An-Najah National University

Faculty of Engineering & Information Technology

Department of Computer Engineering

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

Graduation Project 1

Dish Dash

Students:

Zain Abubaker

Jana Khammash

Supervisor:

Dr. Samer Arandi

June 9, 2025

Acknowledgement

“ We would like to start by thanking Dr. Samer Arandi, our supervisor, for all his help and support throughout this project. His guidance, advice, and feedback were very important in making our work better. We are really thankful for the time and effort he gave us. We also want to thank our families for their love, care, and constant support. You believed in us from the start and helped us stay strong, even when things got tough. Your love and sacrifices gave us the motivation we needed to keep going. Finally, we are grateful to everyone who helped with this project in any way. Whether it was through advice, assistance, or simply encouragement, your support means a lot, and we truly appreciate it. ” -Jana, Zain

Disclaimer

This report was written by Jana Khammash and Zain Abubaker at the Computer Engineering Department, Faculty of Engineering, An-Najah National University. It has not been altered or corrected, other than editorial corrections, as a result of assessment and it may contain language as well as content errors. The views expressed in it together with any outcomes and recommendations are solely those of the students. An-Najah National University accepts no responsibility or liability for the consequences of this report being used for a purpose other than the purpose for which it was commissioned.

Contents

List of Figures

Chapter 1

Abstract

In today's fast-paced world, meal planning and cooking have become more challenging, especially for individuals looking to maintain a healthy lifestyle while balancing a busy schedule. Many home cooks and food enthusiasts struggle to find reliable sources for personalized recipes, structured meal plans, and efficient grocery management. DishDash is designed to make this entire process seamless, enjoyable, and tailored to individual needs, ensuring that cooking remains a convenient and fulfilling part of daily life. DishDash aims to develop a smart recipe and meal planning platform that offers a seamless and personalized cooking experience. The platform features a user-friendly application and website where users can explore an extensive recipe library, receive AI-powered meal recommendations, and generate automatic grocery lists. Users can search for recipes based on available ingredients, track their nutritional intake, and create weekly meal plans effortlessly. A strong community aspect allows users to share their favorite recipes, rate meals, and participate in engaging cooking challenges and discussions. Additionally, the platform integrates with grocery stores for direct shopping and price comparisons, ensuring convenience and cost-effectiveness. Through research, we found that while many global cooking platforms exist, few offer a localized and personalized approach, and none provide an all-in-one solution combining recipes, meal planning, and grocery automation. Our platform bridges this gap, making cooking more accessible, enjoyable, and efficient for food lovers worldwide.

Chapter 2

Introduction

2.1 General Background

DishDash is a smart and interactive culinary application that merges personalized meal planning, grocery management, AI-powered recipe generation, and social community features into a unified ecosystem. Designed with the goal of improving healthy eating habits, fostering local food networks, and leveraging AI for recipe and nutrition analysis, DishDash empowers users to save, share, and explore meals effortlessly. The system was developed using Flutter for the front-end, Node.js and Express for the back-end, and MongoDB for the database. DishDash also integrates APIs such as Google Maps, Cloudinary, Edamam Nutrition Analysis, and Socket.IO for real-time communication. This report details the project's architecture, development process, implementation, and contributions toward health-aware food management.

2.2 Problem

Modern users struggle to maintain healthy eating habits due to busy lifestyles, lack of personalized guidance, or limited access to local food sources. Additionally, there is a gap in integrating grocery needs, meal planning, dietary preferences, and community engagement in a single application.

2.3 Objectives

- Enable users to search, save, and create custom recipes.
- Recommend meals based on dietary needs, time of day, and preferences.
- Generate personalized grocery lists and allow store-based item tracking.
- Build a community with features like chat, posts, likes, comments, and challenges.

- Provide AI-assisted features like image-to-recipe generation and Pinterest scraping.

2.4 Scope of the Project

DishDash spans across mobile (Flutter) and web platforms, with support for RESTful APIs, real-time notifications, AI integrations, cloud storage, and interactive community features. It includes admin, user, and store dashboards with different roles and privileges.

2.5 Importance

DishDash introduces a scalable and modular approach to food tech solutions, promoting both user health and local food commerce. Unlike isolated recipe or grocery apps, DishDash creates a dynamic, interactive, and intelligent ecosystem for culinary engagement.

2.6 Report Organization

This report presents the development of the DishDash application and includes:

- An introduction to the project goals and user needs.
- The constraints and challenges faced during development.
- A literature review of existing meal planning and nutrition apps.
- The methodology used for system design and implementation.
- An analysis of key features, user interactions, and performance.
- A discussion of limitations and areas for future improvement.
- Final recommendations, references, and supporting tools.

Chapter 3

Constraints And Earlier Course Work

3.1 Constraints

3.1.1 Cost and Budget Constraint

- Lack of real device testing during early development stages led to platform-specific bugs (e.g., image or asset loading issues on mobile vs web).
- Integration with some external services (like Edamam API or Pinterest scraping) required rate-limited keys or server workarounds.
- Image size, rendering consistency, and Cloudinary integration presented early deployment issues.
- Real-time features like chat and notifications needed Socket.IO optimization and testing under multiple user conditions.
- Certain Flutter features were not fully supported on the web, delaying the admin dashboard deployment.
- Implementing multilingual support and AI image analysis (image-to-recipe) required external libraries and fine-tuning.

3.1.2 Limited Resources

We were unable to thoroughly examine and execute every facet of the DishDash project because of time and resource limitations. However, we concentrated our efforts on developing a seamless, effective, and user-friendly application by closely examining its fundamental ideas and design objectives. We put an emphasis on creativity and useful navigation in spite of the lack of substantial resources, and the outcome is a cutting-edge platform that shows our capacity to

create significant solutions under pressure.

3.1.3 Time Duration Constraint

The lengthy and intricate project schedule made time management during DishDash's construction very difficult. Every task in DishDash needs to be completed precisely and with great care. Therefore, in order to efficiently handle project complexity and time constraints, our organization gave strategic planning and core functions top priority. In light of these constraints, we developed DishDash's core components as rapidly as possible by optimizing efficiency while maintaining high standards of quality and accommodating a variety of functionality.

3.2 Earlier Course Work

- **Advanced Web Development:** Provided knowledge of various frameworks, WebSockets, and techniques to connect backend APIs with frontend frameworks, crucial for real-time features and seamless integration.
- **Web Programming:** Enabled the development of the admin and store dashboards using Dart and Flutter Web.
- **Database Systems:** Essential in designing the MongoDB-based schema to manage users, recipes, stores, and orders efficiently.
- **Artificial Intelligence:** Supported the implementation of the recommendation system and AI-based recipe generation.
- **Software Engineering:** Taught modular design, user stories, and iterative testing, all applied during planning and feature development.
- **Operating Systems And Networks:** Contributed to backend communication, user session management, and real-time updates using WebSockets.

Chapter 4

Literature Review

Digital meal planning and food management applications have become indispensable tools for modern users due to the quick development of mobile technologies and the increased awareness of nutrition and health. By combining multiple domains like recipe recommendations, grocery organization, personalized nutrition, and social engagement, the DishDash platform supports this trend. This review of the literature looks at recent studies and frameworks that guide DishDash's development and design, pointing out opportunities and gaps that the platform aims to fill.

Studies like [1] and [2] highlight the beneficial effects of digital tools in encouraging healthy eating habits. Mobile apps that provide individualized meal suggestions according to calorie requirements, dietary preferences, and allergies have demonstrated quantifiable advantages in helping users make decisions. In order to provide individualized recommendations, DishDash expands on this research by integrating a recommendation engine that takes into account time of day, favorite recipes, saved meals, dietary restrictions, and BMI.

Current meal-planning applications like Lifesum, Yazio, and MyFitnessPal place a strong emphasis on basic meal logging and calorie tracking. Although these apps are good at tracking intake, they frequently don't have deeper integrations with real-time store data and grocery management. This limitation is addressed by DishDash, which creates a smooth transition between planning, shopping, and eating by automatically creating grocery lists from planned meals and enabling users to compare store prices, check availability, and place orders.

Tasty, Whisk, Yummly, and other social and community-based food platforms promote user interaction through recipe sharing and engagement, but they frequently fall short in providing a meaningfully personalized experience. To create a more curated and pertinent feed, DishDash, on the other hand, blends user data with social features like recipe likes, comments, follows, and real-time chat. Additionally, DishDash facilitates a more dynamic, diverse, and inclusive recipe ecosystem by allowing user-generated content and recipes.

DishDash adheres to contemporary full-stack architecture principles from a technical stand-

point. Node.js and MongoDB facilitate quick, scalable backend operations, while Flutter for cross-platform development guarantees wide device compatibility and performance. Prior studies on mobile health (mHealth) application design [3] have emphasized the significance of data privacy, real-time responsiveness, and usability—features that DishDash incorporates through secure API practices, responsive user interfaces, and real-time Socket.IO integration.

In conclusion, a large body of current research and technological applications inform the DishDash platform, which sets itself apart with its integrative methodology. DishDash offers consumers a smart, connected, and customized food experience by fusing health-focused features with AI-powered improvements, store integration, and community engagement. This closes a growing gap in the digital nutrition and lifestyle market.

Chapter 5

Methodology

5.1 Architectural Design

5.1.1 Database Design and Management

Choosing a NoSQL database (MongoDB) was ideal for DishDash due to its flexibility in handling diverse and dynamic data structures. Its schema-less nature allows rapid development and scalability—important for a platform managing various user roles, recipes, preferences, and orders. MongoDB's horizontal scaling capabilities also ensure smooth performance as user and data volume grow. For real-time data like notifications and messages, Firebase was used to store and retrieve content efficiently.

Database Collections:

- **Users Collection:** Stores name, email, password, avatar, allergies, preferences, grocery list, and saved recipes.
- **Recipes Collection:** Holds both system and user-created recipes with metadata such as ingredients, calories, prep time, tags, and author details.
- **MealPlans Collection:** Tracks daily meal schedules, selected recipes, completion status, and calorie data.
- **Stores Collection:** Contains store profiles, ratings, locations, contact info, and available items.
- **Orders Collection:** Manages user orders including items, quantities, order status, and associated store and user info.
- **Notifications Collection:** Logs events like likes, messages, follows, purchases, and admin announcements.
- **Comments Collection:** Records user comments on posts or recipes with timestamps and user references.
- **Chats Collection:** Stores real-time messages, both text and image, with sender and

receiver metadata.

- **Courses Collection:** Contains educational content including lessons, ratings, and chef details.
- **Challenges Collection:** Includes admin-defined challenges with rules, rewards, timelines, and user submissions.

5.2 Tools, Programming Languages, APIs, Technologies

5.2.1 Programming Languages and Frameworks

DishDash was developed using **Flutter**, allowing us to build a unified codebase for both mobile and web platforms. This reduced development time and ensured a consistent user experience. The backend was built using **Node.js** with **Express.js**, enabling efficient API handling and asynchronous operations. Real-time communication was powered by **Socket.IO**, and **MongoDB** served as the main NoSQL database.

5.2.2 Tools

- **Visual Studio Code:** Used as the main development environment for both frontend and backend.
- **Postman:** Assisted in testing and debugging API endpoints.
- **Android Studio:** Provided virtual devices for mobile app testing.
- **GitHub:** Used for version control and collaborative development.
- **Figma:** Created wireframes and interactive mockups for UI planning.
- **Canva:** Designed visual assets and presentation slides for demos and documentation.

5.2.3 APIs

AI and Intelligence:

- **OpenAI API:** Used for AI recipe generation from image or input ingredients.
- **Edamam API:** Provides nutrition analysis of ingredients and recipes.

Localization, Media, and Translation:

- **Google Maps API:** Enables store location, distance calculation, and directions.
- **Google Translate API:** Translates recipe content (e.g., Arabic to English).
- **Unsplash API:** Supplies high-quality food images during recipe creation.
- **Cloudinary API:** Hosts and retrieves recipe/user images.
- **Base64 Image Encoding:** Handles image uploads for mobile/web platforms.
- **WhatsApp URL Scheme:** Allows direct communication with stores.

Real-Time and Social Features:

- **Socket.IO:** Powers real-time messaging and notifications.

Scraping and External Data Sources:

- **Pinterest Scraper (Puppeteer):** Scrapes recipe title, ingredients, image, and description from Pinterest links.

Custom Backend Services:

- **DishDash RESTful API:** Built using Node.js and Express.js to handle:
  - User registration, login, OTP reset
  - Recipe creation, filtering, saving
  - Grocery and meal planning
  - Orders and store management
  - Notifications, comments, likes, follows, chat
  - Admin controls and challenge creation

5.3 Implementation

5.3.1 Frameworks

Flutter was used to build a responsive and scalable frontend for mobile and web. Node.js and Express.js handled backend services and APIs. MongoDB enabled flexible data modeling, while Socket.IO supported real-time interactivity.

