time video. Among the earliest and most prominent examples of this type of augmented reality, as first shown on the Fox Broadcasting Company’s network in the mid-1990s, were the yellow first-down stripes superimposed on television images of American gridiron football fields and the virtual flight paths added to help television viewers track the paths of hockey pucks and golf balls.

Augmented reality is commonly used in electronic first-person shooter games to add environmental, health, and other information to players’ viewpoints. (Various militaries have begun to experiment with adding similar overlays to real soldiers using personal head-mounted visors.) Augmented reality applications also have been developed for smartphones to display information such as building addresses, real estate signs, retail sales offers, and restaurant reviews on specific sites seen through the devices’ viewfinder or electronic displays. Such information may be supplied using a global positioning system (GPS) linked to a commercial or open-source database. Some visionaries hope to take the next step of adding such informational displays to lightweight sunglasses, and proponents of social networking envision all sorts of personal identification tags being added to such displays.

1.3 Problem statement

The Augmented reality technology may solve the problems of students wasting time in finding their classes or exam halls on time. This is also a waste of time and not entering the hall because of that. It may be useful for new incoming students who do not know the location of a building or which floor or class number. This may pose a problem in being late for classes or attending activities on time. It may also be useful for new visitors who do not know the directions of each building and its number and rely on asking about the directions. This may not help sometimes. By using your smartphone only, you can open the phone guidance application on a university corridor or department and it will give you instructions and suggestions for adding virtual elements in the university environment.

1.4 Project objective

The aim of this project is to create an integrated application with an all-in-one feature From several integrated aspects that serve the student and provide him with all facilitation and mobility within the university campus and buildings using modern augmented reality navigated with QR Code to facilitate mobility and fast payment process for the student using visa card then provide and save the students complete information within the application and update the registrations and save it with the actuals save it with the actual update as firebase.

1.5 Project Significance

This Project importance is limited to students, colleges, and universities project

- Innovation: it replaces the use of traditional and well-known maps, which may be unclear or inaccurate, by identifying locations and using Augmented reality navigation integrated with Google Maps.
- Communication Support: Through real-time updates, updates to activities and events, the latest news, participation, conferences, and festivals held within the university, and festivals held within the university, and updates to monthly holidays
- Facilities: using an in-app installment payment system via Visa Card instead of traditional payment methods.
- Multilingual support: this makes it easy to communicate with and use by all students of all backgrounds,
- Response Speed: it is mentioned that it responds quickly with a QR Code Scan, and information about each building appears when the code is scanned.

It will also improve the image of the university as advanced by using augmented reality. This will attract the largest number of students, which will provide the opportunity to join our university.

1.6 Scope of study

This study focuses on the design and development of the use of an application based on the use of augmented reality navigation integrated with an artificial intelligence-powered chatbot for students, faculty, and new visitors to move comfortably on campus This application provides instructions and information in real time and information about buildings and class details and these instructions enhance the accessibility of the campus

The technical scope includes the use of Flutter and Dart for the front end, which ensures easy and convenient use for the user as well as the use of Firebase and Fire Store to manage the background professionally and allow add-on updates, deletions, Create and develop an application using Flutter and dart

- Get continuous updates, the latest news, exclusive news, and holidays using fire base.
- Maps system: provides an integrated map navigation system so you can search, filter, and test the building and location you want to go to within the university
- Integrated installment payment system: enable the application to pay fees and enable secure online payment via visa card, FIB, or Fast Pay
- Multilanguage: supported in Arabic, English, Sorani
- Information Registration System: complete information for the application including personal information for the application to confirm the payment process and obtain a receipt. the information is updated by the administrator within the firebase in real time
- Integrated with AI: use a specializes chatbot to answer university questions and any other related topics. providing specific and accurate answers when used

This application is Target The students of Cihan university and new visitors and first-year students it provide them with easy navigation and discovery of the university campus through an enjoyable experience using augmented realty navigation technology supported by Google Maps as well as Unity to activate the leading arrow and the QR scanner to Scan the building code and provide information.

1.7 Research Questions:

In this research, we focus on creating an application suitable for students and newcomers, an application with new and developed technology. The aim of the questions is to understand the user's preferences and also whether AR technology serves him in terms of ease of moving and using the application with complete ease.

1. how did you find my application is it easy to use and have facilities?
2. How satisfied are you with the overall performance of this app, did you find it smooth and easy to use, and each link provides you with what you need?
3. How did you find the app interface visually appealing and user-friendly?
4. how did you find the experience of using the method of paying fees through our application? Was it easy? Did you face difficulty in paying via Visa card?
5. Have you found the use of Google Maps integrated augmented reality navigation helpful?

Chapter Two: Literature Review

2.1 Overview

The article discusses the use of SLAM (simultaneous localization and mapping) technology, with the help of which it is possible to build Indoor navigation systems using augmented reality technology, including on mobile platforms [7].

Augmented Reality (AR) campus navigation system based upon AR Core. This system leverages AR Core to present 3D information in real-world scenes, enhancing the user experience. Additionally, we incorporate indoor ranging algorithms for real-time positioning and map generation on mobile devices, specifically tailored for indoor navigation. The development is carried out using Unity3D scripts to enhance usability and adaptability [8].

the author identified an opportunity to implement Augmented Reality (AR) technology as a navigation aid at Universities Klabat, utilizing iOS as the chosen platform. In this application, users can search for various locations, such as buildings, facilities, or faculty residences [9].

"AR@Campus," an AR app designed for campus navigation, aligned with Industry 4.0 principles. It leverages AR to enhance human perception and dynamic digital interaction. The primary goal is to provide an intuitive campus navigation experience using Indoor Atlas positioning technology [10].

Every year when new semester students enroll in the university, some students will have difficulty finding a particular location on the campus. The most searched for building upon arrival at the university is the student halls of residence. While searching for it, students waste time asking others for information or looking for a nearby campus map. Therefore, this project investigates the requirements needed for an AR-based indoor navigation application to be applied within the student halls of residence and identifies technical issues through a small-scale prototype development within a small navigational area [11].

The main goal of the AR development is to help users to get information and direction easily using AR based mobile application when walking inside the library. In normal circumstances, users typically walk around and explore the library area before reaching their targeted destinations. Depending on the library size and number of reading corners, exploration and walking in the library can be time consuming. Therefore, an AR technology is introduced in this paper to improve the user experience inside the library in the right direction and information instantly [12].

2.2 Related work

Smart campus solutions leveraging AR, AI, and mobile-first design have advanced rapidly, with global institutions experimenting across navigation, feedback, and student engagement domains. Below are notable systems and studies that inform the development of our University Assistant App:

1. AR-Based Campus Navigation Systems

CampusAR (MIT) employs ARCore to guide users between lecture halls using directional arrows and voice prompts. While it improves orientation for new students, it lacks integration with real-time event data or indoor navigation.[9]

NAVGATE (South Korea) utilizes Unity and image-based localization to support campus tours with dynamic overlays. Its use of 3D avatars for landmarks enhances engagement but consumes significant mobile processing power.[10]

SmartRoute (India) integrates QR code scanning with Firebase to fetch room coordinates and render virtual arrows in AR. Though lightweight, its reliance on static path data limits scalability for multi-floor buildings.[11]

2. AI-Driven Student Chatbots

UniBot (Canada) supports academic queries and event updates via voice and text. Built on Dialogflow, it improves student satisfaction by 23% but lacks offline fallback and contextual continuity for multi-turn interactions.

EduMate (UK) provides a personalized assistant powered by GPT-3, capable of advising on courses and schedules. It integrates NLP with calendar APIs but shows delays in fetching campus-specific data due to its cloud-only architecture.

Amico (Italy) combines AI with Firebase to offer multilingual support and academic deadlines via push notifications. However, its recommendation system lacks adaptive learning based on user history.

3. Mobile Apps for Campus Engagement

MyCampus App (Australia) offers a modular design for dining, events, and map browsing. While visually cohesive, it does not support AR or real-time feedback integration.

KU Mobile (University of Kansas) uses Flutter to deliver a unified interface for grades, maps, and announcements. Its theming engine adjusts based on user roles (student, staff), but its chatbot is rule-based and limited in scope.

4. Hybrid Systems with Feedback and Admin Dashboards

eCampusHub (Germany) includes a feedback module with screenshot support and admin dashboards powered by Firebase Analytics. Its robust moderation tools improve response time but depend heavily on manual triage.

SmartU (Singapore) integrates payments, AR navigation, and chat assistance into one app. Its use of ensemble models for personalized suggestions is promising, though explainability and real-time route adjustments remain underdeveloped.

Chapter Three: Methodology

3.1 System Architecture and Design

The provided system structure design outlines the architecture of a Ultimate University Application, emphasizing its modular design and the interactions between its components. Below is a detailed breakdown of the application's structure and functionality:

Figure 3.1 DML Diagram Representing the structure and functions of Cihan University Mobile Applications

Key Modules and Their Subcomponents

Authentication Module

- **Sign In:** Allows existing students or administrators to securely log into the app using email and password.
- **Sign Up:** Enables new users to register, creating a profile linked to their university credentials.

Admin Module

- **Admin Dashboard:** Centralized interface for administrators to monitor and manage app content such as feedback, buildings, and announcements.
- **Building Manager:** Allows admins to add, edit, or remove campus buildings and their metadata (images, categories, coordinates).
- **Feedback Management:** Enables viewing, categorizing, and responding to student feedback, including screenshots and feedback types.
- **User Management:** Provides access to user account controls, including viewing user activity or moderating misuse.

User Module

- **Home Screen:**
    - **View Content:** Displays categorized university information (e.g., dining, events, student services) with carousels and announcements.
    - **Dynamic Updates:** Content is fetched in real time from Firestore, personalized by user type or preferences.
- **Map and Navigation Screen:**
    - **Building Map:** Interactive campus map with filterable building categories and theming (light/dark mode).
    - **AR Navigation:** Guides users through campus using QR-scanned destinations and AR arrows rendered in real space.
- **Chatbot Assistant:**
    - **Send Message:** Users can ask questions via text or voice, related to campus life, navigation, or support.
    - **Receive Response:** AI-powered replies using the ChatGPT API provide real-time answers and recommendations.
    - **Suggestions & Voice Input:** Includes contextual prompts and voice-to-text capability for hands-free interaction.
- **Feedback Screen:**
    - **Submit Feedback:** Users can report bugs, share suggestions, or upload screenshots.
    - **Select Feedback Type:** Allows classification (e.g., Bug, Suggestion, Complaint) to improve admin triage.
- **Events and Offers:**
    - **Browse Events:** Displays upcoming campus events or services in a card-based layout.
    - **View Offers:** Shows student-exclusive promotions or time-sensitive campus deals (if applicable).
    - **Receive Response:** AI-powered replies using the ChatGPT API provide real-time answers and recommendations.

- **Suggestions & Voice Input:** Includes contextual prompts and voice-to-text capability for hands-free interaction.

- **Payment Page:**
  - **Pay Fees:** Enables students to securely pay tuition or service-related fees via a Flutter-based form with backend integration.

Interconnectivity Between Modules

The system architecture is designed to promote seamless interaction between all modules, ensuring a unified and user-friendly experience:

- Users authenticate is required before accessing the core features of the application. This ensures secure and personalized access for both students and administrators.
- Admins once authenticated, gain exclusive access to the Admin Module, where they can manage campus buildings, view submitted feedback, and maintain dynamic content across the app.
- The **Chatbot Module** is universally accessible after login, serving as a real-time AI assistant to help users navigate the campus, access services, or resolve queries.
- The **AR Navigation System** depends on QR code input and interacts with Firestore to fetch
- building data and 3D route paths, bridging the **Map Module** and real-world guidance.
- The **Feedback Module** captures user input and connects with the **Admin Dashboard**, enabling
- timely review and response.
- Dynamic content such as events, dining options, or promotions is shared across the
- **Home Screen**, **Map**, and **Chatbot**, ensuring consistent information flow.

Planned Development Technologies

The development stack is selected for performance, scalability, and ease of integration with AR and AI services:

- **Frontend Development:**
    - **Flutter** – Utilized for its expressive UI components, cross-platform support, and fast development cycles.
- **Programming Language:**
    - **Dart** – Used throughout the Flutter application for UI logic, state management, and data handling.
- **Backend Services:**
    - **Firebase** – Chosen for its comprehensive suite of tools including:
        - **Cloud Firestore** (database)
        - **Firebase Authentication**
        - **Firebase Storage**
        - **Cloud Functions** (optional for admin alerts and automation)
- **AR Support:**
    - **ARCore Flutter Plugin** or **flutter_unity_widget** – For implementing 3D AR navigation using device sensors and spatial anchors.
- **AI Integration:**

    - **OpenAI API** – Powers the conversational chatbot using natural language processing.
- **Development Environment:**
    - **Android Studio** – Primary IDE used for Flutter development, debugging, and UI design.

This modular design ensures scalability, maintainability, and ease of development while offering a seamless user experience. Each technology will be mapped to specific modules in subsequent development phases.

3.2 System Main Requirements:

This study investigates the software architecture and technologies utilized in a mobile application development project, with a primary focus on the integration of Dart programming language constituting 60% of the codebase, alongside Flutter framework and Firebase backend services. Through a detailed analysis of the codebase, it is found that Dart serves as the foundational element, offering simplicity, performance, and robustness, while Flutter complements it by facilitating the creation of visually appealing and responsive user interfaces across multiple platforms. The incorporation of Firebase backend services enhances data management and synchronization, contributing to the application's scalability and real-time capabilities. This research underscores the significance of thoughtful technology selection and integration in achieving desired outcomes in mobile application development, providing insights into contemporary practices in software architecture and technology utilization within the domain.

3.3 Introduction to Flutter

is an open-source UI framework developed by Google in May 2017 at the Google I/O developer conference. that allows developers to build native-quality mobile, web, and desktop applications from a single codebase. Flutter is a valuable modern tool used to create stunning cross-platform applications that render native code on each device and OS. Flutter is compatible with Android, iOS, Linux, Windows, etc. [12]

3.3.1 key feature of flutter

Its core strength lies in its ability to expedite development through a variety of features. Firstly, Flutter streamlines the process by allowing developers to write code once and deploy it seamlessly

across diverse platforms, including iOS, Android, web, and desktop environments. This is made possible through its utilization of the Dart programming language, also developed by Google. Additionally, Flutter offers a rapid development cycle with its hot reload functionality, enabling developers to instantly visualize and implement changes to the codebase without the need for restarting the application, thus significantly accelerating the iteration process. Its expressive and flexible UI toolkit empowers developers to craft visually stunning and customizable user interfaces 11 using a rich set of widgets. These widgets can be effortlessly tailored to adhere to the design language of specific platforms or brands, ensuring a consistent and polished user experience.