5.3.2 Application Features

1. **User Registration & Login:**
   Users and stores can register and log in securely, with location selection via map and OTP-based password reset for users.
2. **Personalized Survey:**
   After registration, users fill out a survey with allergies, diet, preferences, weight, and height, used mainly to tailor recommendations.
3. **AI Features:**
   Users can generate recipes using OpenAI. They can choose preferences in diet, mealtime, calories, number of servings, etc. The AI generator provides the recipe's details and ingredients and generates an image using the DALLE model. Users can also upload an image to generate a recipe from it or input ingredients to create a recipe based on available items.
4. **HomeScreen Recommendations:**
   Displays smart recipe suggestions based on user activity (likes, saves, follows) and the user's survey and preferences.

5. **Popular Recipes Feed:**

   Showcases top-rated recipes from users and the system. Includes author info and ratings. Users can save any recipe, and saving triggers an allergy alert if ingredients match known allergies.

6. **Recipe Search & Filtering:**

   Users can search recipes by name or filter by ingredients, tags, diet, mealtime, calories, difficulty, and prep time.

7. **Meal Planner:**

   Allows users to add recipes to a daily schedule, edit them, mark meals as done, and rate them. Calories are tracked and synced with the CaloryScore screen.

8. **Grocery List Management:**

   Automatically generates a grocery list from the meal plan. Users can mark items as purchased and check store availability.

9. **Store Integration:**

   Users can view stores, filter by nearby or top-rated, and access profiles with available items, WhatsApp contact, and Google Maps directions.

10. **Orders & Cart System:**

    Users can add items to a cart, choose delivery or pickup, and place an order with simulated payment. The system compares cart prices across stores, marks items as available after ordering, notifies stores, and updates users with order status changes.

11. **Community Feed:**

    Displays user-shared recipes, filtered by all users or followed users. Users can like, save, comment, follow others, view profiles, and start chats.

12. **Real-Time Chat:**

    Users can chat directly with followed users using a draggable chat modal. Supports text and image messages with real-time updates via Socket.IO.

13. **Notifications System:**

    Users receive notifications for likes, comments, follows, messages, challenge announcements with winners and scores, and admin messages. Stores receive notifications for ratings and orders.

14. **Challenges:**

    Admin-created challenges appear in the Community section. Challenges can be categorized (e.g., meal planning, recipe creation). Users can join active challenges and submit progress to the admin.

15. **Courses:**

    A section for cooking courses with lesson videos, ratings, and auto-split lesson capabilities. Admins can import lessons for various chefs.

16. **Profile Management:**

    Users can view and edit their profile, see saved and custom recipes, followers/following, and app statistics. The profile also shows joined challenges, orders, calorie scores, and survey updates.

17. **Custom Recipe Creation:**

    Users can create and update recipes with images, ingredients, tags, prep time, visibility (public/private), etc. AI can analyze calories, and fields can be translated to Arabic.

18. **Pinterest Scraper:**

    Users can enter a Pinterest recipe link to auto-fill a recipe form with scraped data and an image. The recipe is saved as a custom recipe, allowing future edits.

19. **Recipe Info and Details:**

    When clicking a recipe card, users view detailed steps transformed by AI. Users can translate descriptions, ingredients, and instructions, and AI provides nutrition analysis (protein, fats, etc.).

20. **Admin & Store Dashboards (Web and Mobile):**

    Admins can view app statistics, manage user/store profiles, and oversee recipes. Admins can also create challenges, assign winners, and post announcements. Stores can manage items, view orders, analyze stats, and receive notifications both on web and mobile.

21. **User's Website:**

    Users can log in via the website and access most DishDash features, including Community, Recipes, and more.

5.3.3 Mobile Application

1. Welcome Screen: The figure below shows the welcome screen that allows both users and stores to either Login or register to the application.

2. Login and Sign-up Page: You can see the Login page and the process of registering, including the map modal for choosing location.

3. Forgot Password Page: When a user or a store forgets their password, this screen helps them recover access to their account by sending an OTP code to the email registered, which expires in 60 seconds.

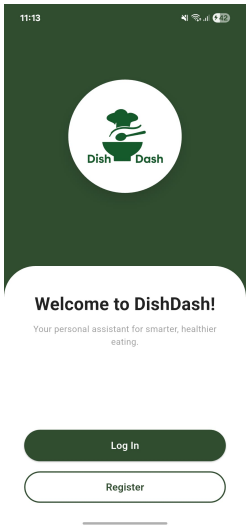

Figure 5.1: Welcome screen.

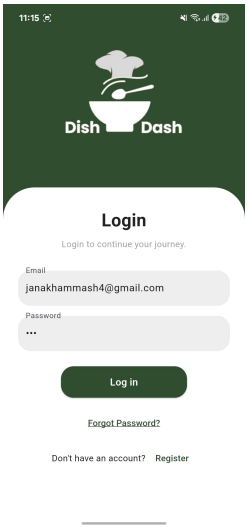

Figure 5.2: Login screen.

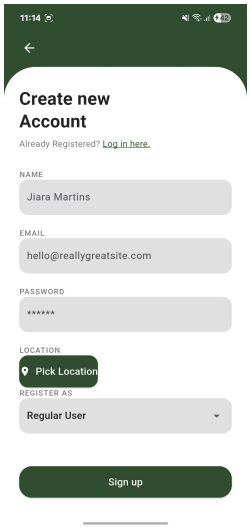

Figure 5.3: Register screen.

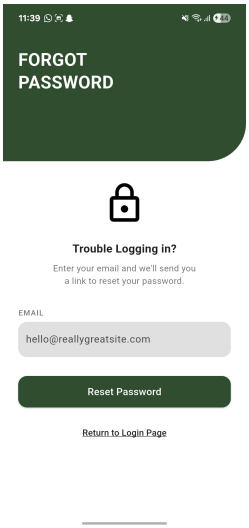

Figure 5.4: Forgot Password screen.

5.3.3.1 User Side

1. Home Screen: These figures Show all functionalities of DishDash's home screen that were mentioned earlier, such as filtering searching and allergy alerts, with the recommendations system and servery recipes.

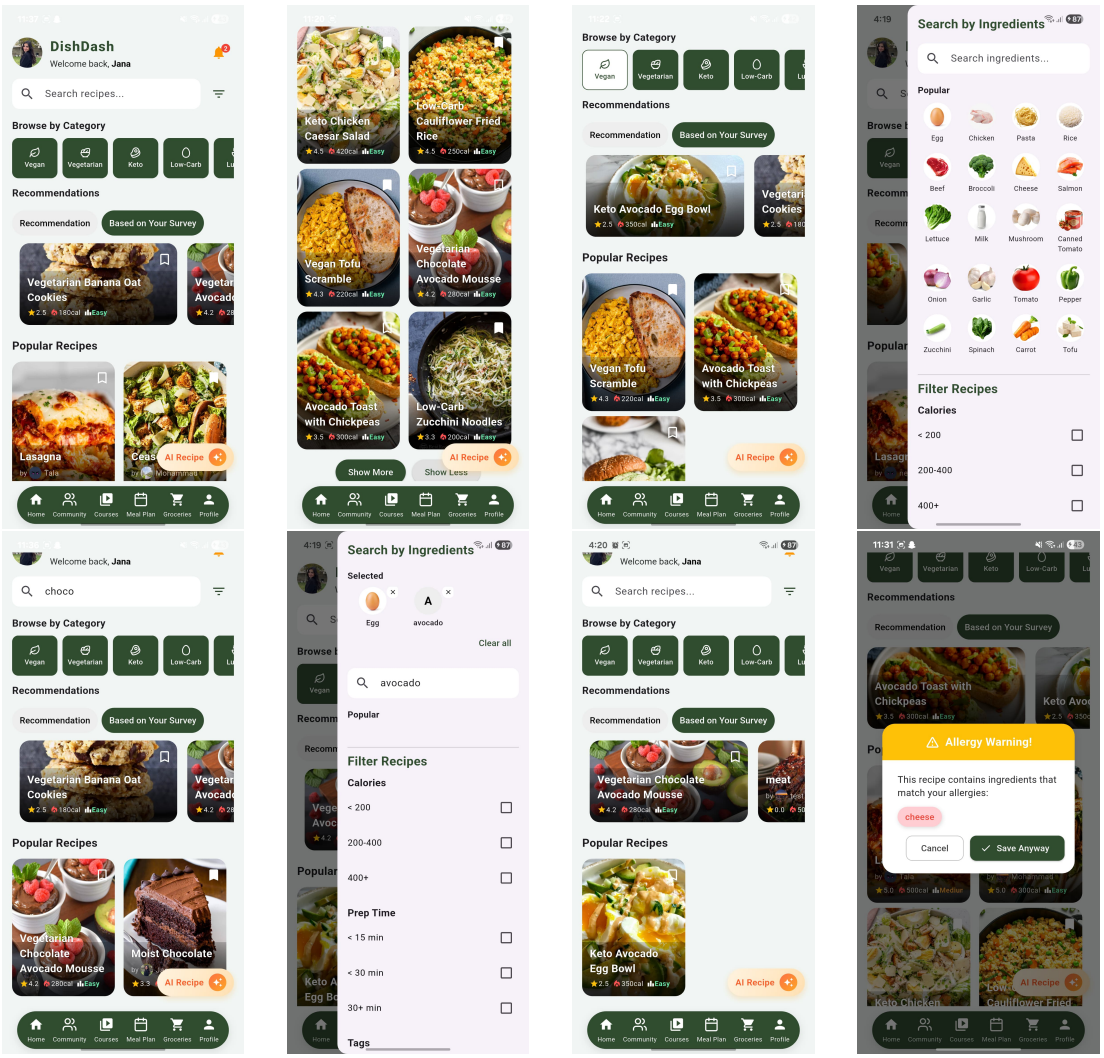

Figure 5.5: Home And Filtering screens.

2. Recipe details Screen: This screen displays the recipe's details, with translation, Nutrition Analysis and AI steps features.

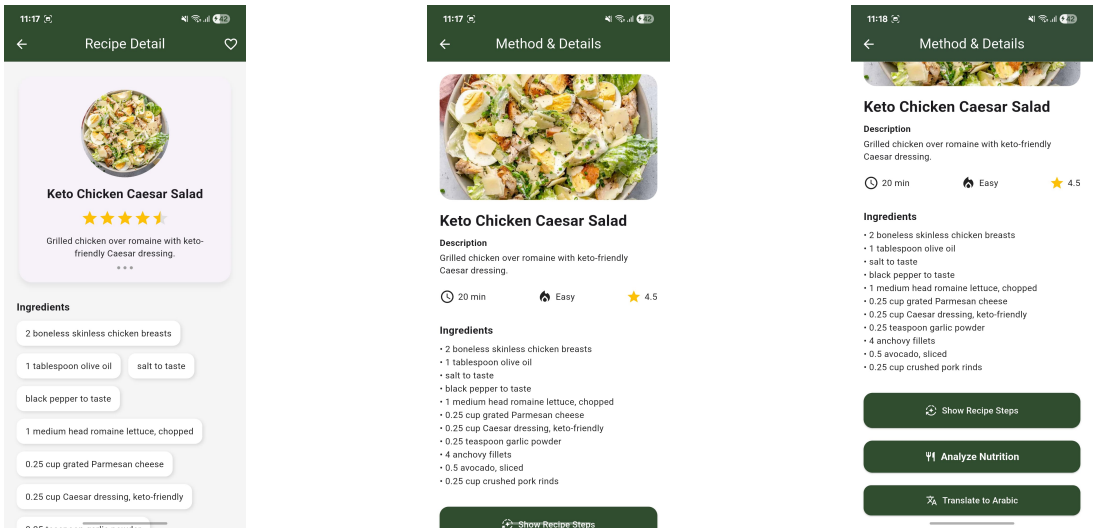

Figure 5.6: Recipe Info Screens.

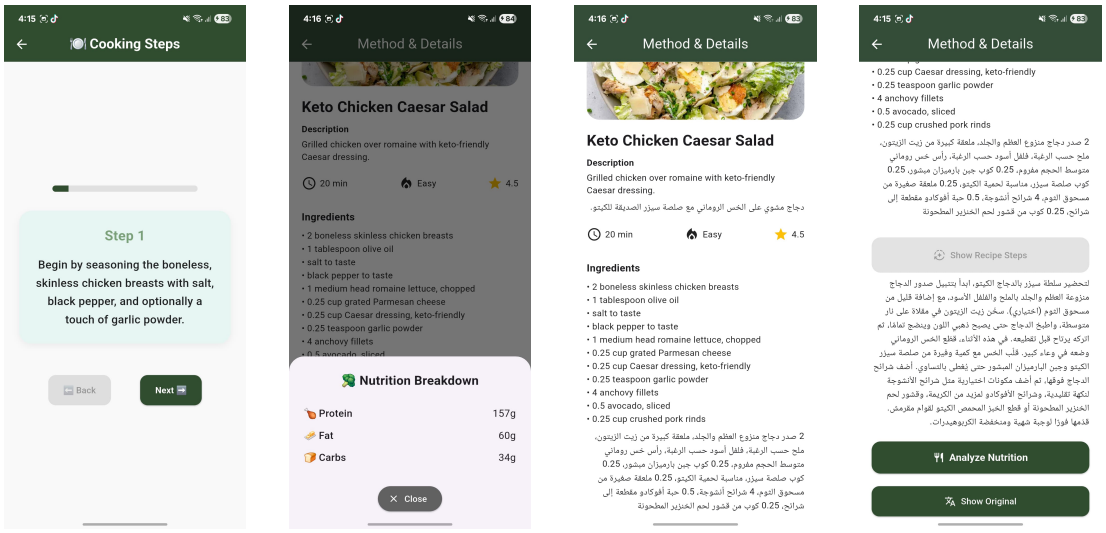

Figure 5.7: AI Analyzing Steps, Nutrition, And Translation Screens.