3.4 Strength of Using Dart Programing Language

Dart is an open-source programming language that is general-purpose, object oriented, and has C-style syntax developed by Google in 2011. Dart programming aims to build UIs for web and mobile applications. It is continuously being enhanced and supports compiling into native machine code, enabling it to create mobile apps. Dart borrows elements from Java, JavaScript, C#, among others, and can be said to be a strongly typed language. [13]

3.5 Utilizing Android Studio for Mobile Application Development:

This section describes how to create Android apps with Google's Android Studio, an integrated development environment (IDE) that offers all the tools you need to create apps quickly and effectively.

Installation and First Setup: Downloading and installing Android Studio from the official website is the first step in the procedure. When creating a new project, developers first choose an appropriate template and set up the project name, package name, and programming language (Java or Kotlin).

Development Process: To write and polish code, developers utilize Android Studio's sophisticated

code editor, which offers real-time error detection and code completion among other capabilities. The IDE makes it possible to develop user interfaces using both visual editing and XML coding.

Testing and Optimization: To test apps in a variety of scenarios, Android Studio provides an emulator, or developers can utilize a real device. Debugging and performance analysis tools are essential for maximizing the functionality of the program.

Deployment: The application is ready for release once it satisfies all requirements. Android Studio makes it easier to package and distribute apps by supporting version control systems, project management, library and SDK integration, and project integration. [14]

3.6 Integrating Firebase for Backend of Restaurant Finder Mobile Application:

This This section outlines the integration of **Firebase**, Google's cloud-based platform, as the backbone of the **University Assistant Mobile Application**. Firebase provides a powerful suite of tools that supports secure, scalable, and real-time app functionality—critical for delivering seamless campus experiences to students and administrators.

Core Firebase Services Utilized

- **Authentication:**

  Manages secure user login and registration, with support for role-based access control (e.g.,

student and admin privileges).

- **Cloud Firestore:**

  Serves as the real-time NoSQL database storing dynamic and structured data such as:

  - Campus buildings
  - Navigation paths
  - Chatbot content
  - Event listings
  - User feedback

- **Firebase Storage:**

  Handles the upload and retrieval of media files, including:

  - Feedback screenshots
  - Carousel images
  - Building photos

- **Cloud Functions:**

  Supports automated backend processes like:

  - Sending admin alerts when new feedback is submitted
  - Triggering push notifications for upcoming events or important campus updates

Key Reasons for Choosing Firebase

- **Real-Time Synchronization:**

  Ensures that updates to campus data (e.g., new buildings, event changes, or chatbot updates) are instantly reflected across all user devices, enabling a responsive and reliable user experience.

- **Scalability and Security:**

  Firebase effortlessly scales with user growth, and its integrated security rules help protect sensitive information such as authentication credentials, user roles, and submitted feedback.

- **Integrated Analytics and AI Support:**

  Allows tracking of in-app behavior (e.g., most-viewed buildings or frequent chatbot queries)

to inform future updates. Firebase can be paired with AI services (like OpenAI) for intelligent and personalized user interactions.

- **Development Efficiency and Cost-Effectiveness:**
  Reduces backend complexity through a serverless architecture. This minimizes infrastructure management overhead and accelerates feature development, making Firebase ideal for academic and startup-level projects.

Benefits of Integration:

Firebase's seamless integration with third-party APIs—such as Google Maps for building geolocation and Open AI for chatbot interaction—significantly enhances the functionality and user engagement of the University Assistant App. Features like real-time data updates, dynamic content filtering, and AI-powered assistance contribute to a highly personalized and interactive user experience. In addition, Firebase Performance Monitoring and Crashlytics help maintain app reliability by identifying and resolving issues proactively, ensuring smooth performance even during peak usage (e.g., registration periods or event launches). By offering essential backend services such as secure authentication, scalable cloud storage, and real-time database management, Firebase strengthens the app's foundation. This integration directly supports the project's goals of building a robust, user-friendly, and scalable campus assistant platform that meets the daily needs of students and administrators alike.

3.6.1. Firebase Configuration Requirements:

Firebase was integrated into the **University Assistant Mobile Application** through a series of structured configuration steps to establish a secure and scalable backend. The process began by creating a **Google account** and accessing the **Firebase Console** at firebase.google.com. A new Firebase project was initialized by selecting **"Add Project"** and assigning a valid name such as **"UniversityAssistantApp"**, following Firebase's naming conventions.To enable detailed performance tracking and usage analytics, **Google Analytics** was activated during project setup. A default Analytics account was linked, with tracking parameters customized to monitor

interactions like **building searches**, **chatbot usage**, **event views**, and **navigation actions**.For Android integration, the **Android icon** was selected from the Firebase project dashboard. The app's **package name** (e.g., com.universityassistant.app) was entered—this identifier was retrieved from the build.gradle file of the Flutter project. Optionally, a **SHA-1 debug signing certificate** was also added to enhance authentication capabilities and enable advanced features such as Google Sign-In. Next, the **google-services.json** configuration file was generated and downloaded. This file was placed in the android/app directory of the Flutter project. It was crucial to keep the filename unchanged to ensure proper linkage between Firebase and the app. To complete the integration, the application's build.gradle and app/build.gradle files were updated with the necessary **Firebase SDK dependencies** and **Google services plugin**. Key services included:

- **Firebase Authentication**
- **Cloud Firestore**
- **Firebase Storage**
- **Firebase Analytics**

This comprehensive setup allowed Firebase to function as the core backend infrastructure, providing real-time data syncing, secure user management, and performance monitoring essential for maintaining a responsive and feature-rich university assistant platform

below:

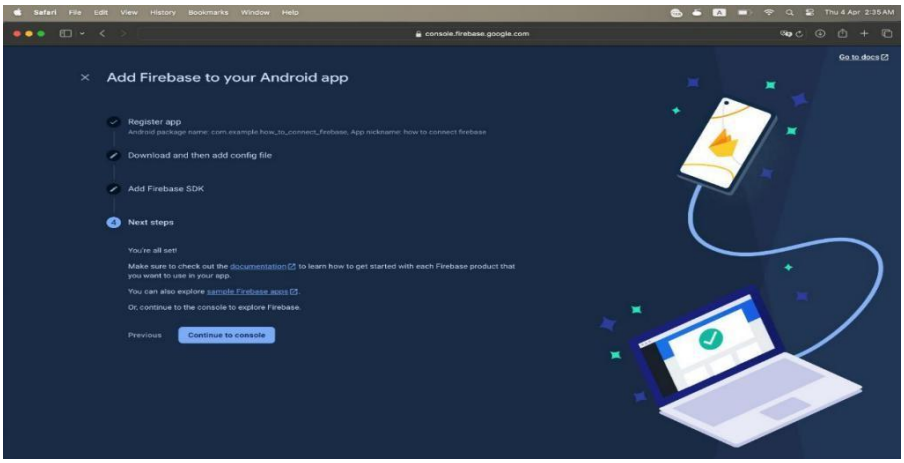


Figure 3.2, Project dashboard on firebase

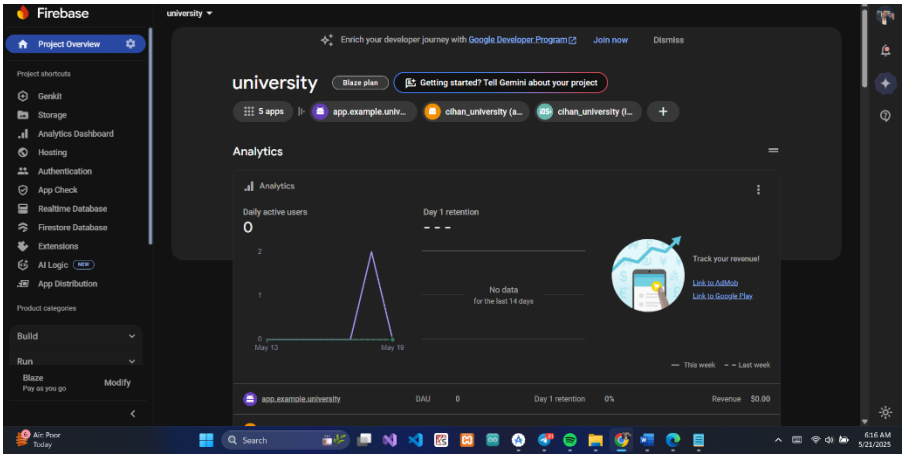


Figure 3.3, Complete configuring Firebase to flutter app

Chapter Four: Implementation

4.1 Implantation Overview:

This implementation overview presents the development and deployment of the **University Assistant Mobile Application**, designed to enhance student life and campus navigation within the **Kurdistan Region of Iraq**. The application covers all major components—from initial setup and secure user authentication to core functionalities such as location-based building discovery, event management, AI-powered chatbot support, and feedback submission.

The project integrates multiple technologies to create a modern, responsive, and scalable system:

- The **Flutter framework** was chosen for its cross-platform compatibility and ability to deliver a clean, Material Design-compliant user interface.
- **Firebase services** (Firestore, Authentication, and Storage) serve as the backend infrastructure, ensuring secure data handling, real-time updates, and media storage.
- An **AI chat assistant** powered by **OpenAI's GPT API** is embedded to support students with tasks such as navigation assistance, event inquiries, and general campus information.

Notably, a fine-tuned model—**ft:gpt-4o-mini-2024-07-18:personal::BK2HmLcW**—was utilized to enhance the chatbot's performance with domain-specific knowledge, enabling it to deliver personalized, relevant, and context-aware responses tailored to university-related interactions. Through this cohesive tech stack, the University Assistant App offers a seamless, student-centered experience that simplifies daily campus activities, improves accessibility to information, and supports a digitally transformed academic environment.

4.2 User Authentication Process

This This section describes the user authentication workflow implemented in the University Assistant Mobile Application, a critical component for securing user access and protecting sensitive information. Upon launching the app, students and staff are prompted to log in using their registered email and password credentials. To enhance convenience, the app also supports Google Sign-In, allowing quick and secure access through existing Google accounts. New users can create an account by selecting the Sign Up option, where they provide essential information such as username, email, and password, with a confirmation step to ensure accuracy. Both login and registration processes are securely handled via Firebase Authentication, which manages user credentials and sessions with strong security measures. The system uniquely identifies each user by storing their email as a User ID, enabling personalized experiences and secure access to individual data, such as saved campus locations, event preferences, and chatbot interactions. By leveraging Firebase's robust authentication services, the application ensures data integrity and safeguards user privacy, reinforcing the overall security framework essential for campus operations and user trust.

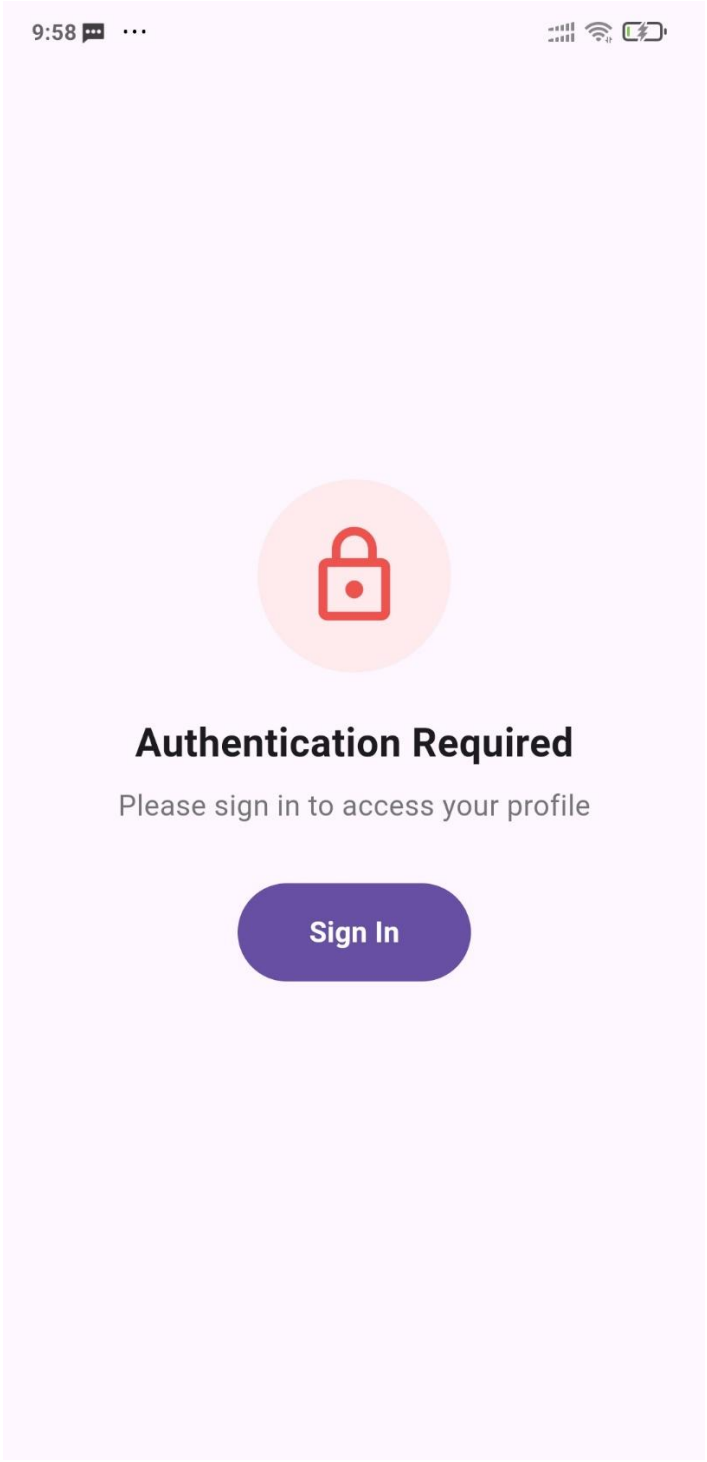


Figure 4.1, Authentication Page

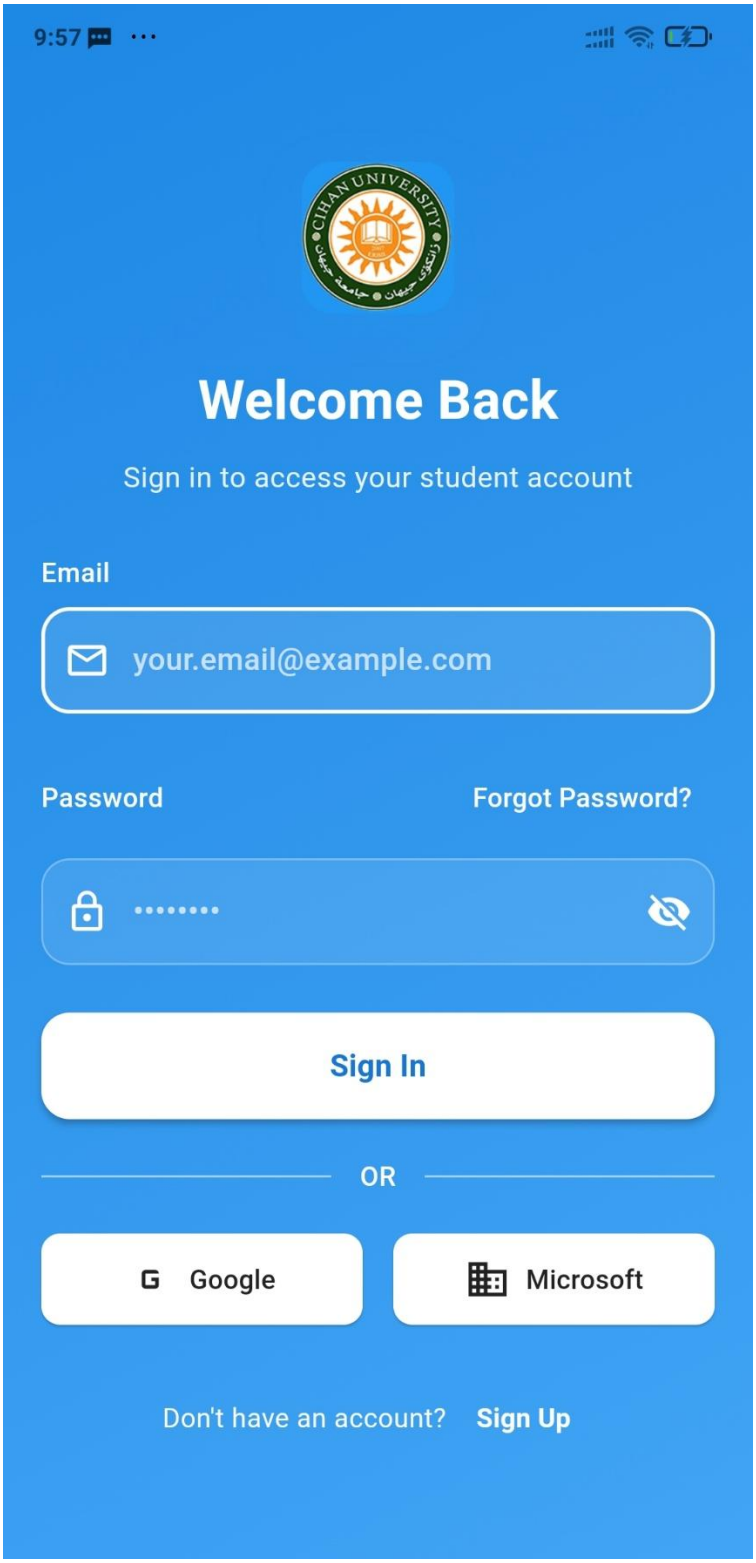


Figure 4.2, Sign in Page

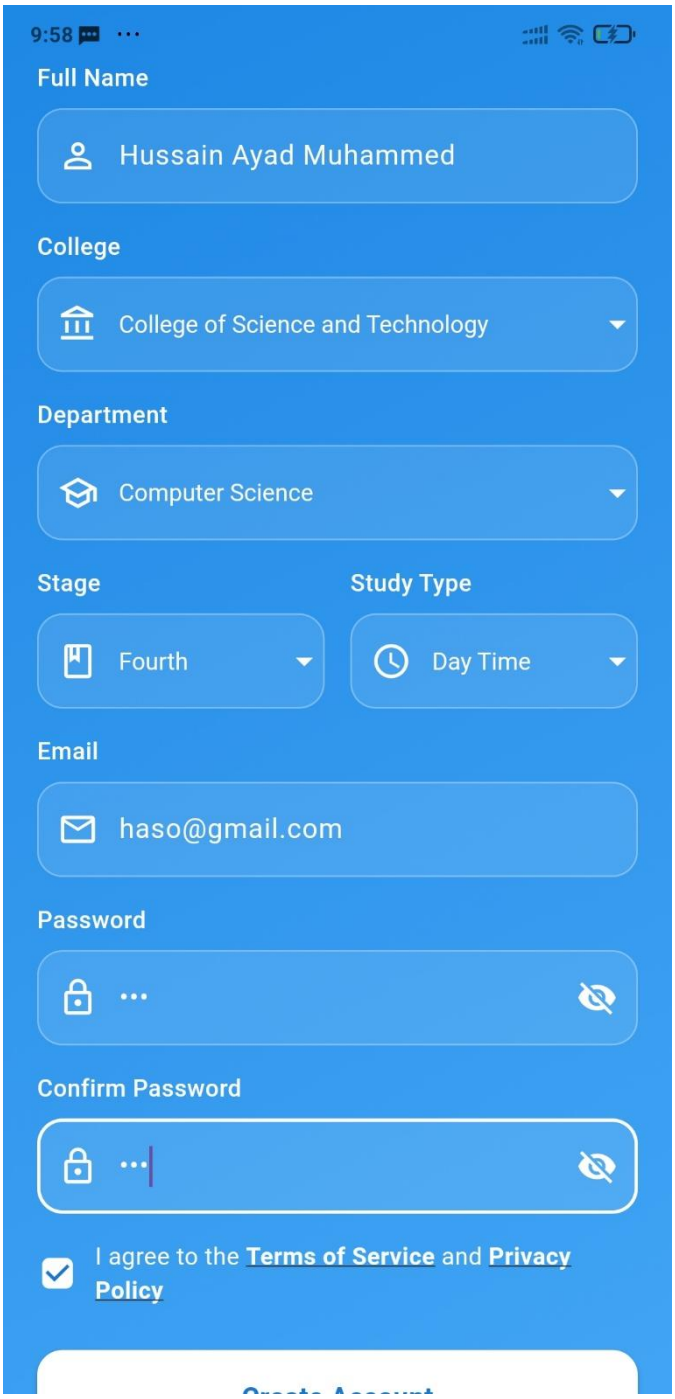


Figure 4.3, Sing up Page

This function is responsible for registering a new user in a Flutter application using Firebase Authentication. It takes an email and a password as input and attempts to create a new user account on Firebase with these credentials. If the registration is successful, it retrieves the user object from the response and proceeds to store that user's information in Firestore, which is a cloud-based NoSQL database provided by Firebase.

The function first waits for Firebase to create the user with the given email and password. If this operation succeeds, the user information is extracted from the result. The function then checks if the user object is not null, which means the account creation was successful. In that case, it calls another function to add the user details to Firestore, ensuring that the user is both registered in the authentication system and also has their data saved in the app's database.

However, if anything goes wrong during the registration process, such as the email being already in use, the password being too weak, or the email format being invalid, the function catches the Firebase authentication error. It prints the error message for debugging purposes and then returns null to indicate that the registration failed.

```dart
// Sign up with email and password
Future<User?> signUpWithEmail(String email, String password) async {
  try {
    UserCredential userCredential =
        await _auth.createUserWithEmailAndPassword(
      email: email,
      password: password,
    );

    User? user = userCredential.user;
    if (user != null) {
      await addUserToFirestore(user);
    }
    return user;
  } on FirebaseAuthException catch (e) {
    print("Error: ${e.message}");
    return null;
  }
}
```

Figure 4.4, Sign-up Code

4.2 Home Screen

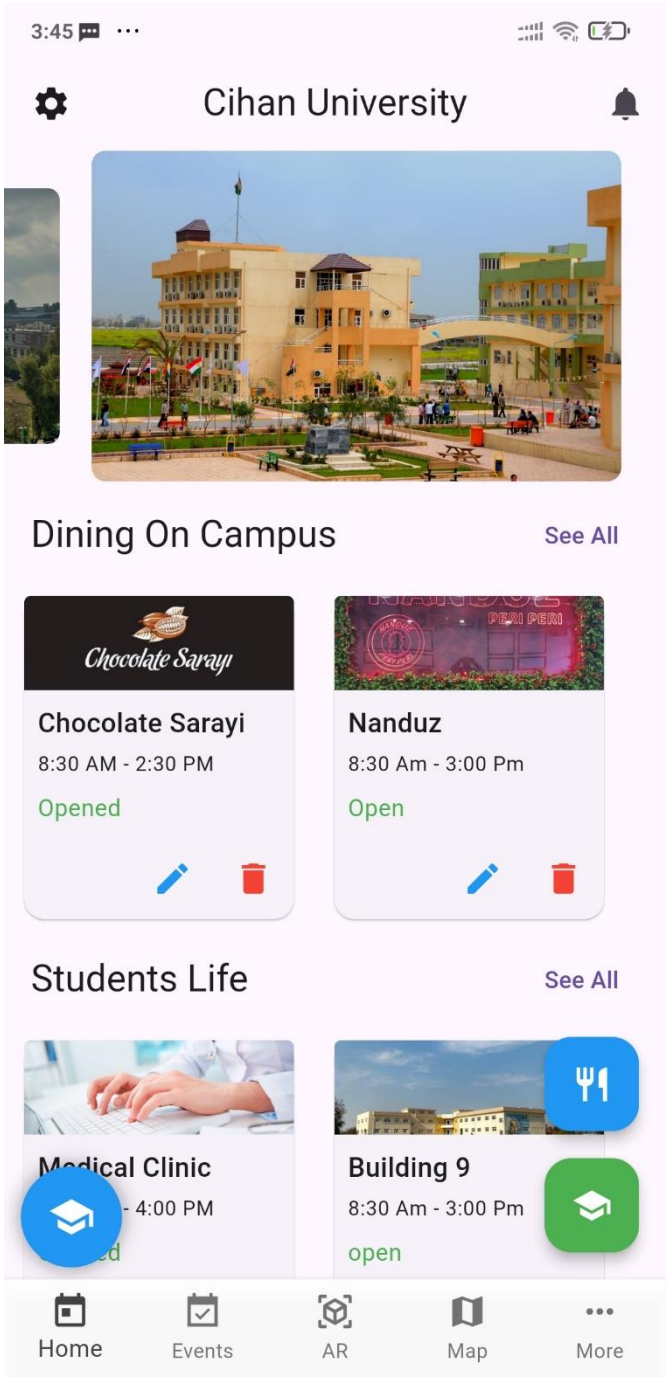


Figure 4.5, Home Screen

The HomeScreen serves as the central hub of the Cihan University app, offering students a clear overview of on-campus dining options and student life services. It displays real-time status updates

for facilities like Chocolate Sarayi and Nanduz, including their operating hours (e.g., 8:30 AM - 2:30 PM) and current availability (e.g., "Opened"). Additionally, the screen provides essential information about student services such as the Medical Clinic and Building 9, with their respective schedules. This feature ensures students can easily access up-to-date information about campus amenities.

```
class _HomeScreenState extends State<HomeScreen> {
  int _selectedIndex = 0;

  static final List<Widget> _widgetOptions = <Widget>[
    HomePageContent(),
    EventsPage(),
    ARPage(),
    MapScreen(),
    MorePage(),
  ];  // <Widget>[]

  static final List<String> _appBarTitles = [
    'Cihan University',
    'Events',
    'AR',
    'Campus Map',
    'More',
  ];

  bool _shouldShowIcons(int index) {
    return index == 0;
  }
```

Figure 4.6, Navigator Nail Code

The Cihan University app's HomeScreen dynamically manages campus information through intelligent background processes. Key features include:

- **Real-Time Status Updates**
  Automated checks verify the operating status of dining facilities (like Chocolate Sarayi and Nanduz) and student services (such as the Medical Clinic), displaying accurate "Open"/"Closed" labels and hours (e.g., "8:30 AM – 3:00 PM").
- **Efficient Data Handling**
  Smart caching reduces server requests, while optimized timers refresh critical information without draining resources. This ensures students always see up-to-date schedules (e.g., "Building 9: Open until 3:00 PM").
- **Seamless Navigation**
  The "See All" sections for **Dining On Campus** and **Students Life** are powered by lazy-loaded content, enabling fast access to details like clinic hours ("4:00 PM") or dining options.
- **Resource Management**
  Background processes clean up unused data and timers when not needed, maintaining app performance and battery efficiency.

By combining live updates with streamlined operations, the HomeScreen delivers a reliable hub for students to navigate campus services effortlessly.
application.

4.3 Event Screen

The **Events** section of the Cihan University app serves as a centralized platform for students and faculty to stay updated on university activities, news, and important announcements. Key features include:

- **Upcoming Events**
  Displays notable occasions like the *Gratitude & Appreciation Ceremony* (Dec 10, 2024), including details such as participants (e.g., *Prof. Dr. Amjad Saber Al-Dalawi, President of Cihan*) and event types.

- **TOP News**

  Highlights the latest university updates with a *"See All"* option for expanded coverage, ensuring users never miss critical information.

- **Department-Specific Content**

  Tailored sections (e.g., *Computer Science Students*) provide targeted announcements and community engagement opportunities (*"Hundred You"*).

- **Easy Navigation**

  The bottom menu (*Home, Events, AR, Map, More*) offers quick access to all app functionalities, enhancing user convenience.