3. Community Screen: Full access to all user's posts and their profiles in addition to interacting with the posts (Liking, commenting and saving). Admin's challenges and the ability to join them, Stores profile with filtering, locations and items.

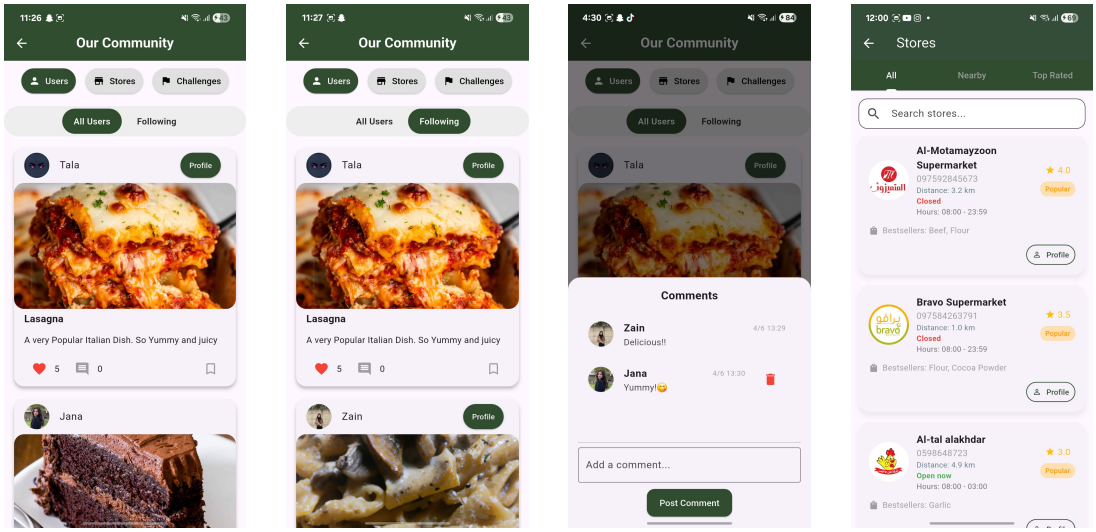

Figure 5.8: Community Screen.

4. User's Profile: This page mainly allows users to follow and message each others.

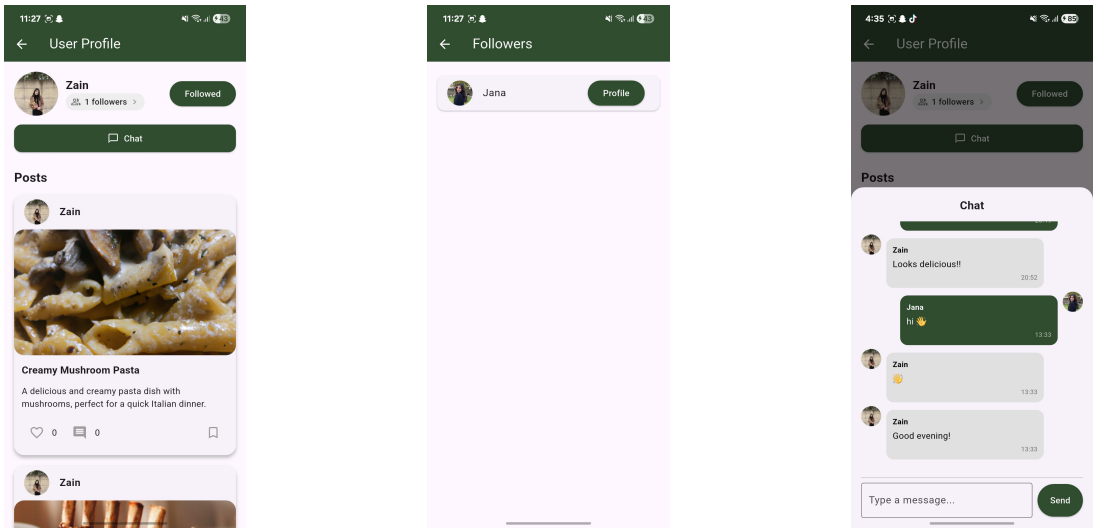

Figure 5.9: User's Profile screen.

5. Store's Profile: Shows the store's categorized menu and items, opening and closing status with a countdown, Google Maps directions. Allowing users to rate them and contact them via WhatsApp.

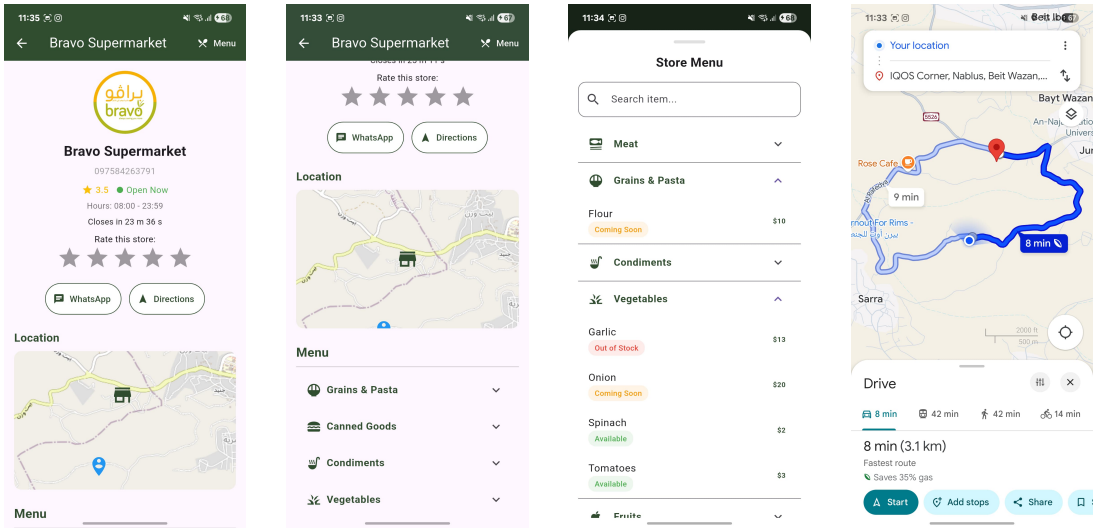

Figure 5.10: Store's Profile Screens.

6. Courses Screen: This screen displays cooking lessons that were added by the Admin. It divides the courses into multiple lessons for a better user experience. Each course has a title, chef, and description. Users can rate courses.

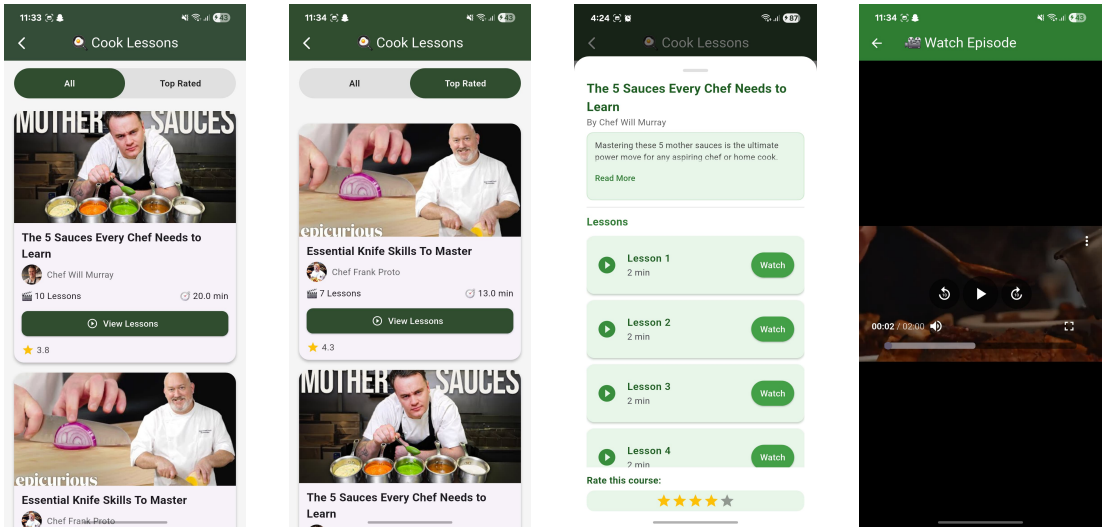

Figure 5.11: Courses Lessons Screens.

7. Meal Plans Screen: This is one of the main features of DishDash, where users can add a meal plan by choosing one of the recipes they saved or customized. Choosing the date of the plan. They can edit the date and mark the plans as Done, which allows them to rate the recipe.

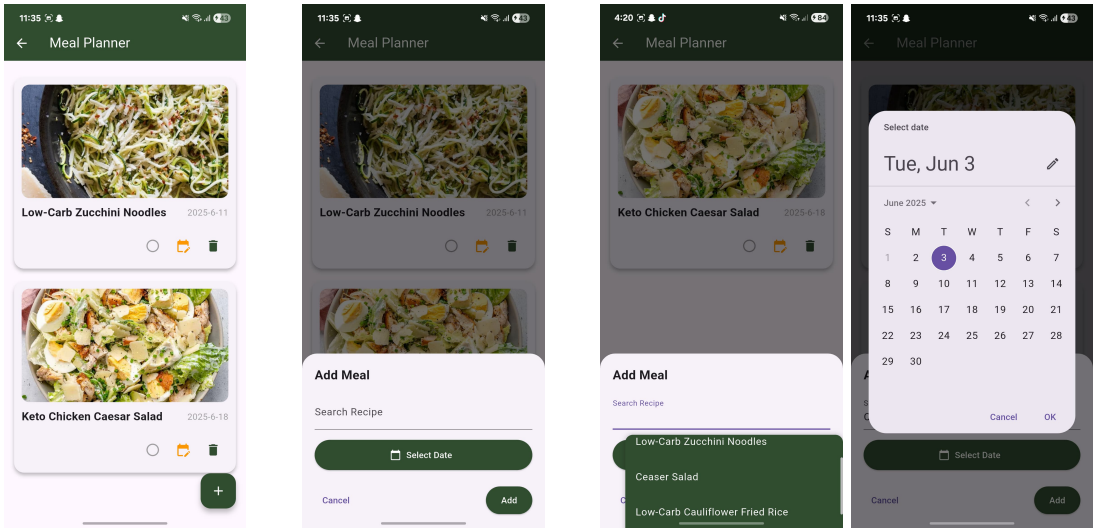

Figure 5.12: Meal Planning Screen.

8. Groceries Screen: Shows a list of ingredients that are autofilled after assigning a meal plan. Users can view the stores that includes the ingredient in their menu, showing the price of it. Users can pick multiple ingredients and add them to the cart which will allow them to order. Stores that have all ingredients in the cart will appear so the user can compare the overall price. After picking the store, they can choose their delivery method and enter their card information. Ingredients after ordering will be listed as "Available Ingredients", moved to the side bar that includes the user's available ingredients.