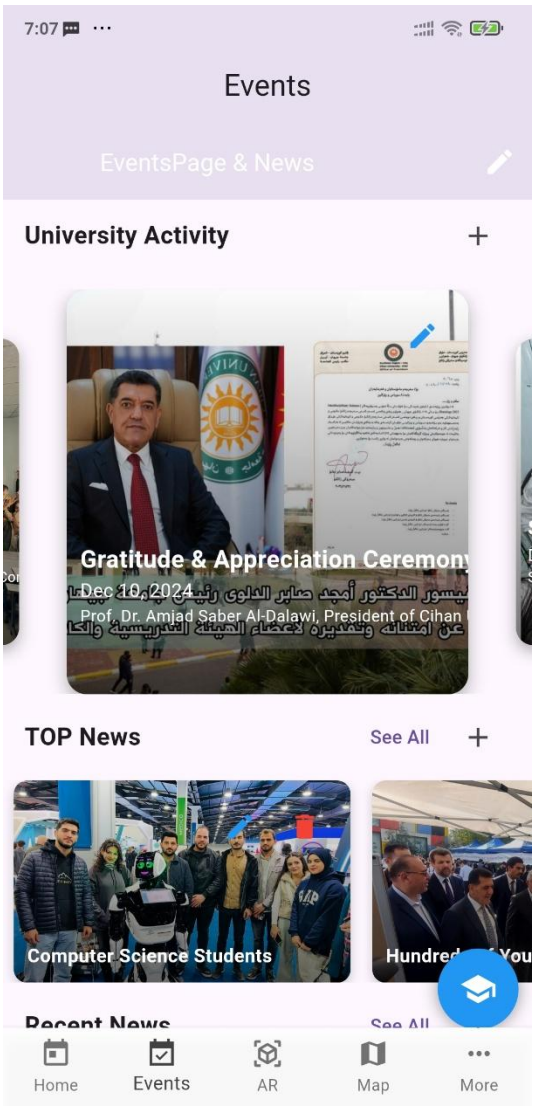


Figure 4.7, Events Screen

Figure 4.8, Top News Page

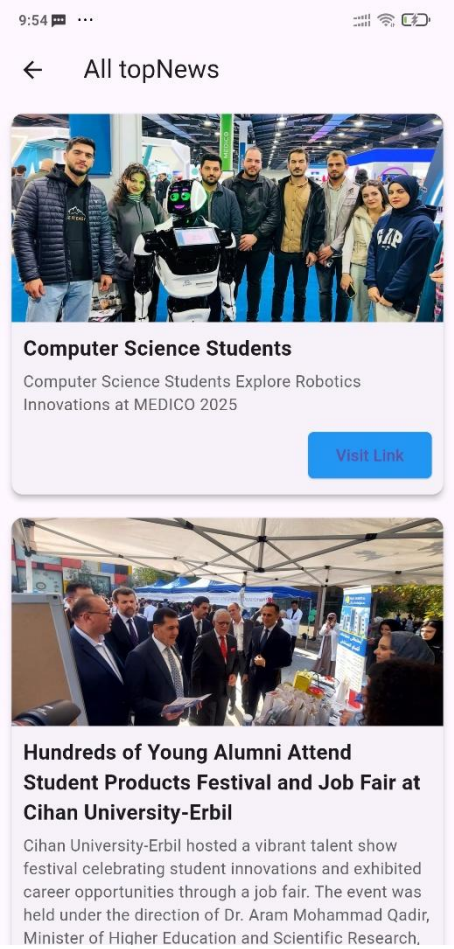
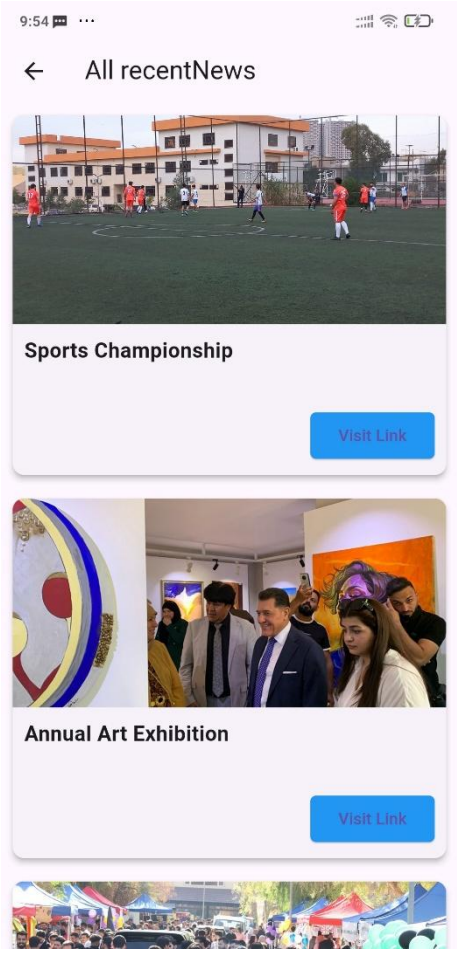


Figure 4.9, Recent News Page



4.4 AR Screen

The AR Navigation System that combines QR code scanning with augmented reality wayfinding and voice-guided directions. This feature helps students navigate campus efficiently by scanning destination QR codes (like those on admission forms or department signs) and displaying real-time AR arrows with turn-by-turn voice assistance.

Key Features:

1. **QR Code Scanning & Destination Detection**

- Users scan QR codes (e.g., from admission forms or department posters) containing encoded location data (e.g., *"Computer Science Department"*).
- The system decodes the QR (e.g., #FO-BK-87-94) and fetches the destination coordinates from the university's database.

2. **AR Wayfinding with Visual Arrows**

- **3D Path Indicators**: Overlays directional arrows on the camera feed (e.g., *"Continue straight in 216 m"*).
- **Distance Tracking**: Displays real-time remaining distance (e.g., *"7346.6 meters remaining"*).
- **Google Maps Mini-View**: Embeds a small interactive map (via Google Maps API) showing the full route.

3. **Voice-Guided Navigation**

- Provides spoken instructions (e.g., *"Turn left at the next corridor"*) until the destination is reached.
- Adjusts volume dynamically based on ambient noise.

4. **Multi-Platform Integration**

- **Backend**: Links QR data to campus maps (e.g., *"Evening study"* locations).
- **Google Maps API**: Pulls building layouts and optimizes indoor/outdoor routes.

Technical Implementation:

Backend Workflow

1. **QR Decoding**:
    - Extracts unique IDs (e.g., Application Form Number) and matches them to campus locations.
2. **Route Calculation**:
    - Uses Google Maps' Directions API to generate the shortest path.

Frontend AR Components

- **ARKit/ARCore**: Renders arrows and distance markers.
- **Voice Synthesis**: Converts route steps to audio via TTS (Text-to-Speech).

State Management

- **Navigation Session**: Ends automatically when the user arrives or manually via the bottom menu (*Home, Events, AR, Map, More*).

Example User Flow

1. **Scan**: Student scans a *"Computer Science"* QR code.
2. **Start Navigation**: AR arrows appear, and voice says, *"Head northwest for 50 meters."*
3. **Progress Tracking**: Mini-map updates as the user moves.
4. **Completion**: System announces, *"You have arrived at the Computer Science Department."*

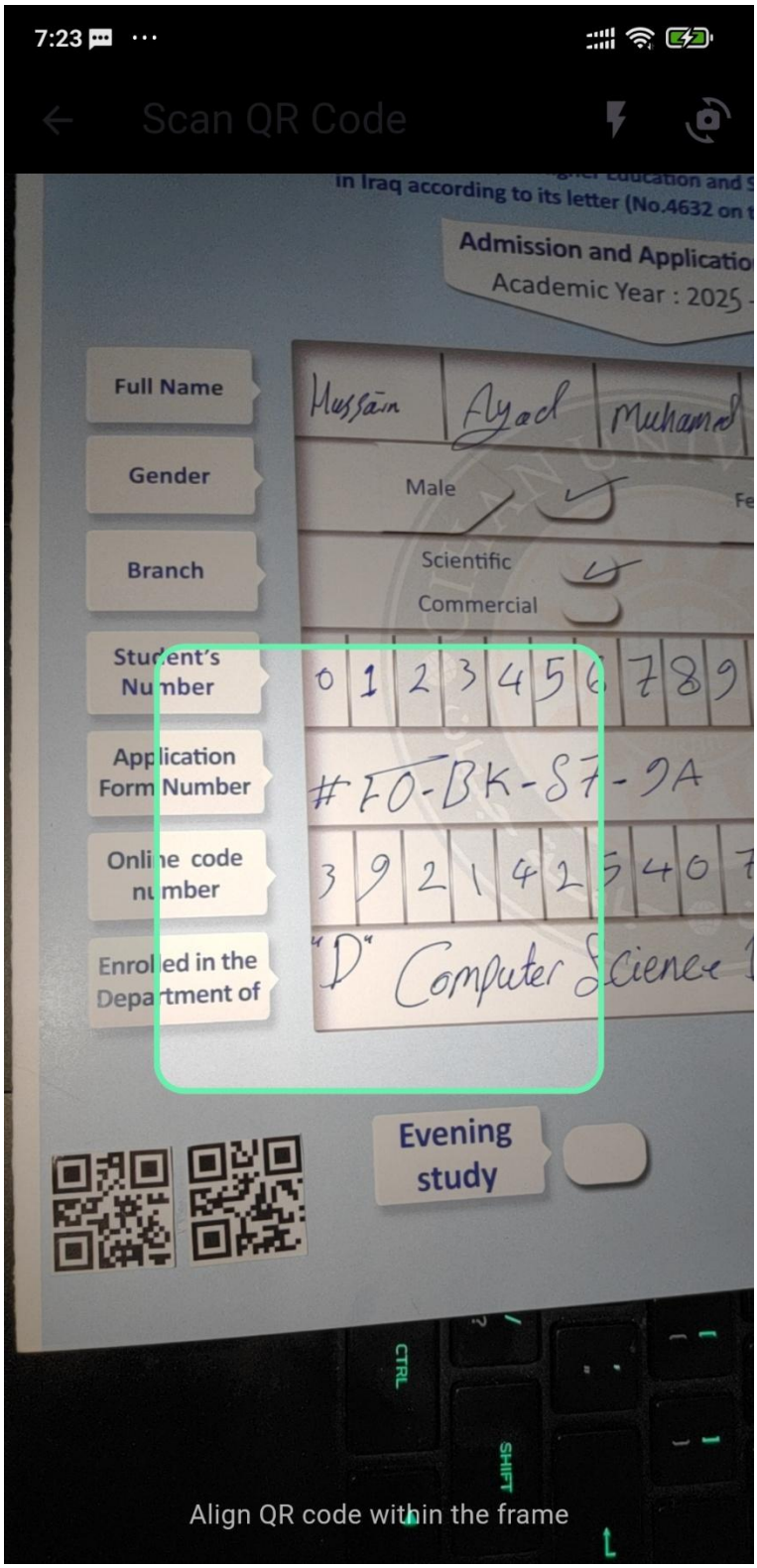


Figure 4.10, QR Code Scanner Screen

Figure 4.11, AR instruction and mini map Screen

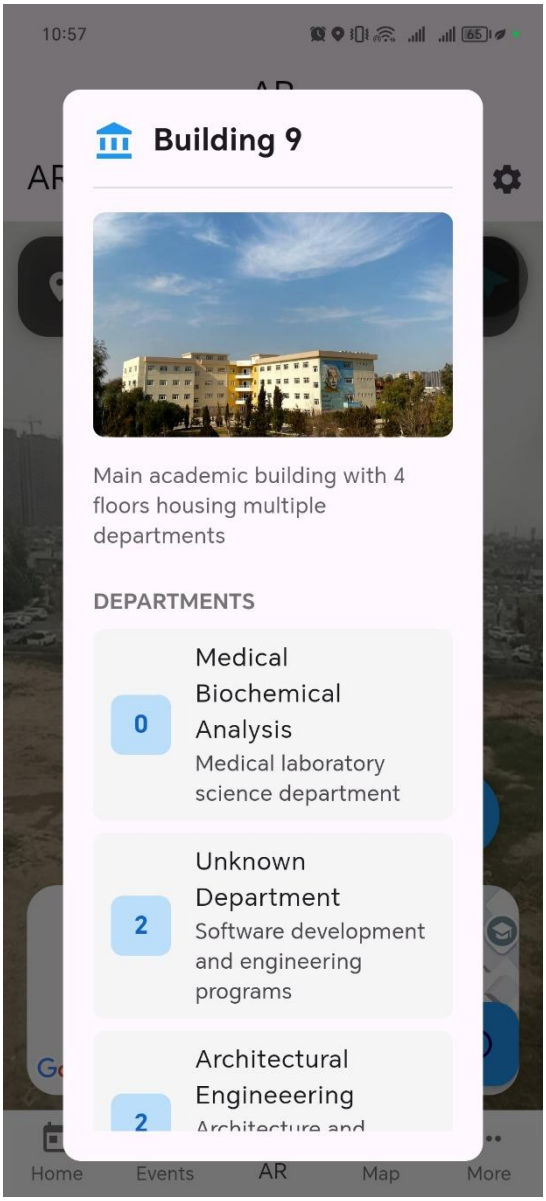


Figure 4.12, AR POP-UP Screen

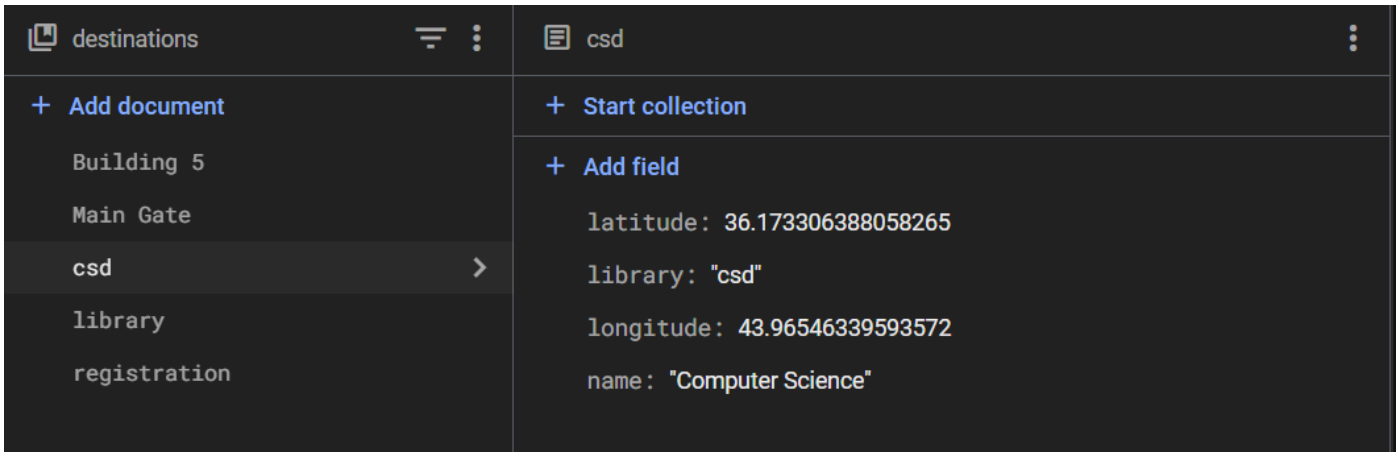


Figure 4.13, Buildings Database that stored in FireStore Screen

```dart
void _checkIfReachedBuilding() {
  if (_currentPosition == null || _loadingBuildingInfo) return;

  final building9Position = LatLng(36.1731660175881, 43.96549316381114);
  final distanceToBuilding = _calculateDistance(_currentPosition!, building9Position);

  if (distanceToBuilding < 20 && !_hasShownBuildingInfo) {
    _showBuildingInfoDialog();
    _hasShownBuildingInfo = true;
  } else if (distanceToBuilding > 30) {
    _hasShownBuildingInfo = false;
  }
}
```

Figure 4.14, Near Destination Code

```dart
Future<void> _getRoutePoints() async {
  if (_currentPosition == null || _destinationPosition == null) return;

  try {
    PolylineResult result = await polylinePoints.getRouteBetweenCoordinates(
      _googleApiKey,
      PointLatLng(_currentPosition!.latitude, _currentPosition!.longitude),
      PointLatLng(_destinationPosition!.latitude, _destinationPosition!.longitude),
      travelMode: TravelMode.walking,
    );

    if (result.points.isNotEmpty) {
      _pathPoints = result.points
          .map((point) => LatLng(point.latitude, point.longitude))
          .toList();
    } else {
      _pathPoints = [_currentPosition!, _destinationPosition!];
    }

    _updatePolylines();
  } catch (e) {
    print("Error getting route: $e");
    _pathPoints = [_currentPosition!, _destinationPosition!];
    _updatePolylines();
  }
}
```

Figure 4.15, MiniMap Code

```
String _getDirectionFromBearing(double bearing) {
  final adjustedBearing = (bearing - _deviceHeading + 360) % 360;

  if (adjustedBearing >= 337.5 || adjustedBearing < 22.5) return "straight ahead";
  if (adjustedBearing < 67.5) return "slightly to your right";
  if (adjustedBearing < 112.5) return "to your right";
  if (adjustedBearing < 157.5) return "sharply to your right";
  if (adjustedBearing < 202.5) return "behind you";
  if (adjustedBearing < 247.5) return "sharply to your left";
  if (adjustedBearing < 292.5) return "to your left";
  return "slightly to your left";
}
```

Figure 4.16, Text to Speech short code

```
void _onDetect(BarcodeCapture capture) async {
  if (_hasScanned) return;

  final barcode = capture.barcodes.first;
  final code = barcode.rawValue;

  if (code != null) {
    _hasScanned = true;

    print("✅ QR Scanned: $code");

    await controller.stop();
    await _audioPlayer.play(AssetSource('sounds/qr.mp3'));

    if (mounted) {
      Navigator.of(context).pop(code); // ✅ closes the page
    }
  }
}
```

Figure 4.17, QR-CODE Short Code

4.5 Map Screen

The **Map Screen** provides an interactive, searchable campus navigation system, integrating **Google Maps API** with categorized location filters. Users can quickly find colleges, cafeterias, services, and specific buildings across all campus.

1. **Key Features**

Dynamic Map Interface

- **Google Maps Integration**:
    - Displays real-time campus layouts with building markers.
    - Supports zooming, panning, and location tracking (blue dot).
- **Custom Markers**:
    - Color-coded pins for categories:
        - **Colleges** (Blue)
        - **Cafeterias** (Green)
        - **Services** (Red) (e.g., Medical Clinic, Library).

2. **Sort and Search Functionality**

- **Quick Filters**:
    - Tabs for **All**, **Colleges**, **Cafeterias**, and **Services**.
    - Subcategories (e.g., *Medical Biochemical Analysis*, *Accounting*).
- **Search Bar**:
    - Autocomplete suggestions (e.g., typing "Cihan" shows *Cihan University-Erbil*).
    - Supports fuzzy search (e.g., "Lib" → *University Library*).

3. **Building Details on Tap**

- Tapping a marker opens a **bottom sheet** with:
    - **Building name** (e.g., *Building 9*).
    - **Departments** (e.g., *Computer Science, Architecture*).
    - **Quick Actions**:
        - *AR Navigation* (launches AR arrows to the building).
        - *Save to Favorites*.

4. **Cross-Feature Integration**

- **Bottom Navigation Bar**:
    - Switch between *Home*, *Events*, *AR Navigation*, *Map*, and *More*.
- **AR Sync**:
    - *"Navigate Here"* button in building details launches AR mode with the same destination.

---

5. **Technical Implementation**

Backend

- **Google Maps API**: Fetches live campus maps and calculates walking paths.
- **Firebase Database**: Stores building coordinates, departments, and categories

6. User Flow Example

- Open Map Screen: Default view shows all buildings.
- Search for "Lib": Autocomplete suggests University Library.
- Tap Library Marker: Bottom sheet opens with AR navigation option.
- Select "AR Navigation": Switches to AR screen with directional arrows.

```
Future<void> _fetchBuildingsFromFirestore() async {
  try {
    final snapshot = await FirebaseFirestore.instance.collection('buildings').get();

    final buildings = snapshot.docs.map((doc) {
      final data = doc.data();
      return {
        'name': data['name'],
        'emoji': data['emoji'],
        'category': data['category'],
        'description': data['description'],
        'image': data['imageUrl'],
        'position': LatLng(data['lat'], data['lng']),
      };
    }).toList();

    final categories = ["All"];
    for (var b in buildings) {
      if (!categories.contains(b['category'])) {
        categories.add(b['category']);
      }
    }

    setState(() {
      _allBuildings = buildings;
      _categories = categories;
    });

    _loadMarkers();
  } catch (e) {
    print("Error fetching buildings: $e");
  }
```

Figure 4.18, Firebase Database code

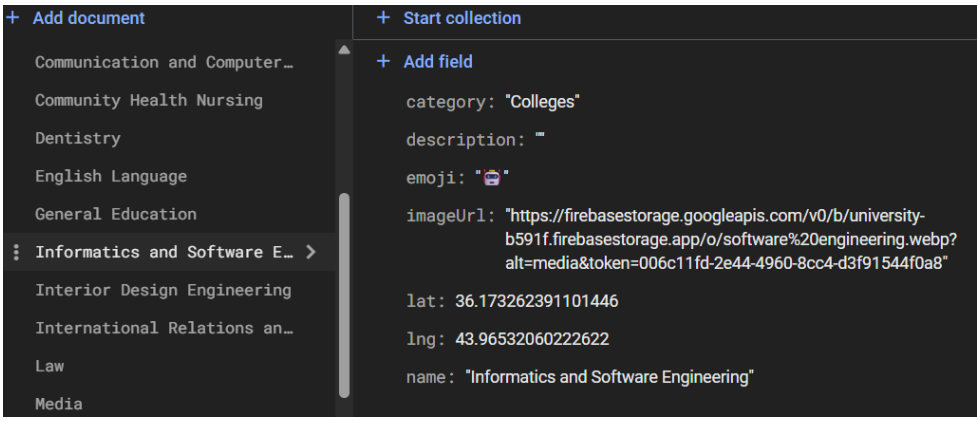


Figure 4.19, the database in FireStore Screen

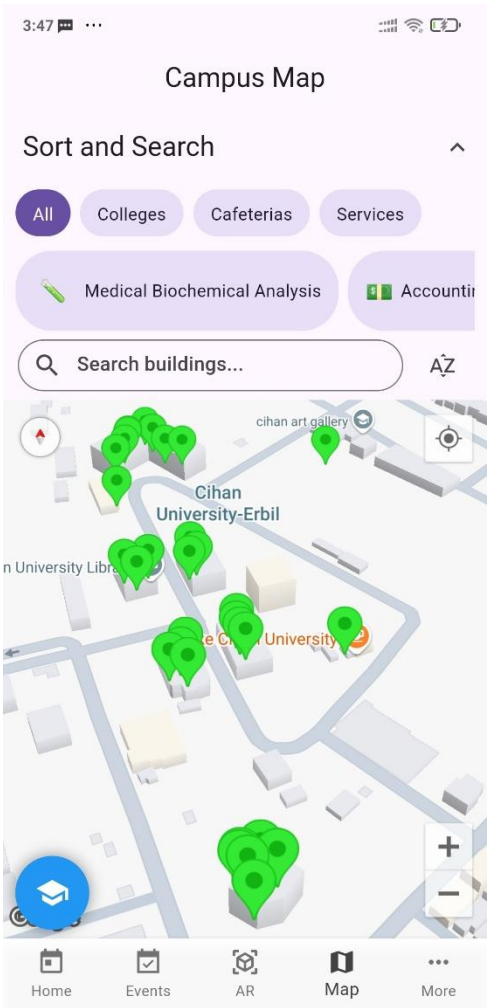


Figure 4.20, Interactive Map Screen

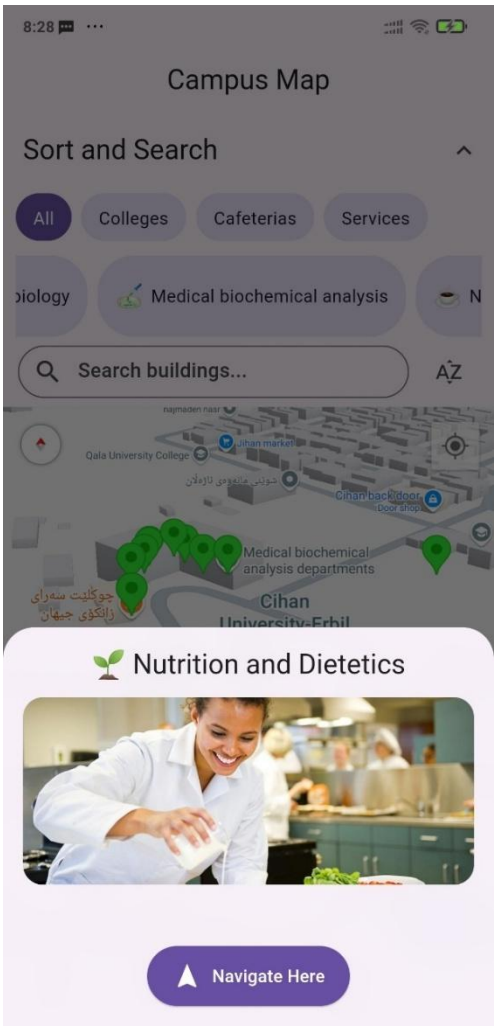


Figure 4.21, When You click the Pin the info screen shows up Screen

Admin Actions

The admin system allows authorized users to manage all aspects of the application including restaurants, users, Locations, payment installment and reservations. Admin functionality is role-based, requiring specific permissions stored in the user profile.

Core Features

1. **App Management**
    - Add, edit, and delete restaurants
    - Update restaurant details (location, hours, pricing)
    - Update Department details (location, Fee, Details)
    - Manage restaurants, users, Locations, payment images and descriptions
2. **User Management**
    - View all registered users
    - Change user roles (admin/regular)
    - Delete user accounts when necessary
3. **Fee Management**
    - View all pending, confirmed, and completed reservations
    - Confirm or reject installment requests
    - Mark installment as completed
4. **Analytics Dashboard**
    - View key metrics (active users, reservations, locations,)
    - Monitor business performance with charts and graphs
    - Track geographic distribution of users and campus