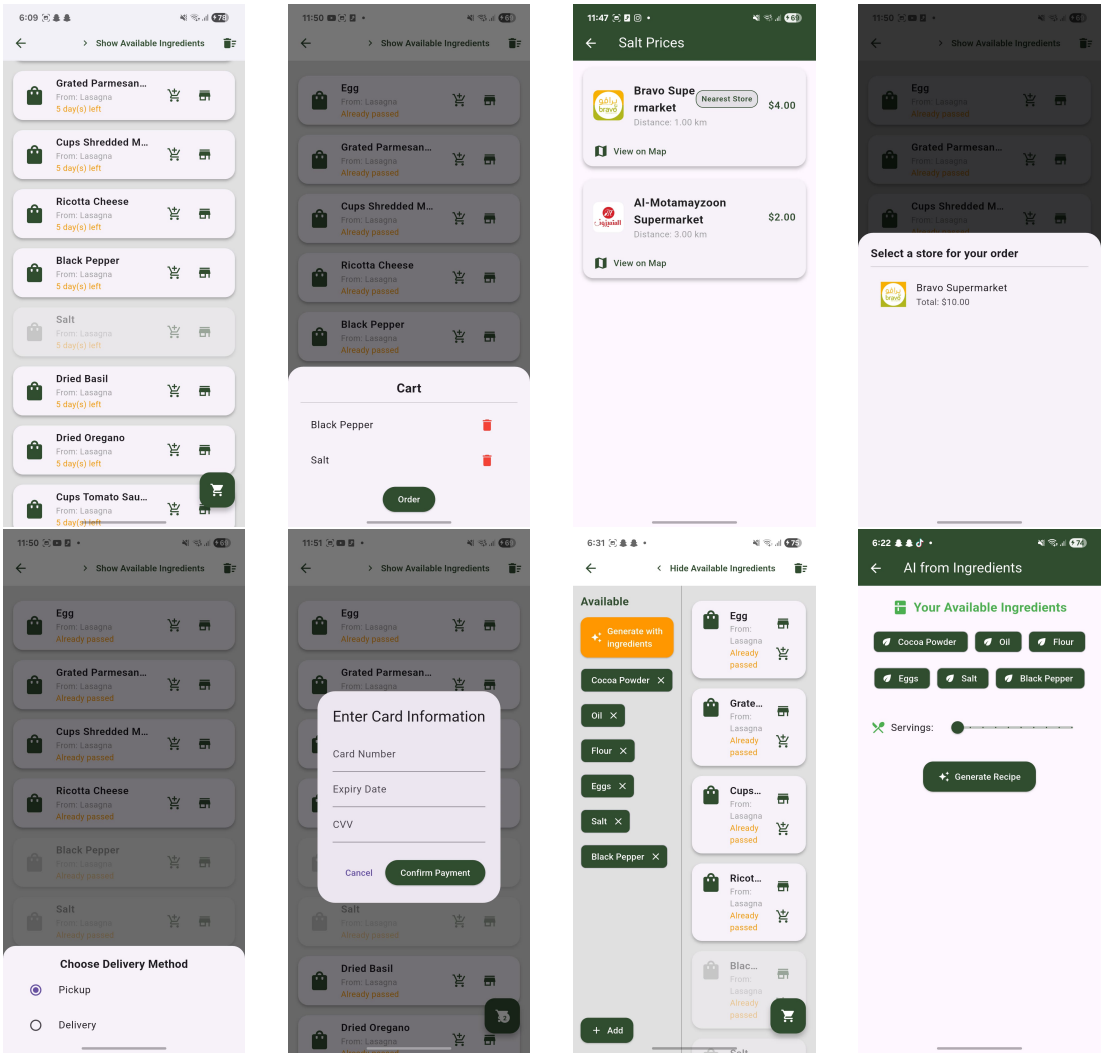

Figure 5.13: Grocery Screen with Cart and AI Available Ingredients Feature.

9. AI Features Screen: A button from the Home Screen navigates to the AI features screen. A Recipe Generator: Allows users to choose their preferences in the recipe they want to generate(diet, mealtime, calories, number of servings..etc). It generates the recipe's details with an image. Users can either save the recipe or regenerate it. Image-to-Recipe generator is DishDash's second AI feature that allows users to upload an image. The AI will process it and illustrate the recipe's details. Users can also save the recipe.

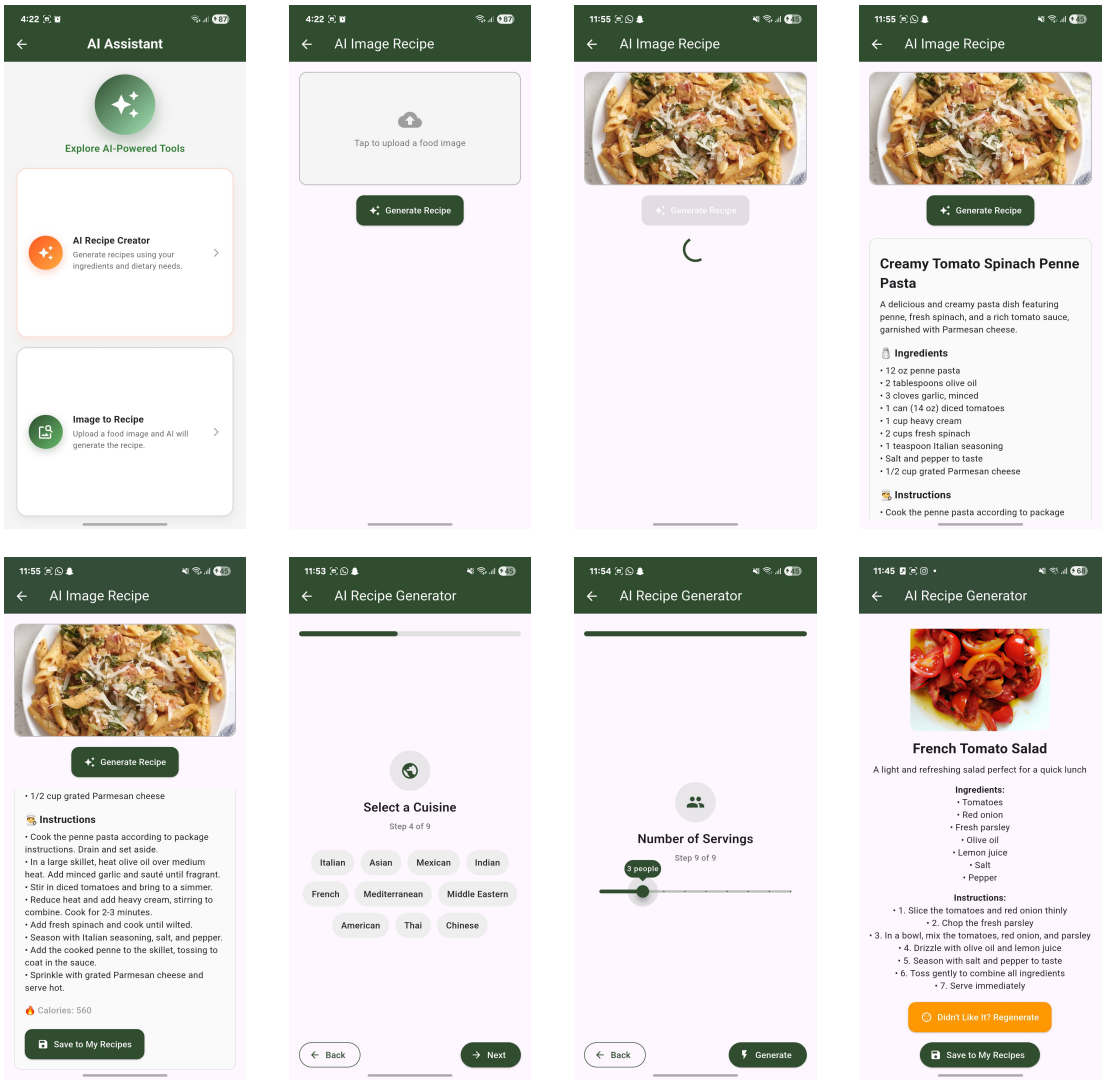

Figure 5.14: AI Features and Recipe Generator Screens.

10. Profile Screen: This screen is for users to manage their profile, upload an avatar, edit their info, view following and followers list, update on their survey, and other management screens. On the bottom center of the screen there is a Pinterest logo button that gives the user the ability to import a recipe from Pinterest and save it to his recipes.

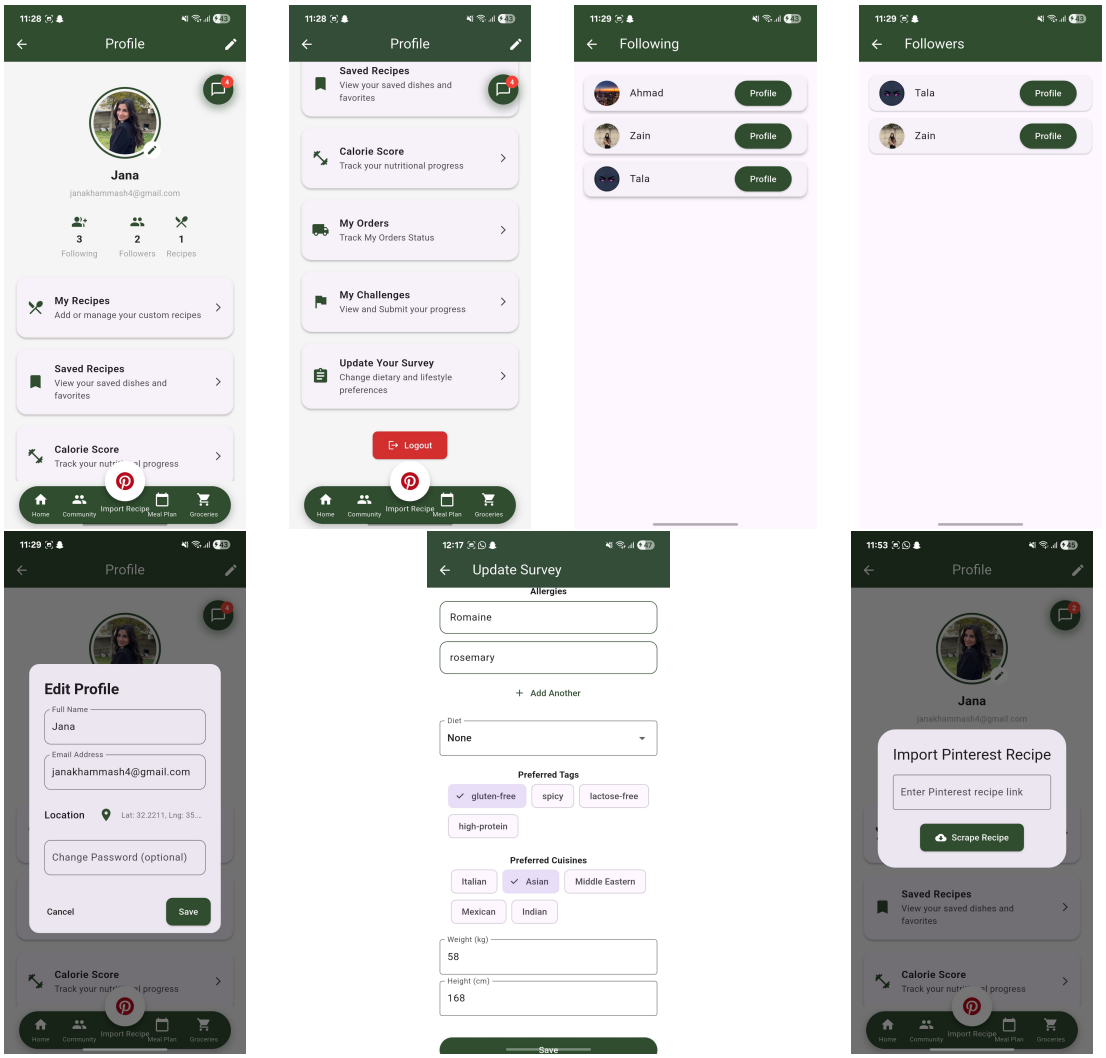

Figure 5.15: Users Profile Screen.

11. My Recipes Screen: This screen shows all user-customized recipes (public, and private ones) as well as AI generated recipes for each user. The button at the end of the screen is for users to customize their own recipes, filling its details, and using AI to translate and analyze calories upon their input.

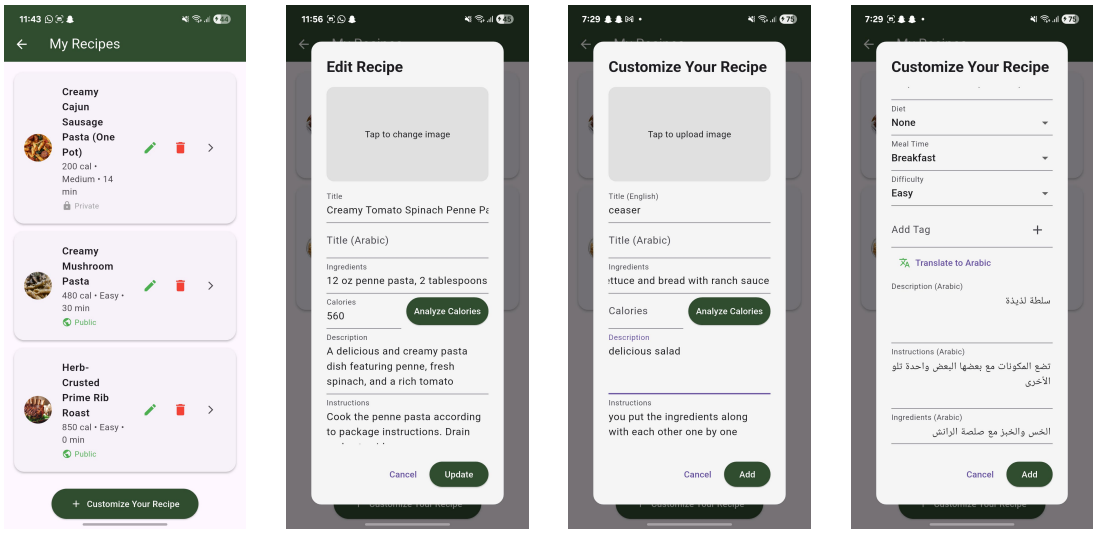

Figure 5.16: My Recipes Screen.

12. Saved Recipes Screen: A screen that shows all recipes that the user saves and allowing them to un-save them.

Figure 5.17: Saved Recipes Screen.

13. My Challenges Screen: This screen displays all challenges that the user joined in the community, they can submit their progress to the admin

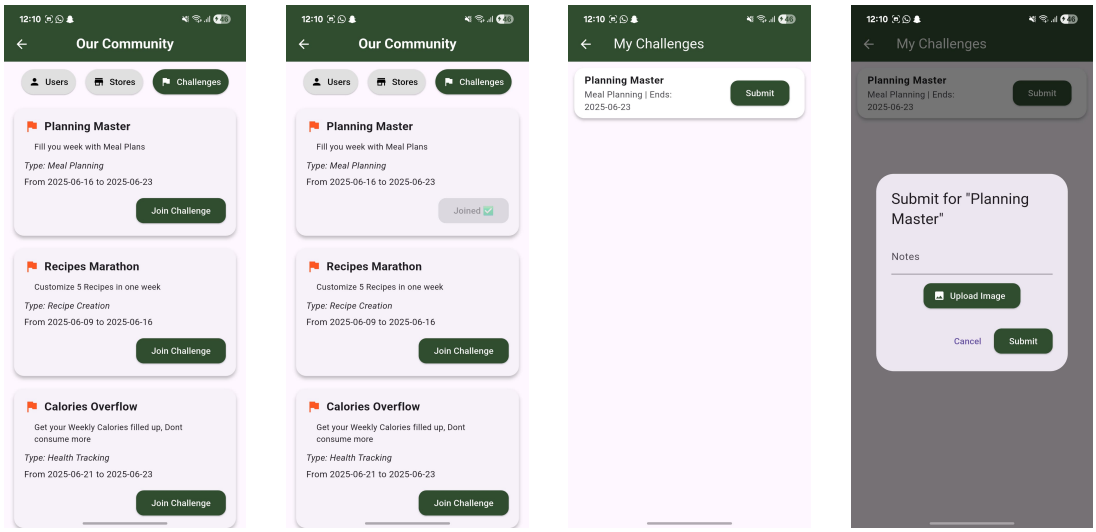

Figure 5.18: Users Notifications Screen.

14. My Orders Screen: Users can view their orders, they can see the directions using Google Maps with an estimated time of delivery. Every order has a badge to show the order status, it changes whenever the store updates on the order's status.

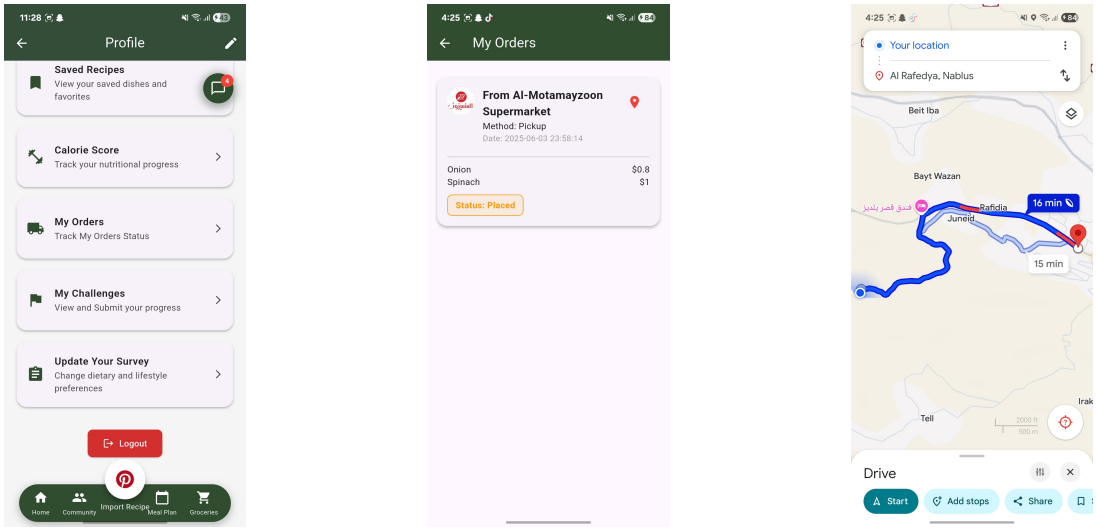

Figure 5.19: My Orders Screen.

15. Calorie Score Screen: This Screen shows the calorie intakes of users, it changes whenever the user mark a meal plan as done. It allows the user to enter a weekly target of calories to display statistics and motivational Labels.

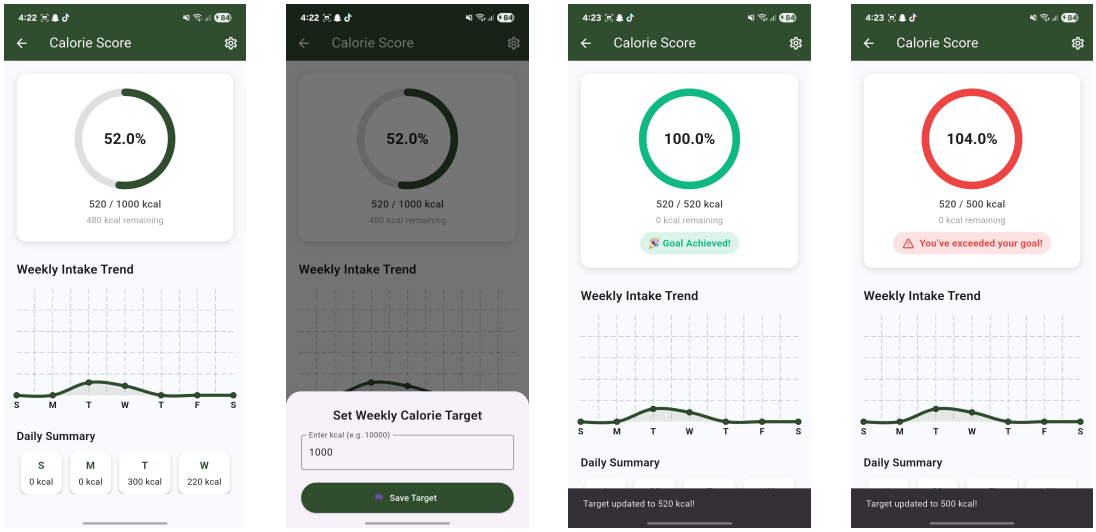

Figure 5.20: Calorie Score Screen.

16. Chatting Screen: On top of the Profile Screen you can see a floating chatting icon with a label of the count of unread messages. This button navigates to the chatting screen where the user can view their chats. online and offline users, followed users, and Chats between them. Users can send Real-Time messages and pictures.

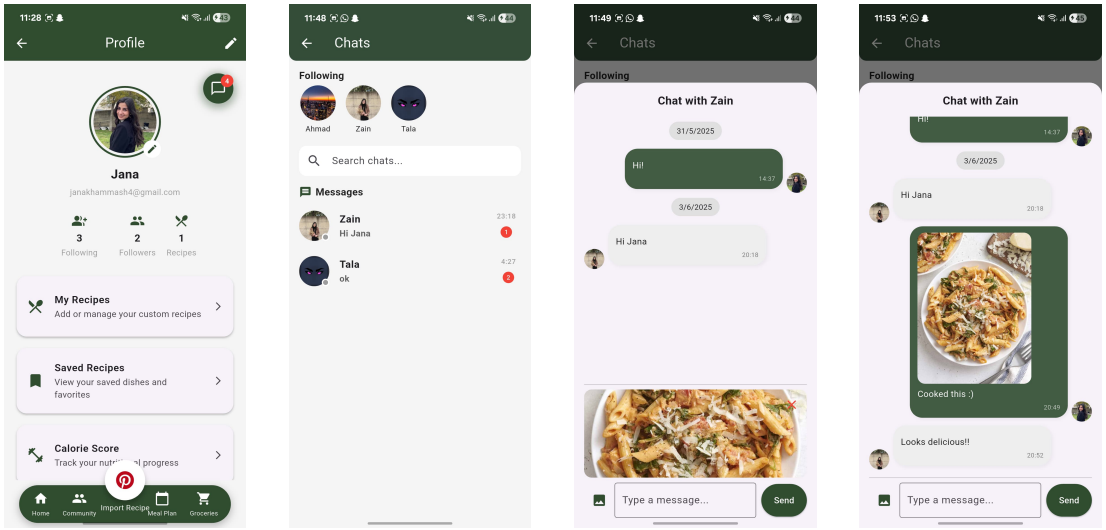

Figure 5.21: Chat Screen.

17. Notifications Screen: An icon button on top of the Home Screen with a label of the count of unviewed notifications. The notification screen displays all notifications the users receive. Filtered by All and Unread.

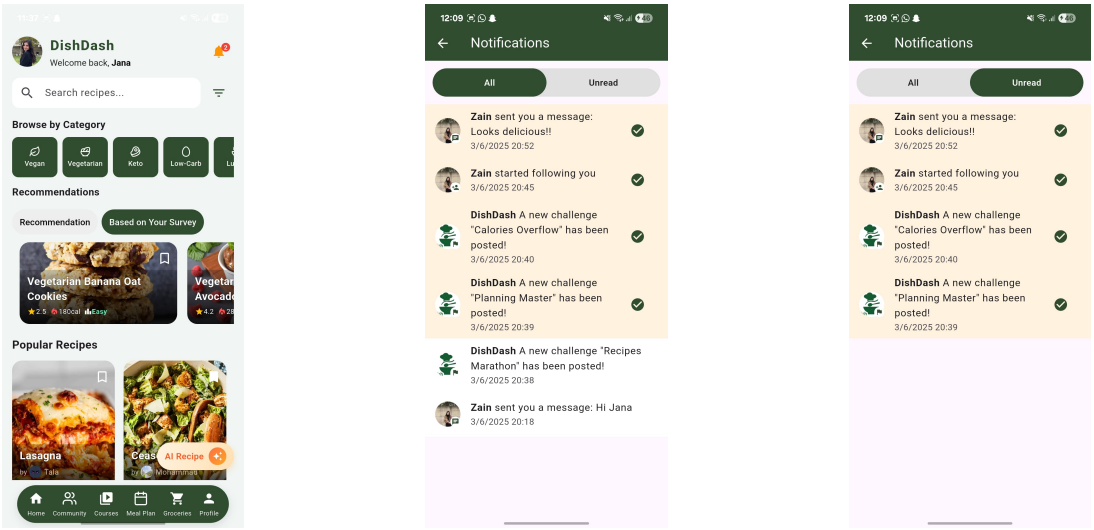

Figure 5.22: Users Notification screen App.

5.3.3.2 Store Side

1. Store Dashboard Screen: This screen gives the store owner statistics of the user activity in its store. It gives him the ability to change its store profile picture and shows the total purchases in its store, total items, and average of ratings given by the users. The store owner can see the users that rate it and the purchases they make. The items button gives the store access to the menu of the store with the items and their status (available, will be available soon, out of stock).