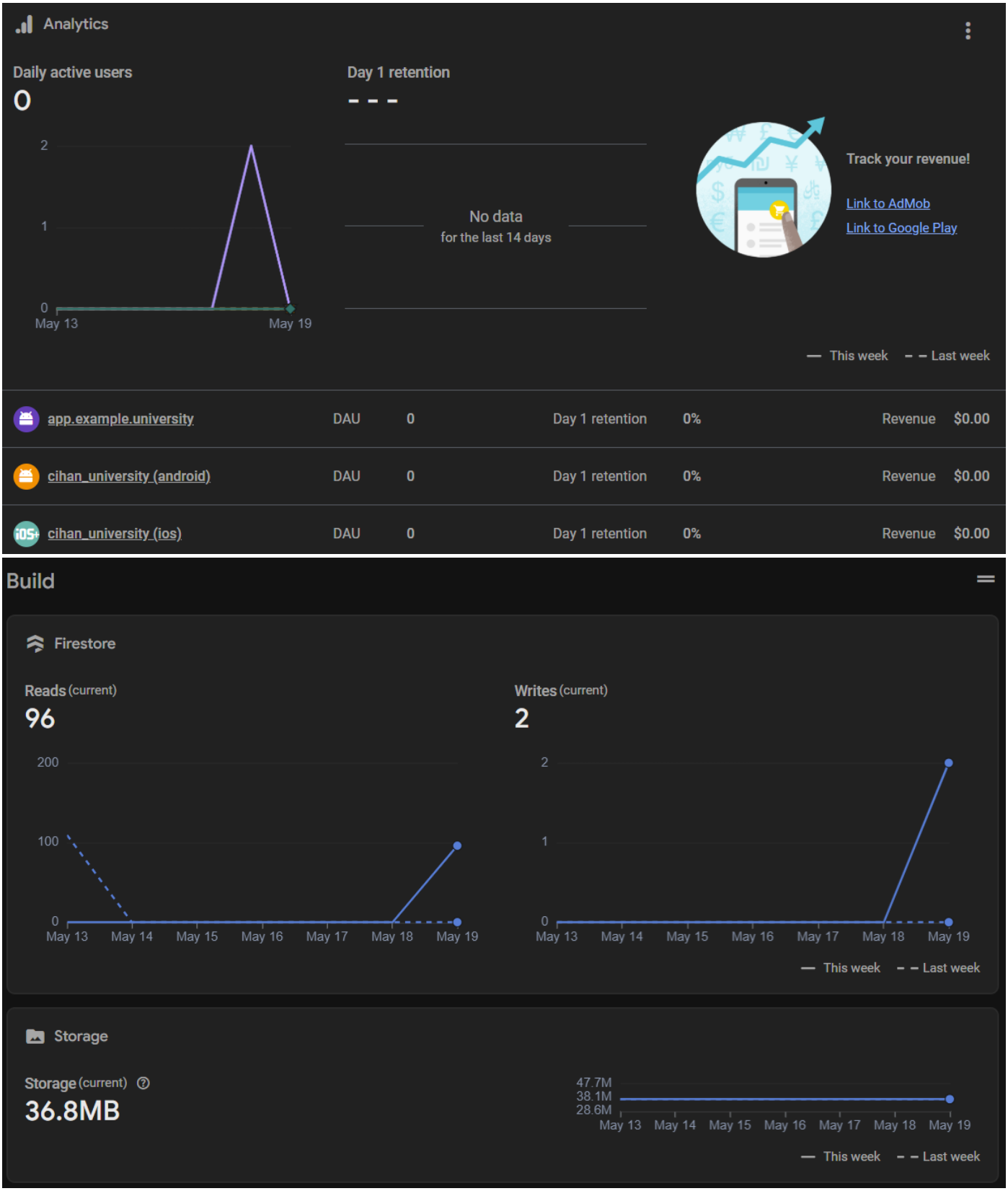


Figure 4.22, Analytics Dashboard

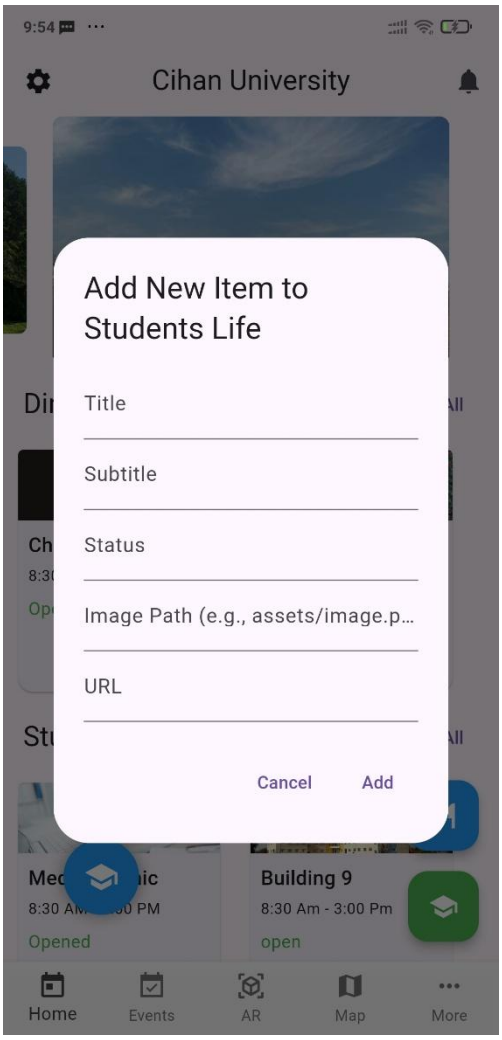


Figure 4.23, Admin Actions

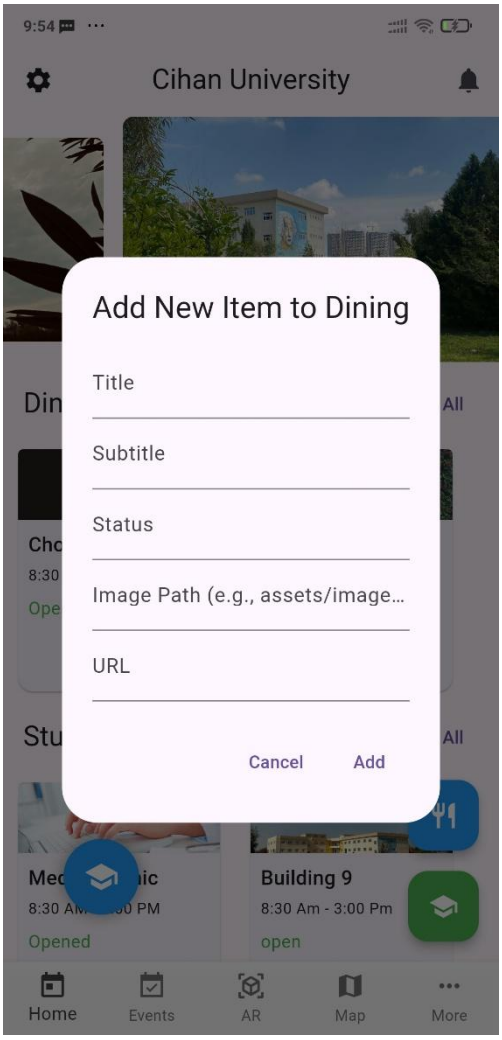


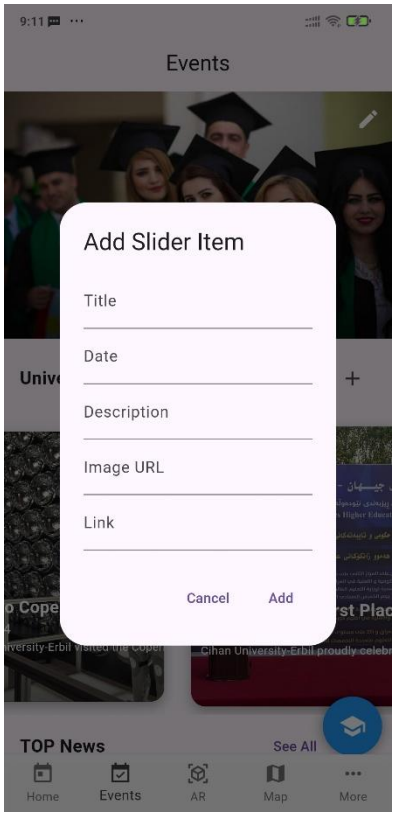
Figure 4.24, Admin Actions(2)

The admin system uses Firebase Authentication for secure access control and Firestore for data persistence, ensuring that only authorized users can perform administrative actions.

Add Restaurant

The "Add Restaurant" functionality is a key administrative feature that allows authorized users to add new restaurants to the system's database.

- **Form Input**: The add screen collects comprehensive restaurant details:
    - Basic info (name, image URL, description)
    - Location (area selection + map coordinates)
    - Cuisine selection from predefined options
    - Time for closing and opening
    - Operating hours for each day of the week




```
TextButton(
  onPressed: () async {
    await FirebaseFirestore.instance.collection(collection).add({
      'title': _titleController.text,
      'subtitle': _subtitleController.text,
      'status': _statusController.text,
      'imagePath': _imagePathController.text,
      'url': _urlController.text,
      'timestamp': FieldValue.serverTimestamp(),
    });
    Navigator.of(context).pop();
  },
  child: Text('Add'),
),  // TextButton
```

Figure 4.26, Add News Integrated with FireBase Code

Figure 4.25 Add University News

Manage Colleges

This admin-only dashboard allows authorized university staff to manage all campus facilities (colleges, labs, cafeterias, services) with real-time data control. Built with Flutter and Firebase, it provides a secure, intuitive interface for updating locations, departments, and facility details.

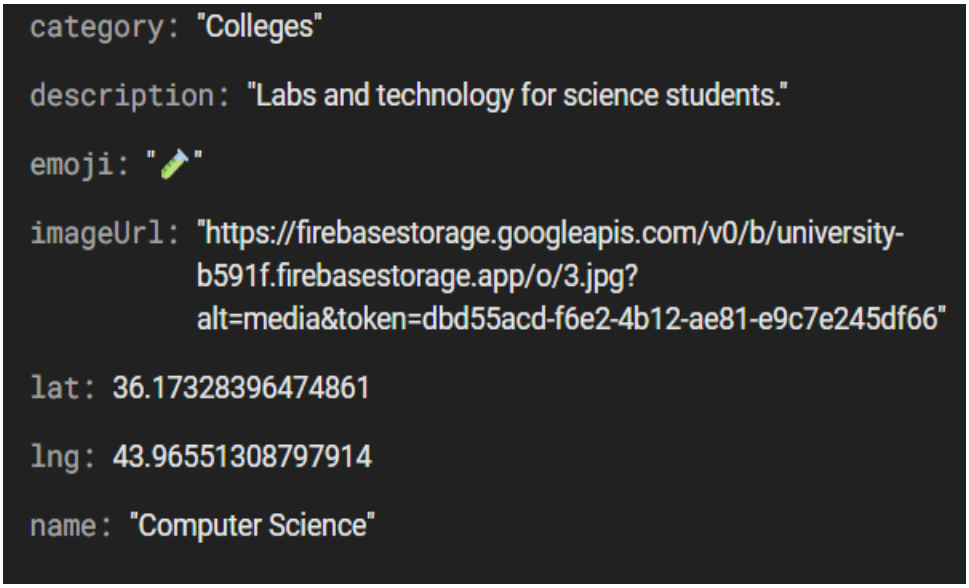


Figure 4.27, Edit Restaurant

User Management

The User Management system is an integral part of the admin panel in the university App, designed to give administrators full control over user accounts and permissions. Through this system, admins can access a complete list of all registered users, with options to filter by name, email, or user role. The list can also be sorted based on different attributes such as registration date, name, or level of activity, allowing for quick navigation and management. Selecting a user reveals detailed information including their profile, reservation history, submitted reviews, and login activity.

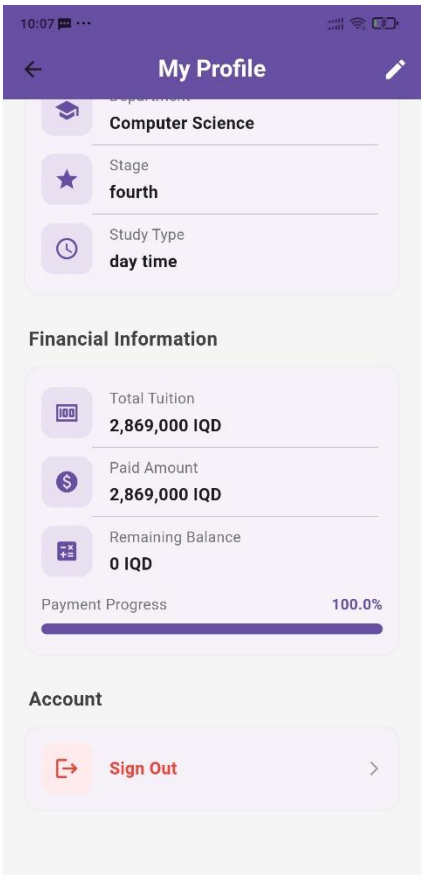


Figure 4.28, User Management

Admins can also manage roles directly from this interface, with the ability to assign or remove administrative privileges and adjust access to specific features within the app. If a user's information needs updating, the admin can edit their profile, reset their password, or modify their contact details. This centralized system ensures that user data is easily accessible and manageable while maintaining proper control over who has access to what within the application.

Analytics Dashboard

The Analytics Dashboard provides restaurant administrators with a comprehensive business intelligence center to track key performance metrics, visualize trends, and make data-driven decisions. This centralized hub collects and transforms raw data from across the application into actionable insights.

At the top of the dashboard, critical Key Performance Indicators (KPIs) are displayed in distinct metric cards. These include the total number of active users on the platform, the total number of reservations made, the average rating across all restaurants, and the total number of restaurants available. These cards give administrators an immediate snapshot of the platform's health, with visual indicators for quick performance assessment.

The dashboard also offers time-filtered analytics, allowing administrators to view data from different time periods, such as the past seven days, thirty days, ninety days, or one year. This allows for a flexible understanding of the trends and performance over various timeframes.

4.6 More Screen

The **Students** section of the Cihan University app is a meticulously designed hub that empowers students to manage their academic, financial, and logistical needs with ease. Combining intuitive navigation with robust functionality, this portal ensures every tool is just a tap away.

Users can interact with the screen by on the seaction.