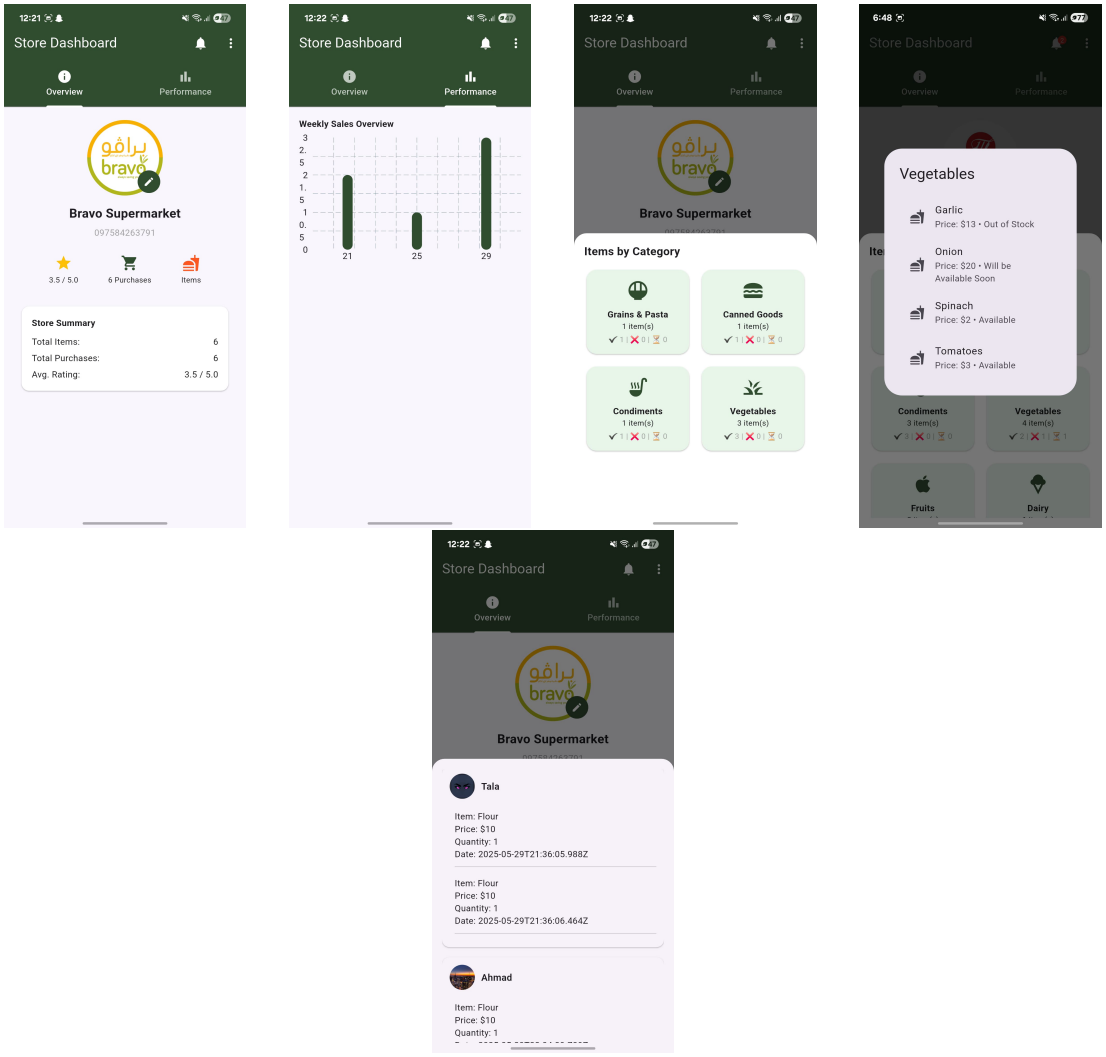

Figure 5.23: Store Dashboard screen App.

2. Store Notifications Screen: On the top of the store dashboard screen there is a notifications icon button with the unread notifications count. This screen displays all the notifications that come from the admin or the user whenever he orders an item/items from this store or if a user rates the store so it notifies him.

Figure 5.24: Store Notifications Screen.

5.3.3.3 Admin Side

1. Admin Dashboard Screen: This screen gives the admin statistics of the whole application activity. It shows the total users, stores, and recipes in the app. With charts that show the distribution of each. It also displays the recent users and stores in the application. At the top of the screen there is an announcement icon button that gives admin the ability to announce sth or send a notification for all the users and stores within the application.

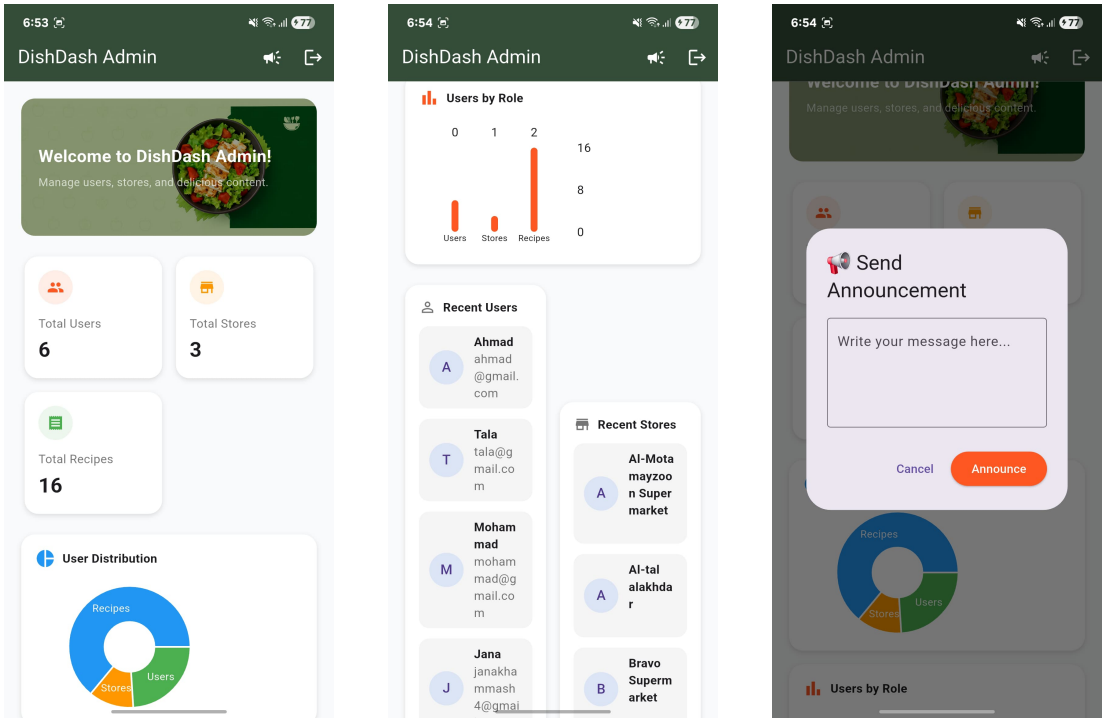

Figure 5.25: Admin Dashboard Screen.

5.3.4 Web Application

This is the Login Screen for Users, Stores and Admin in DishDash's Website

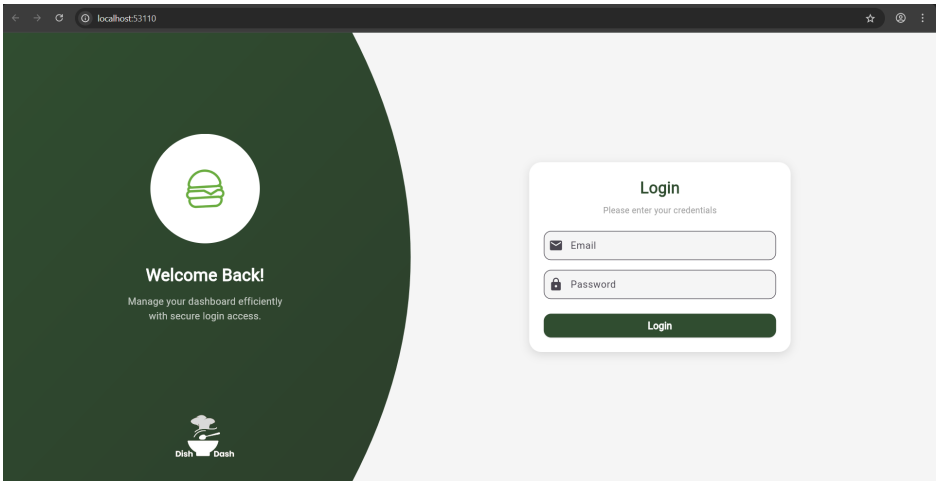

Figure 5.26: Website's Login Screen

5.3.4.1 User Side

The user's Website includes most of the Application features, excluding AI features and Meal planning.

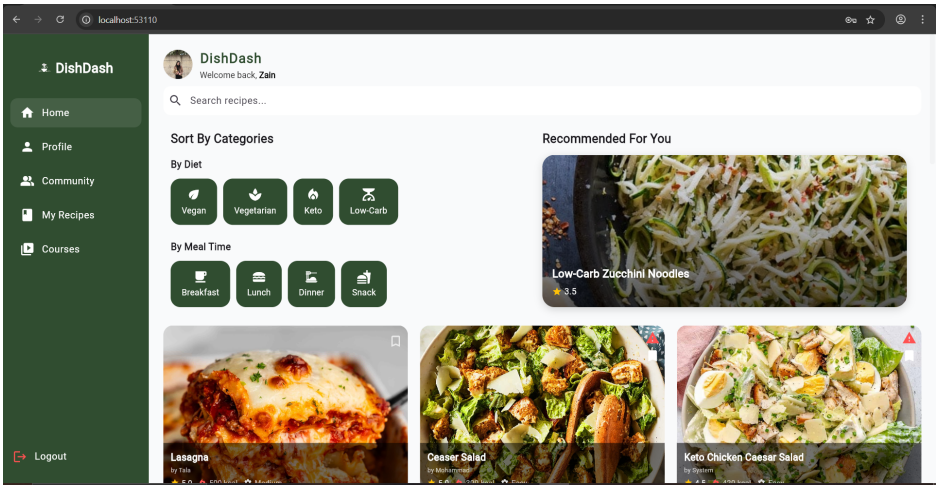

Figure 5.27: HomeScreen in User's Website

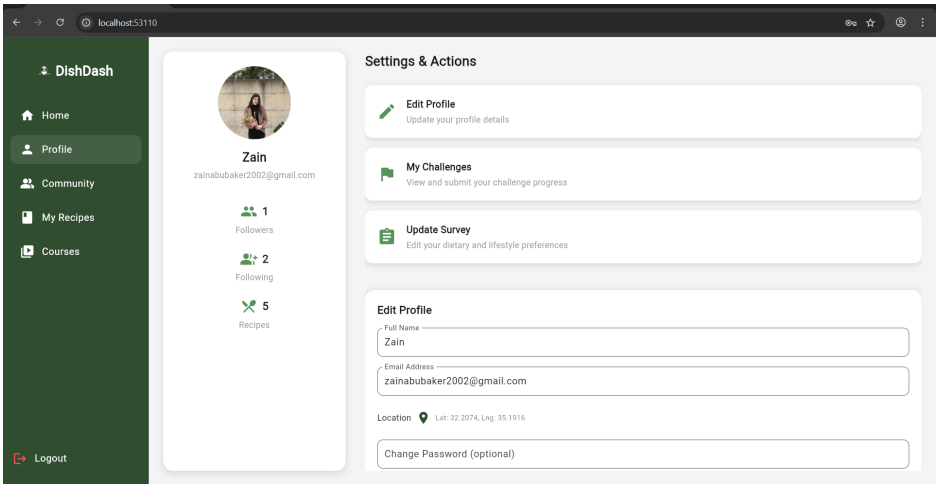

Figure 5.28: Profile in User's Website

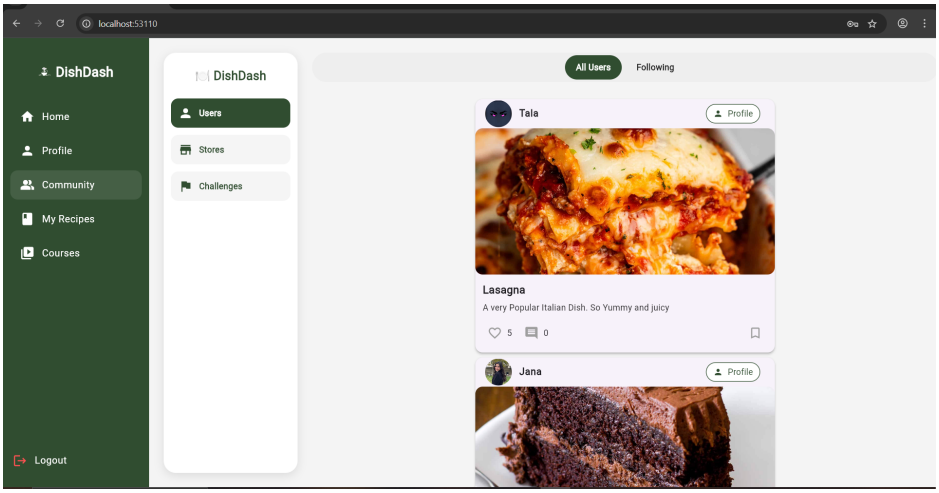

Figure 5.29: Community in User's Website

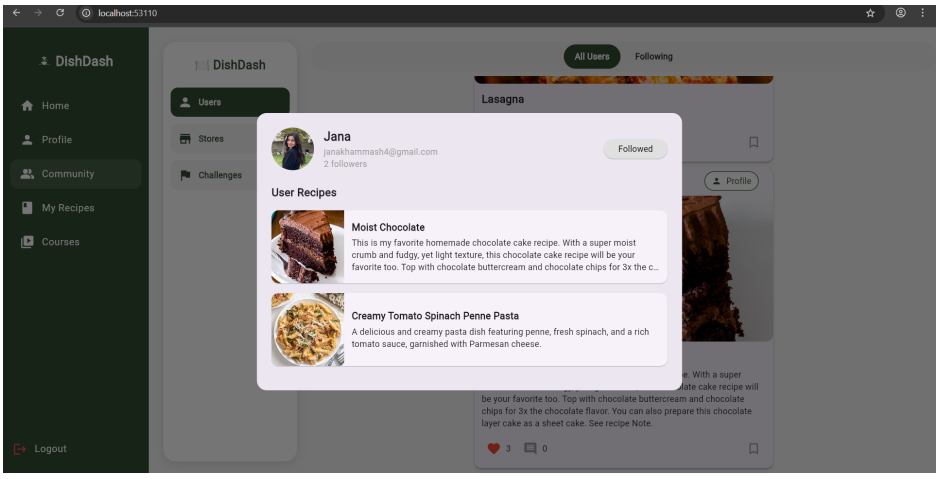

Figure 5.30: Users' Profile in Community Screen

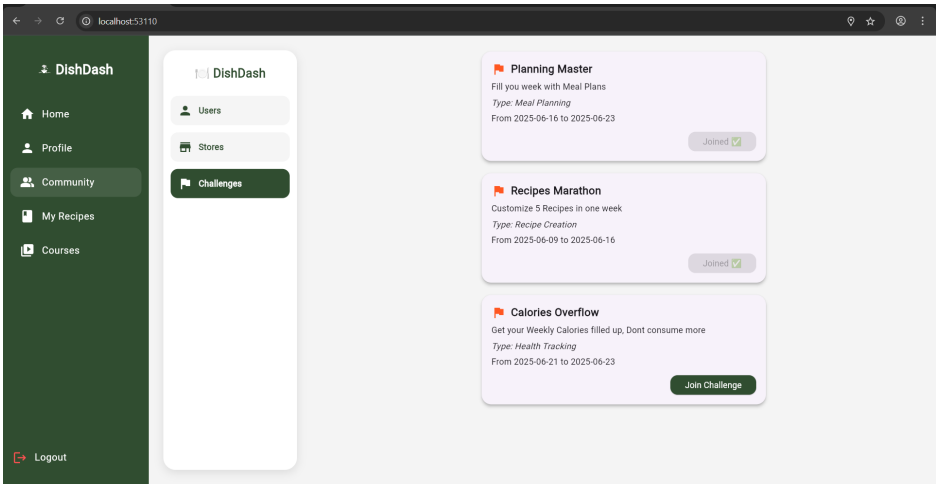

Figure 5.31: Challenges in Community Screen

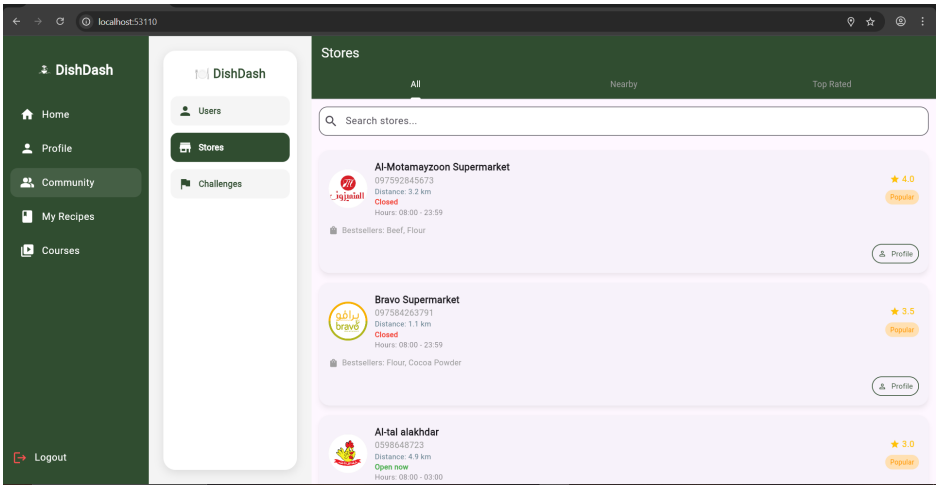

Figure 5.32: Stores in Community Screen

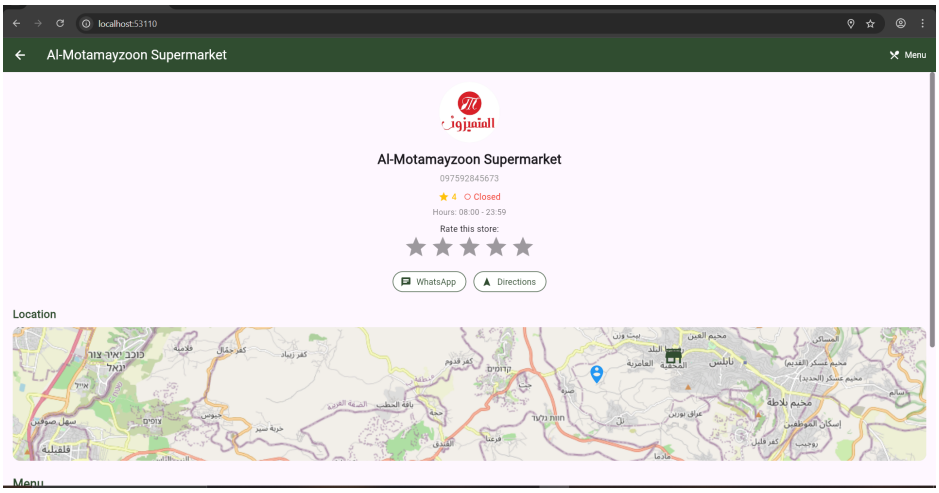

Figure 5.33: Stores' Profile

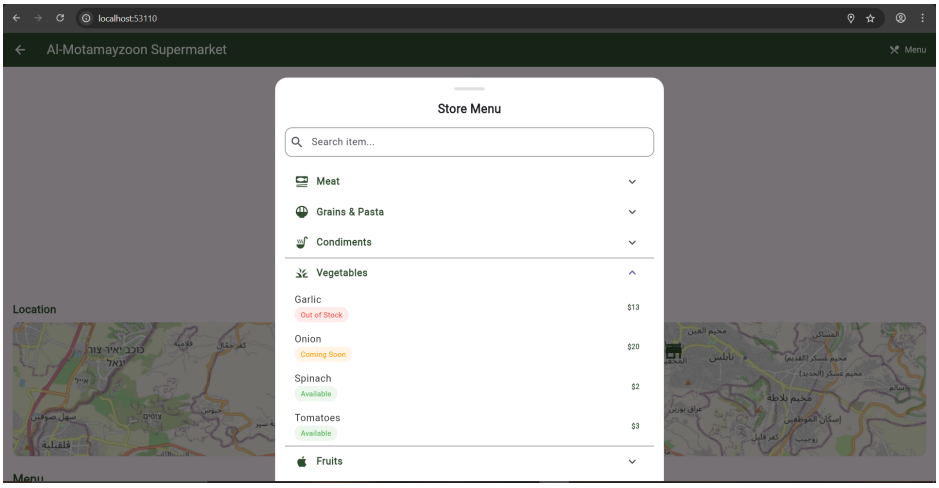

Figure 5.34: Stores' menu in Stores' Profile

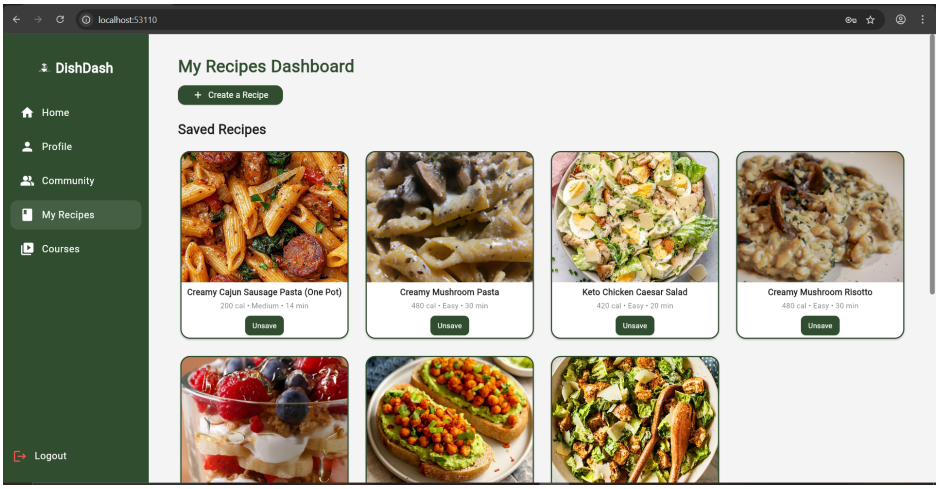

Figure 5.35: User's Recipe Screen in User's Website

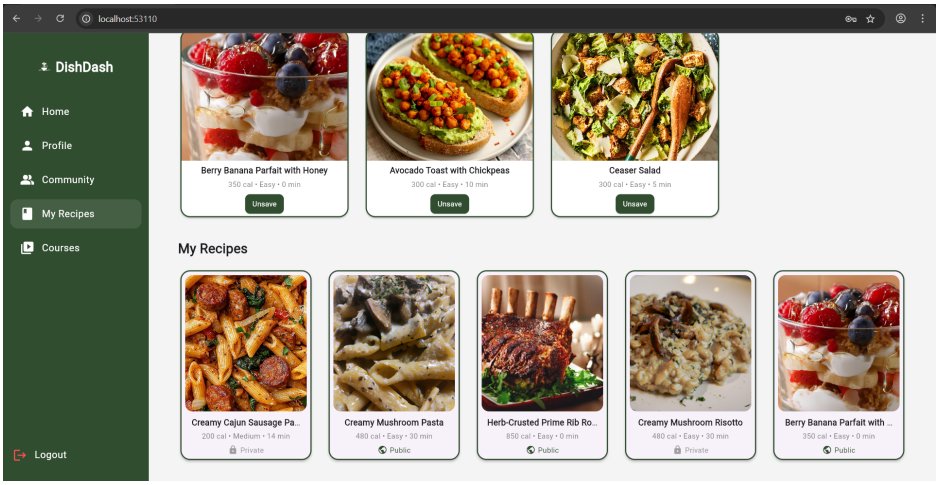

Figure 5.36: User's Recipe Screen in User's Website pt2

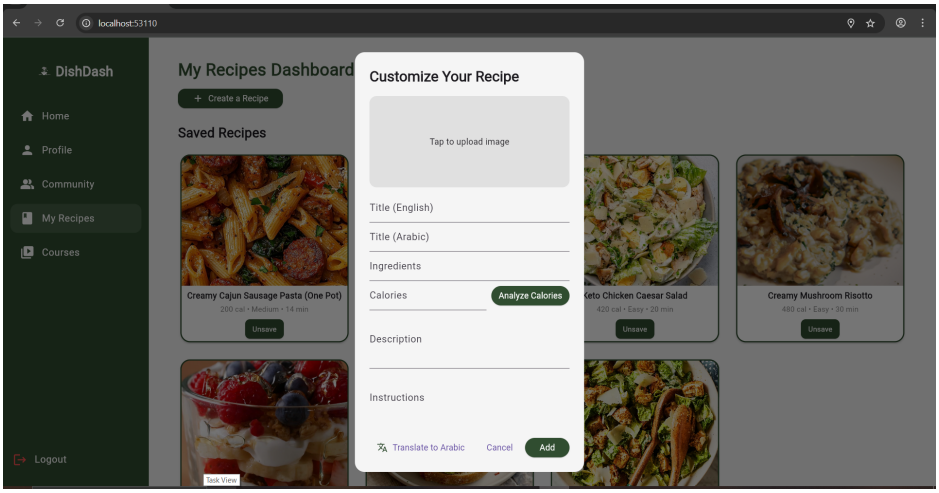

Figure 5.37: Customizing Recipes in User's Website

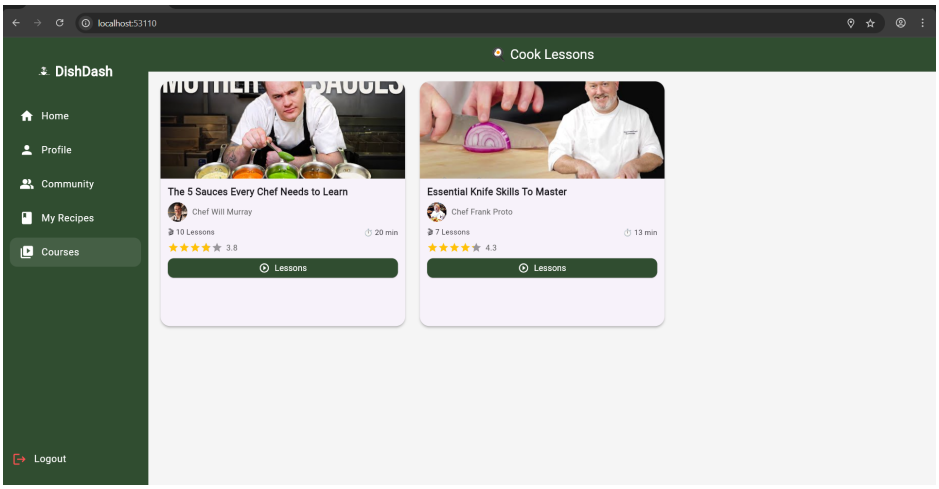

Figure 5.38: Courses Screen in User's Website

5.3.4.2 Store Side

The store's website allows store owners to add items under various categories, set prices, and manage user orders, including updating their status. It also provides key statistics on purchases and customer ratings.