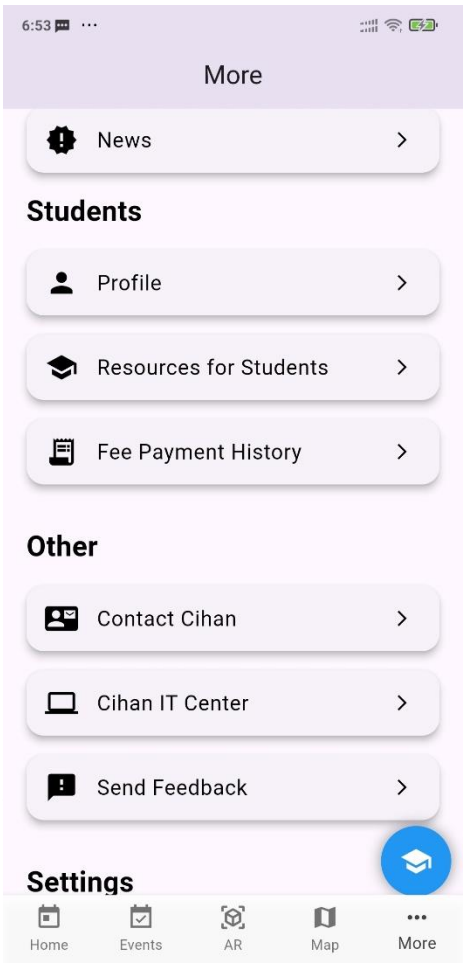


Figure 4.29, More Screen

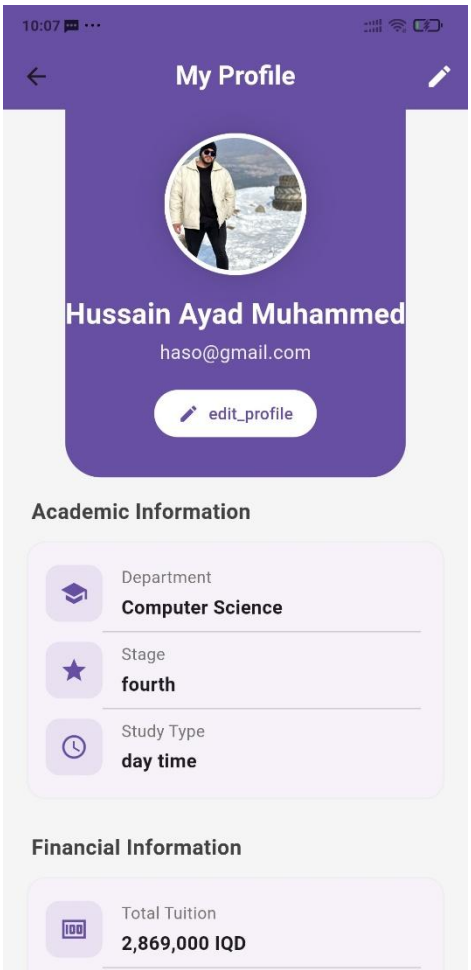


Figure 4.30, Profile Page

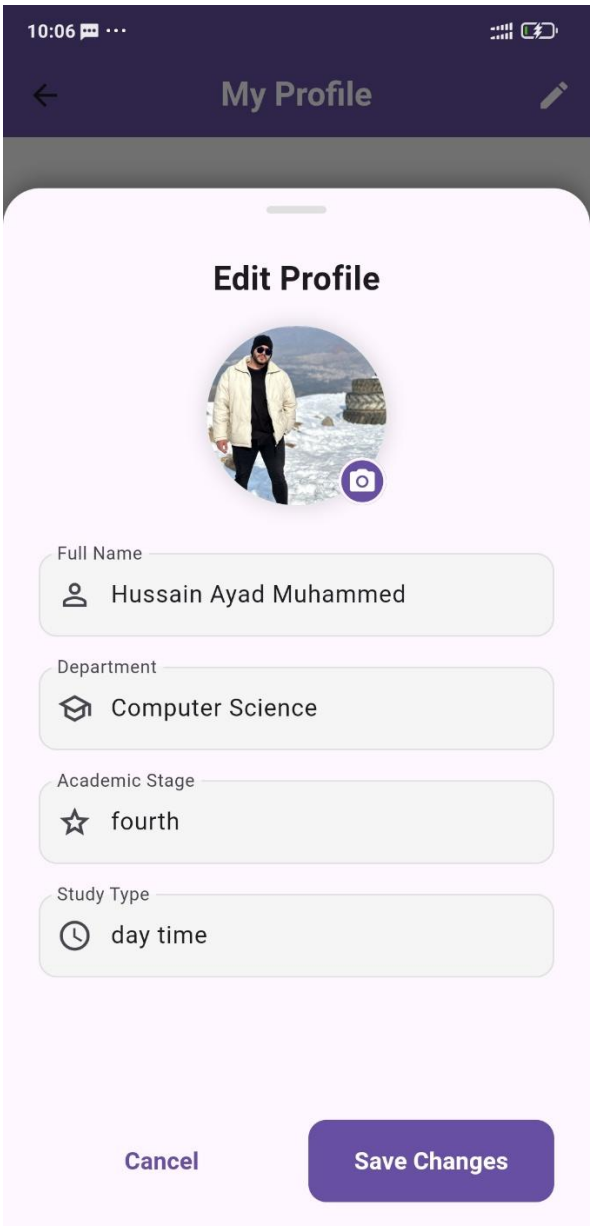


Figure 4.31, Edit Profile Page

4.6.2 Fee Payments

The The payment module enables students to view their tuition fee status and make payments directly through the app. This was implemented by integrating the app with a secure payment gateway and the university's financial database. Users have a Fee Payments screen where they can track their tuition fees and installments. The app fetches payment records from the Firestore collection (e.g., fee_payments) for the logged-in student and displays each transaction with details such as amount, date, payment method, and transaction ID. The progress bar and summary calculations (total, paid, remaining) are computed in real time based on these records.
To facilitate payments, the app includes an in-app Visa/Mastercard payment form (Figure 8). When a student opts to pay an installment, this form is presented for entering credit card information. The methodology required ensuring secure handling of sensitive data: card details are not stored on the device or Firestore, but transmitted via an encrypted connection to the payment gateway (for example, using a PCI-compliant service). Upon successful payment, a confirmation is returned and the Firestore fee_payments collection is updated with the new transaction entry (marked as "Paid"). We also implemented a receipt download function (note the download icon next to the payment entry in Figure 7) to allow users to save proof of payment. The payment interface was tested thoroughly for validation of inputs (card number format, expiration date, etc.) and error handling, to provide a smooth and secure user experience.easily.

First, the Restaurant Location Display shows the restaurant's location on the map. The map is interactive and centered on the restaurant's spot. A custom marker with a unique design highlights the location. It works well in both light and dark modes, adjusting automatically to the theme.

For getting directions, the Navigation & Routing feature helps users find the restaurant from their current location. It tracks the user's real-time position, calculates the route, and shows the distance and estimated travel time. The route is color-coded for clarity, and markers show both the user's location and the restaurant's destination. The route updates as the user moves, and there's a "My Location" button to quickly recenter the map on the user.

The External Maps Integration feature lets users open the restaurant location in their phone's default map app, like Google Maps or Apple Maps. This happens through an "Open in Maps" button, and the app passes the restaurant's coordinates for easy navigation.

The app uses FlutterMap with Leaflet maps for smooth display. It's compatible with both light and dark modes, and custom markers match the app's theme. The map adapts to different screen sizes, making it easy to use on any device.

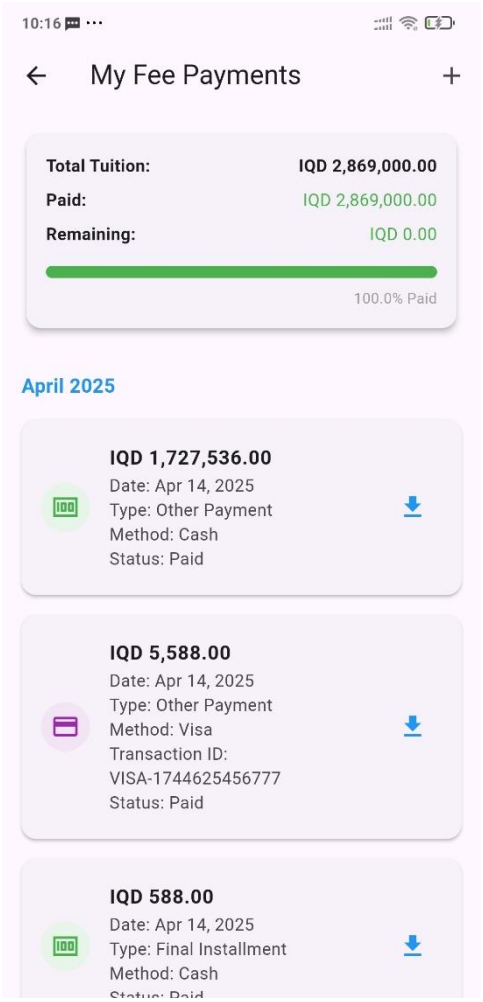


Figure 4.32, Fee payment dashboard

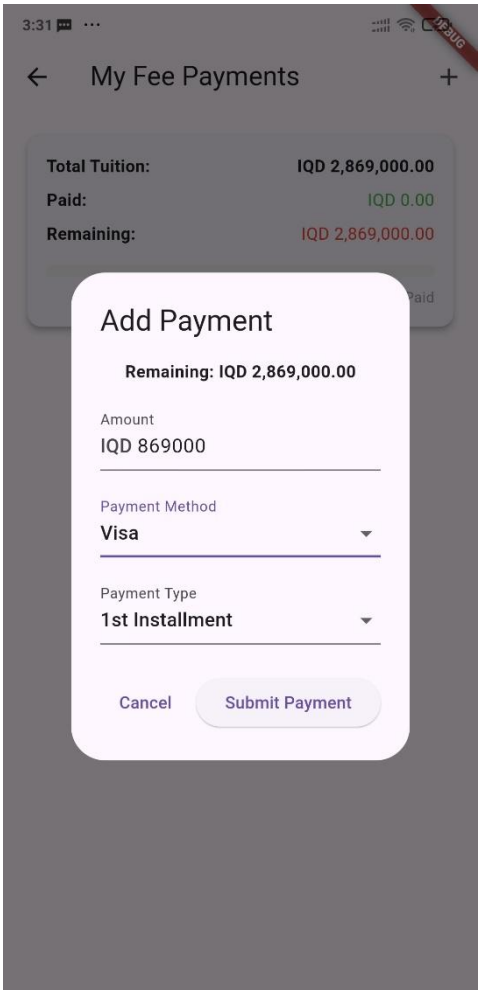


Figure 4.33, Choosing the way of payment

4.6.3 Other Section Page

In this section we have three options and they are Contact Chian, Cihan It Center, Send Feed back. And for each on of them there is a specific use and the will be shown down below as Figures.

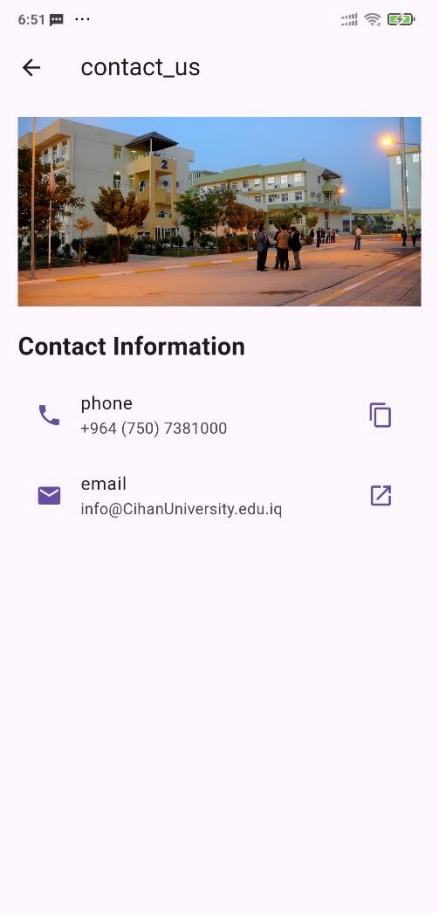


Figure 4.34: Contact Cihan Page

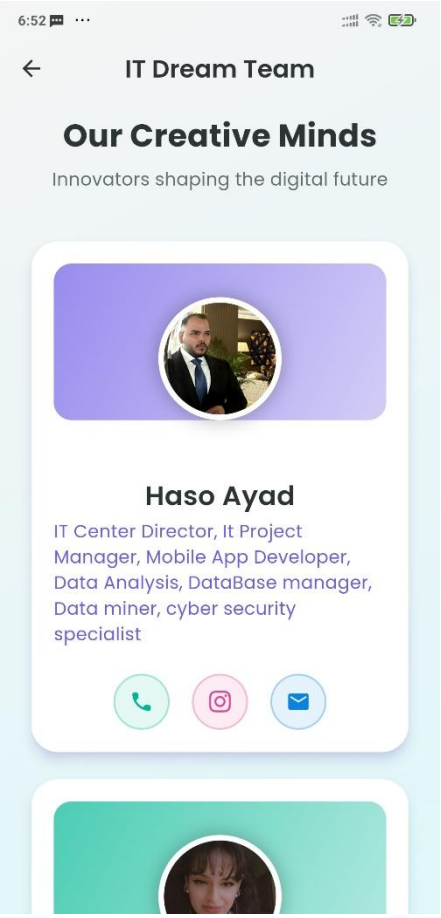


Figure 4.35: Cihan It Center Page

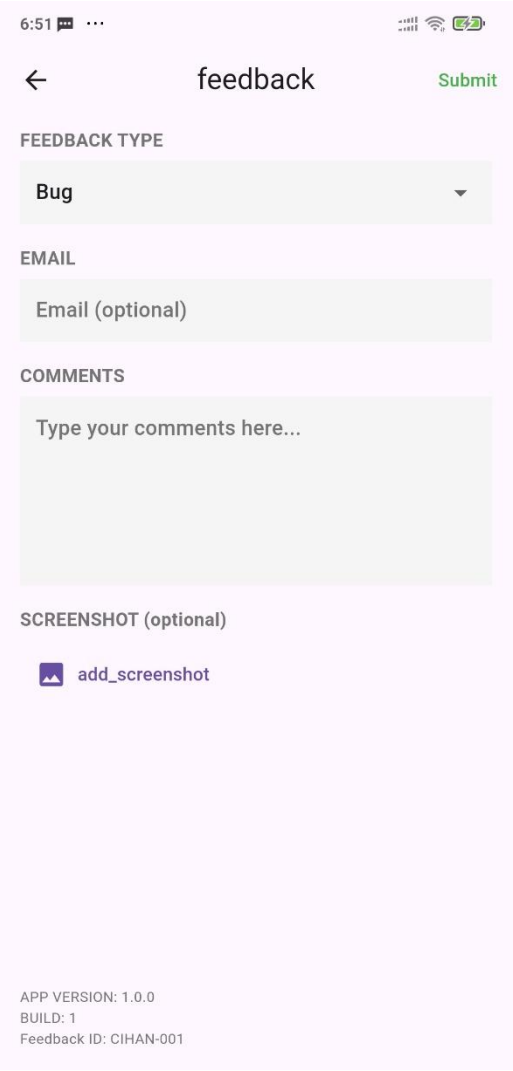


Figure 4.36, Sending feedback Page

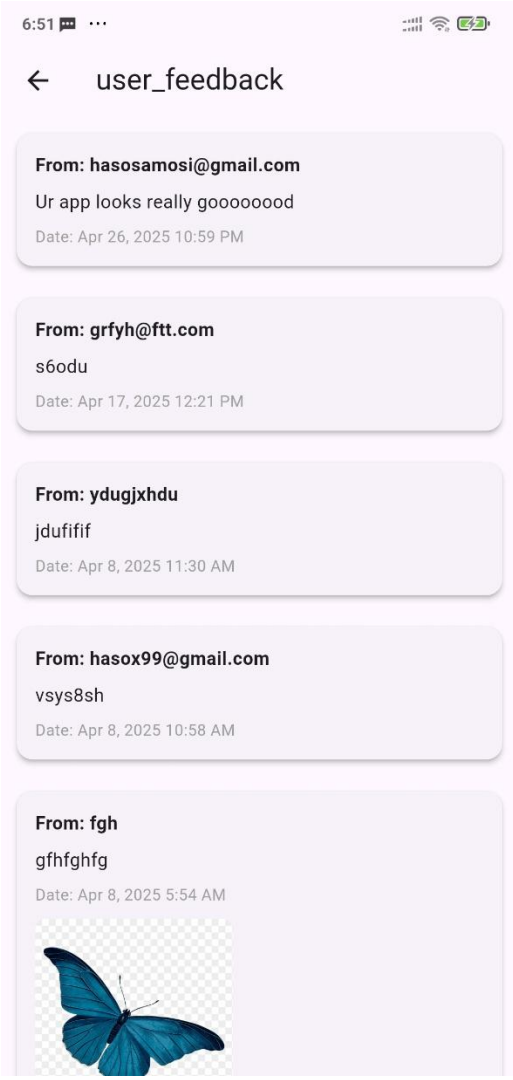


Figure 4.37, Reviewing the Feedback Page

4.7 Settings Screen

The **Settings** page from a university mobile application. The page includes several options for users to customize their experience:

1. **Language**: Set to "English," indicating the app's current language with potential options to switch to (Kurdish, Arabic, English Languages).
2. **Dark Mode**: A toggle likely available to switch between light and dark themes.
3. **Notifications**: Settings to manage app alerts and updates.
4. **App Version**: Displays the current version (1.0.0) for reference.
5. **Send Feedback**: A button or link for users to submit app-related suggestions or issues.
6. **Privacy Policy**: A section linking to the app's data handling policies.

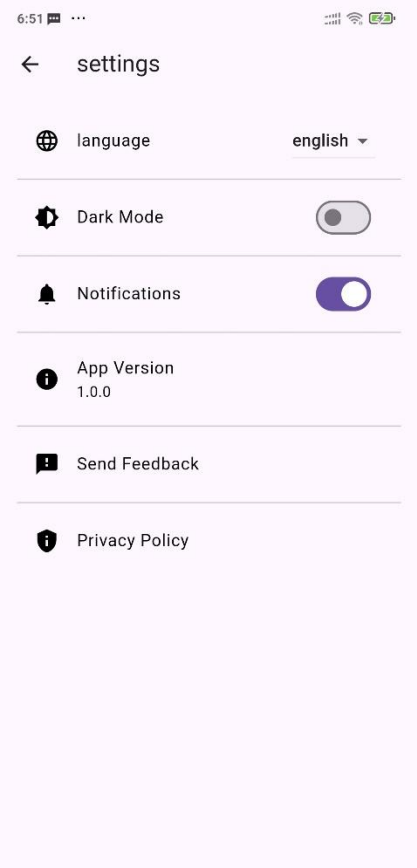


Figure 4.38, Settings Page

4.8 Ai chatbot

The **University Assistant Mobile Application** features a smart **AI-powered chatbot** designed to support students in navigating campus life and accessing key services with ease. This virtual assistant is capable of understanding user preferences, contextual needs, and real-time campus data to provide accurate and personalized responses.

The chatbot helps students with a range of tasks, such as:

- Finding nearby buildings or classrooms based on their current location.
- Answering questions like “Where is the cafeteria in Building A?” or “What events are happening today?”
- Providing real-time status updates—such as whether a facility is open or closed—and offering the next available hours if it’s currently unavailable.
- Tailoring responses based on the **time of day**, **day of the week**, and **user’s previous interactions** to improve recommendations over time.

Users engage with the chatbot through natural language conversations, receiving helpful and context-aware replies. Integrated with a fine-tuned GPT model, the assistant continuously learns and adapts, offering increasingly relevant support as students interact with it throughout the semester.

Core AI Components:

At the heart of the app’s assistant is a fine-tuned GPT-4o-mini model, specifically trained to handle restaurant recommendations. The model ID is ft:gpt-4o-mini-2024-07-18:personal::BK310z9c, and the assistant communicates with OpenAI’s API via HTTP requests to ensure the most up-to-date and accurate information is provided.

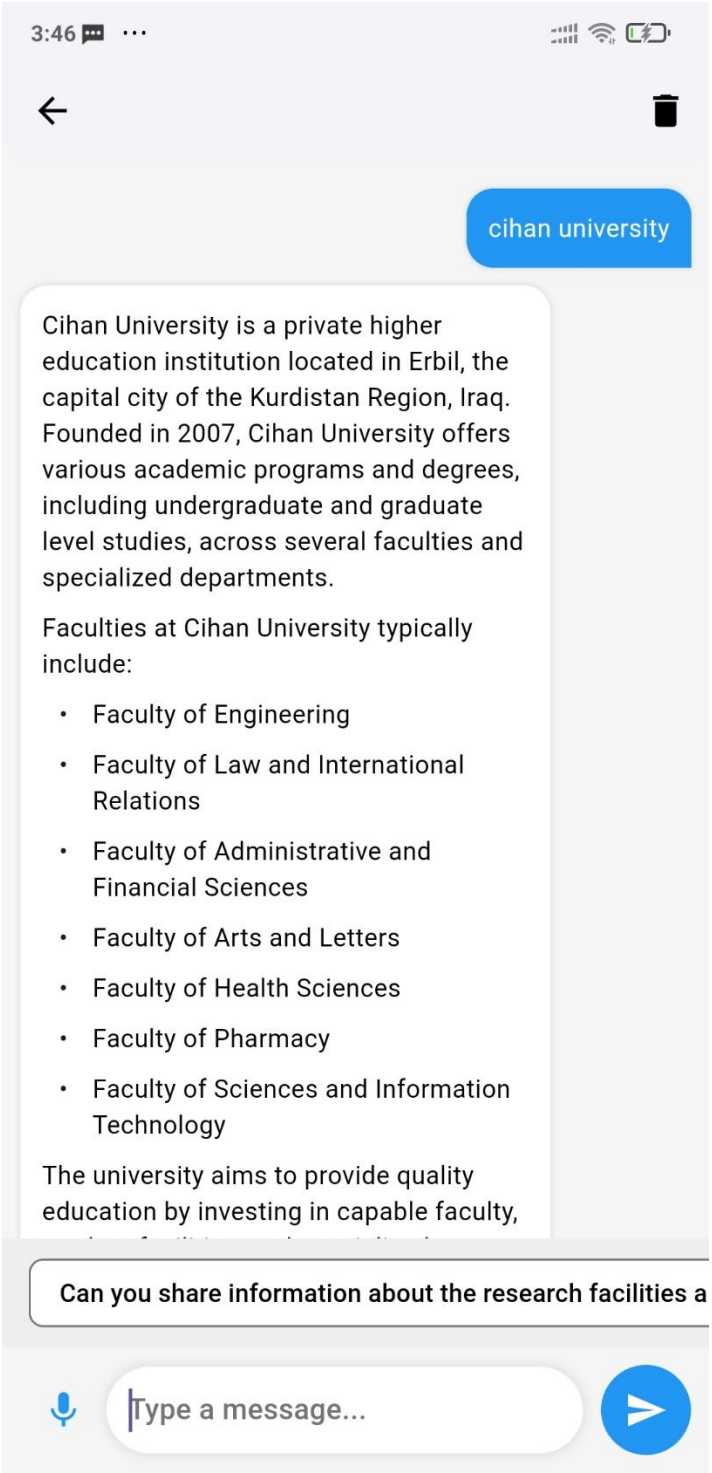


Figure 4.39, AI Chatbot Page

Intelligent Query Processing:

When a user sends a query, the app processes it in multiple steps:

1. **Query Analysis**: The assistant analyzes the query to identify location mentions, cuisine preferences, price range, and other relevant factors.
2. **Restaurant Matching**: It filters restaurants in the database to find those that match the user's criteria.
3. **System Prompt Creation**: The assistant constructs a specialized prompt that includes relevant restaurant information to ensure an accurate response.
4. **Formatted Response**: The final response is a well-structured message with restaurant details, such as names, locations, ratings, and open/closed status.

This intelligent, context-aware process ensures that users receive tailored, relevant restaurant suggestions, making the app both efficient and user-friendly. By considering location, time, and user preferences, the AI assistant delivers accurate, valuable recommendations that continually improve based on user interactions.

Chapter Five: Discussion & Result

5.1 Discussion & Result

Required to build a real-time AR navigation system Mult-technical including GPS and compass direction and 3D display is major challenge that was achieving challenge that was achieving accurate stock placement and alignment especially in external environments where movement and speed can change and take advantage of firebase the application is maintained directly and synchronized across the units and ensures that the student can immediately access the details of tuition fesse and installments in addition to payment methods and full details and public holidays for Kurdistan Region and full information about class and the teaching staff and to force improvements to mobility allow scanning of the QR code and quickly activate the track while voice instructions and vibration made the comments more smooth, easy and accessible and not forgetting to mention the tripartite support for Arabic, Sorani, English is crucial so the application made it comprehensive for all students this application also features a chatbot integrated with AI via the OpenAI application programming interface, which provides a reply and answers immediate for the students who has questions Provide immediate answers to common questions for students and direct them through the university Services it provides such as class schedules, events, latest news and support links This causes a significant and noticeable reduction in waiting time for help and adds a personal experience The interactive campus map enhances usability and the interactive campus map enhances usability and allows students to explore the university Restaurants Library Categories Classes Search by name and category and click on buildings to view descriptions and pictures The trends and all these features are creating a dynamic, user-centered ecosystem that has successfully bridged the gap between students and university resources that have confirmed real-world testing and high and accurate usability, making this application not only functional but a model for future campus platforms.

As a Result it is proven that the application of navigation with augmented realty technology integrated with AI and QR code Scanner for the smart campus at Cihan University is stable, easy to use facilitates the life of students and combines the provision of navigation, guidance, academic ,services , real-time updates and realization of the AR steering system Accuracy of 90% with QR-based activation the three-dimensional arrows work with GPS improve ,movement on campus and enhance interactive maps and chatbots powered by AI cable usage allows the student to explore the campus and access information quickly Multilingual support the Arabic , English ,Sorani

language worked smoothly, ensuring wide.

Chapter Six: Conclusion & References

Conclusion

This project is all about making campus life more connected and accessible-right from your pocket.

Weve created a smart, all-in-one university mobile app designed to make everyday tasks simpler and the campus experience more engaging. With flutter and firebase as its foundation, the app delivers a smooth, responsive performance while offering powerful tools that really matter to students and staff-whether its checking updates, getting around, or staying informed, here's a quick look at what the app do:

Real-time data integration: stay in the know with up-to-the-minute updates on events, announcements, and key campus information-thanks to firebase fire store a syncing everything live inside the app

AR navigation: Getting around campus is now a breeze, with AR Core powering the experience, you'll be guided in real time by floating 2D-3D arrows and path markers that lead the way.

QR code Scanning: need to find a specific building? Just scan a QR code on campus and instantly launch navigation to your destination, fully integrated with AR system.

Dynamic Mapping: Google Maps is built right into the app, the complete with personalized building icons, filtering options, and a smart search that adjust to your location and preferences for easy exploring.

Localized Multilingual Support: provides comprehensive localization supporting Arabic, Kurdish Sorani, and English languages, enhancing accessibility and user engagement across diverse user groups.

Feedback and communication: enables efficient communication between students and university administration through dedicated feedback portal, supporting text input, screenshot uploads, and categorized feedback submission.

Limitations Of Research

- The main constraint is to make sure that the AR of the various devices is correct and accurate and is operable
- Optimize the Firestick to improve its real-time optimization and its ability to store a lot of data
- Device compatibility issues and remedies across devices
- Making sure there is a stable internet connection for real-time data
- Battery consumption AR-enabled apps and synchronized real-time data can lead to battery consumption.
- Provide user guidance People unfamiliar with AR technology may find it difficult to use it or not understand it, so it is important to provide guidance and educate the user on how to use it.
- Storage can also store large amounts of data in Firestore.
- Security issues Protecting application data in real-time and with unauthorized access is a challenge if it is not implemented securely and reliably.

Scalable, which makes it possible to grow the application and expand it to more features and different locations, which improves performance and ease of use and access, but makes the ability to control more complex data

Future Work

As the **University Assistant Mobile Application** continues to evolve, several enhancements are planned to further enrich the student experience and expand the system's capabilities. These future developments focus on personalization, accessibility, and advanced technologies to ensure the app remains a vital tool in daily campus life.

1. Voice Assistant Integration

To enhance accessibility and convenience, future versions of the app will include **voice input support** for the AI chatbot. This will allow students to ask questions or request information hands-free, making the app more user-friendly, especially for students on the move.

2. Augmented Reality (AR) Campus Navigation

An **AR-based navigation system** will be implemented to guide users around the campus using their smartphone camera and on-screen 3D directions. This feature will help new students or visitors easily locate buildings, departments, and event venues.

3. Expanded Admin Controls

The admin panel will be extended to support more advanced features, including:

- Role-based user access (e.g., for faculty or event managers)
- In-app announcements and push notifications
- Analytics dashboards for monitoring student engagement

4. **Offline Mode Support**

Plans include the introduction of an offline access feature, enabling users to view static campus maps, building info, and saved schedules even without an internet connection. This will improve reliability for students in low-connectivity areas.

References

[1] Andri, C., Alkawaz, M.H. and Sallow, A.B., 2018. Adoption of mobile augmented reality as a campus tour application. Int. J. Eng. Technol, 7(4), pp.64-69.

[2] Garzón, J., 2021. An overview of twenty-five years of augmented reality in education. Multimodal Technologies and Interaction, 5(7), p.37.

[3] Mohd Nizam, D.N., Shin, L.W., Abdullah Sani, Z.H., Thamrongrat, P. and Tuah, N.M., 2021. An Indoor Navigation Support for the Student Halls of Residence using Augmented Reality: A Design Perspective. Pertanika Journal of Science & Technology, 29(4).

[4] Rajagopal, R.D., Sripriya, A., Shakthi, B., Rathinam, M. and Rajagopal, H., 2024. AR Based Application for Campus Navigation.

[5] Ng, X.H. and Lim, W.N., 2020, October. Design of a mobile augmented reality-based indoor navigation system. In 2020 4th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT) (pp. 1-6). IEEE.

[6] Nordin, N., Markom, M.A. and Suhaimi, F.A., 2021, August. A web-based campus navigation system with mobile augmented reality intervention. In Journal of Physics: Conference Series (Vol. 1997, No. 1, p. 012038). IOP Publishing.

[7] Sukhareva, E., Tomchinskaya, T. and Serov, I., 2021, September. Slam-based indoor navigation in university buildings. In Proceedings of the 31th International Conference on Computer Graphics and Vision (Vol. 2, pp. 611-617).

[8] Qi, W., Wang, W., Han, H., Yu, S., Gu, Y. and Liu, Y., 2024, July. Augmented reality campus navigation for both indoor and outdoor spaces based on ARCore. In Third International Conference on Electronic Information Engineering and Data Processing (EIEDP 2024) (Vol.

13184, pp. 679-685). SPIE.

[9] Putra, E.Y., Kom, M., Wahyudi, A., Nabut, P. and Pelle, A., 2024. Development of a Location-Based Augmented Reality Application for Navigation. CogITo Smart Journal, 10(1), pp.171-181.

[10] Romli, R., Yew, C.C., Ghazali, N.H., Zahri, N.A.H., Amir, A. and Hasnan, R., 2024, May. AR@ campus: Augmented reality (AR) for indoor positioning and navigation apps. In AIP Conference Proceedings (Vol. 3135, No. 1). AIP Publishing.

[11] Mohd Nizam, D.N., Shin, L.W., Abdullah Sani, Z.H., Thamrongrat, P. and Tuah, N.M., 2021. An Indoor Navigation Support for the Student Halls of Residence using Augmented Reality: A Design Perspective. Pertanika Journal of Science & Technology, 29(4).

[12] Romli, R., Razali, A.F., Ghazali, N.H., Hanin, N.A. and Ibrahim, S.Z., 2020, February. Mobile augmented reality (AR) marker-based for indoor library navigation. In IOP Conference Series: Materials Science and Engineering (Vol. 767, No. 1, p. 012062). IOP Publishing.

[13] https://www.britannica.com/technology/augmented-reality