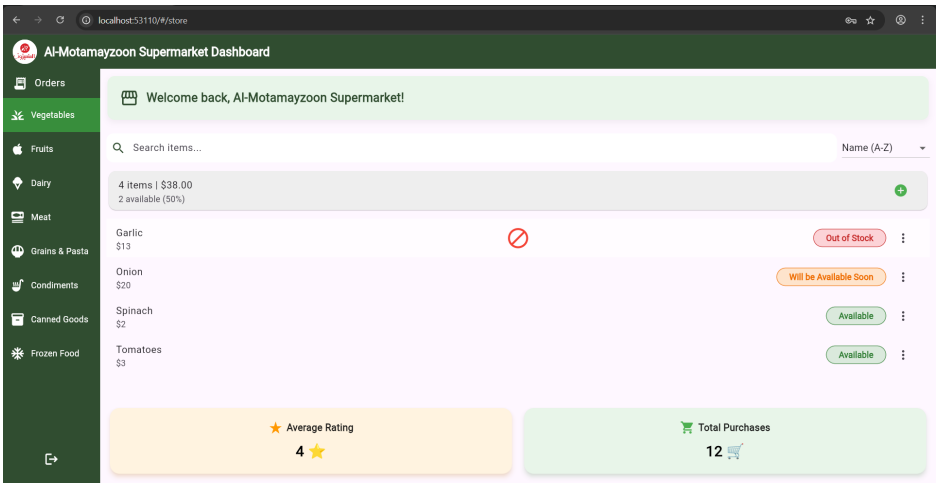

Figure 5.39: Stores' items Categories in Stores' Website

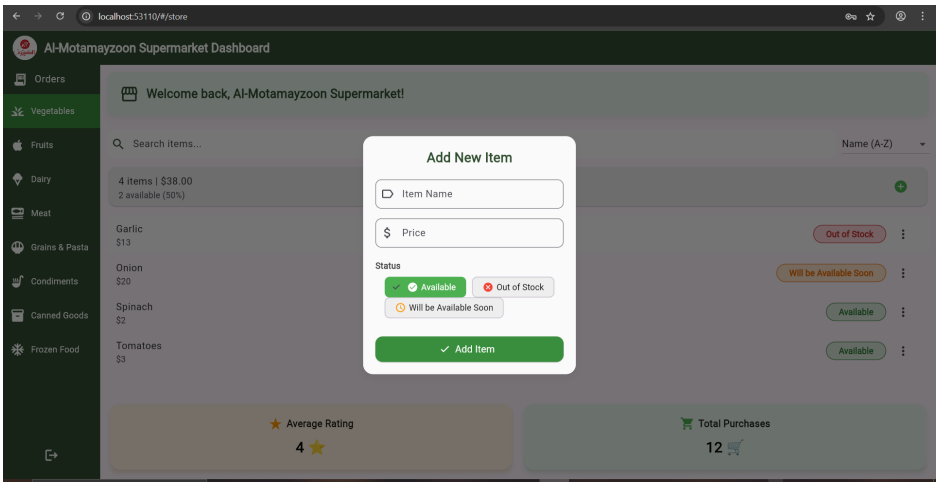

Figure 5.40: Adding Items in Stores' Website

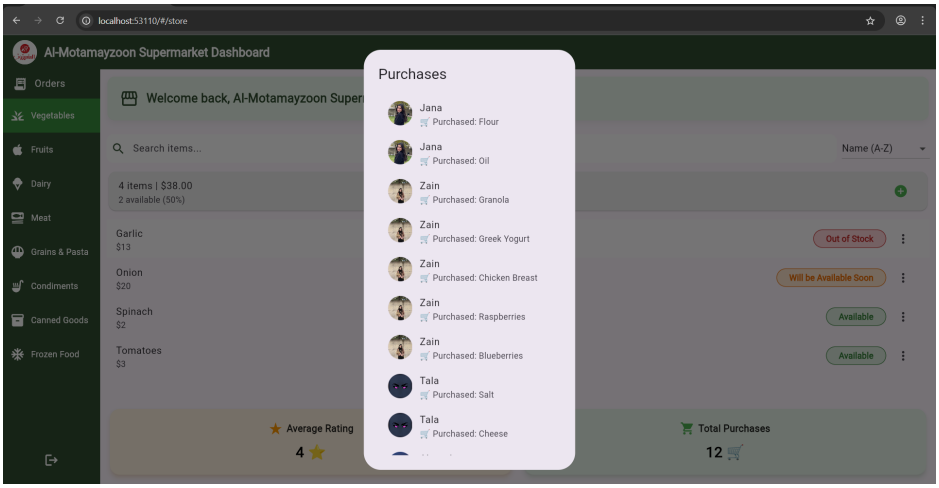

Figure 5.41: Purchases Statistics in Stores' Website

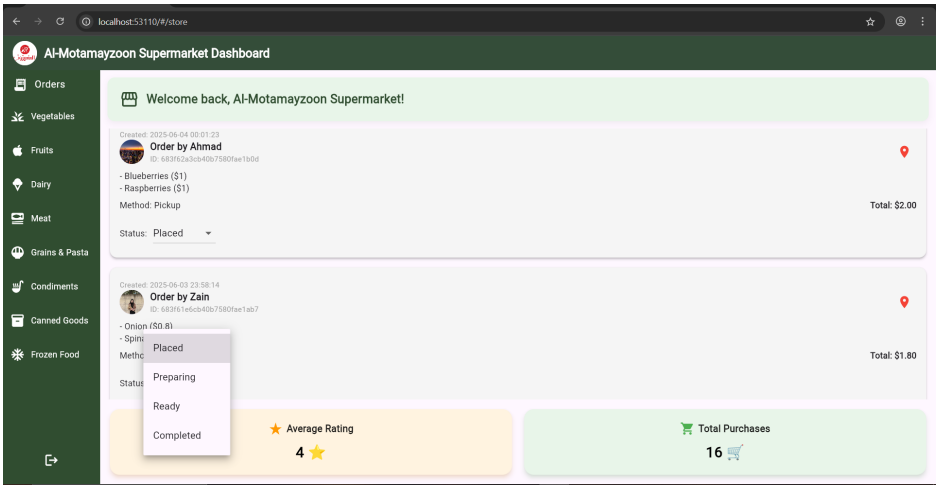

Figure 5.42: Orders Screen in Stores' Website

5.3.4.3 Admin Side

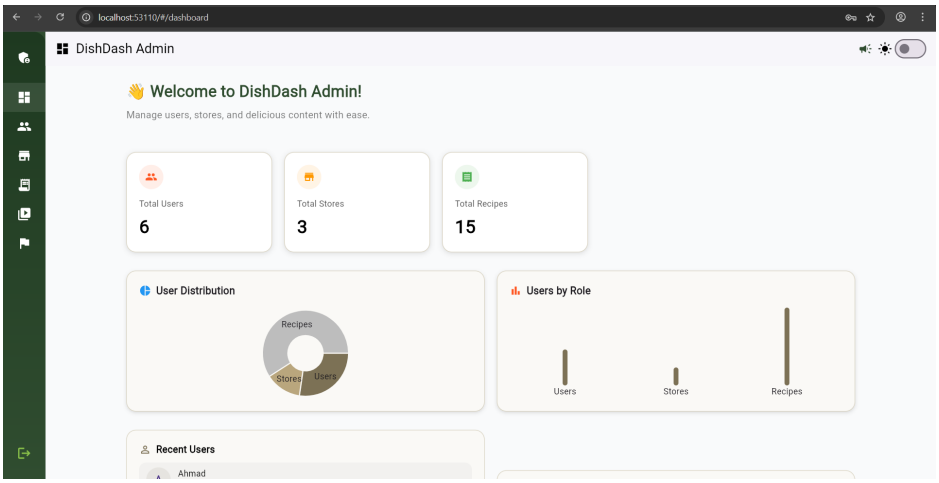

Figure 5.43: Admin's Dashboard and Statistics

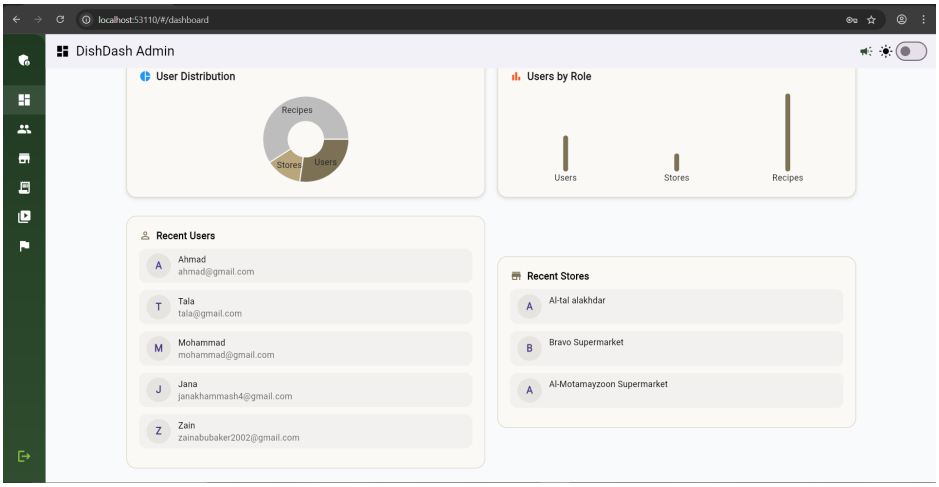

Figure 5.44: Admin's Dashboard and and Statistics pt2

The Admin's primary management platform is the DishDash website. Through it, the Admin can access detailed user information, view their public recipes, monitor users' online or offline status, and send user-specific notifications or announcements. Additionally, the Admin has the authority to delete any user account if necessary.

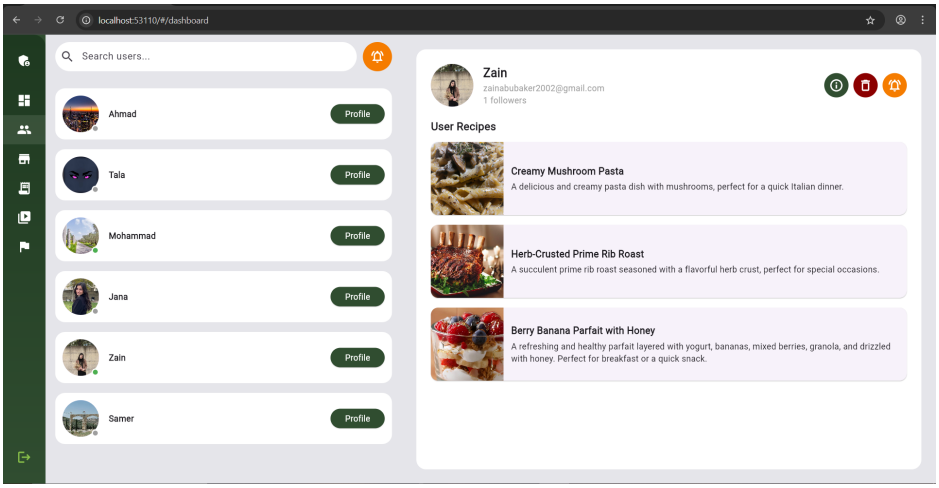

Figure 5.45: User's management for Admin

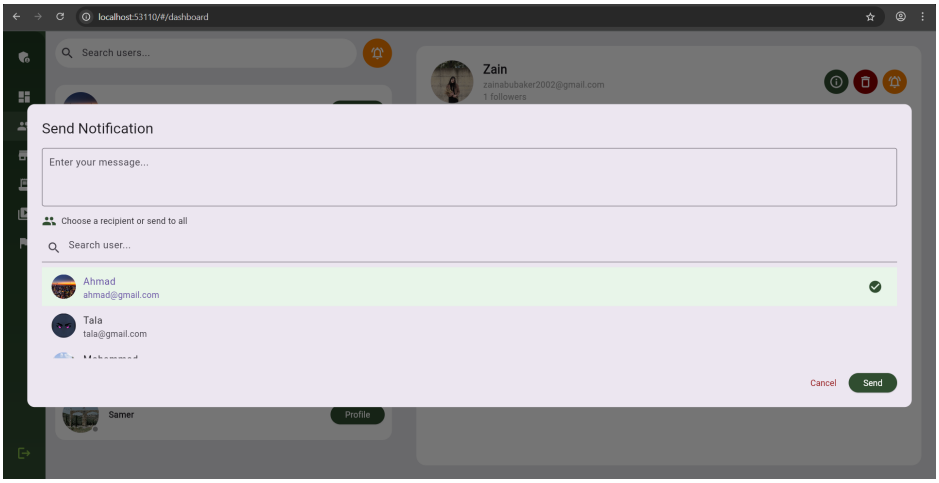

Figure 5.46: Admin Sending user-specific notifications and announcements

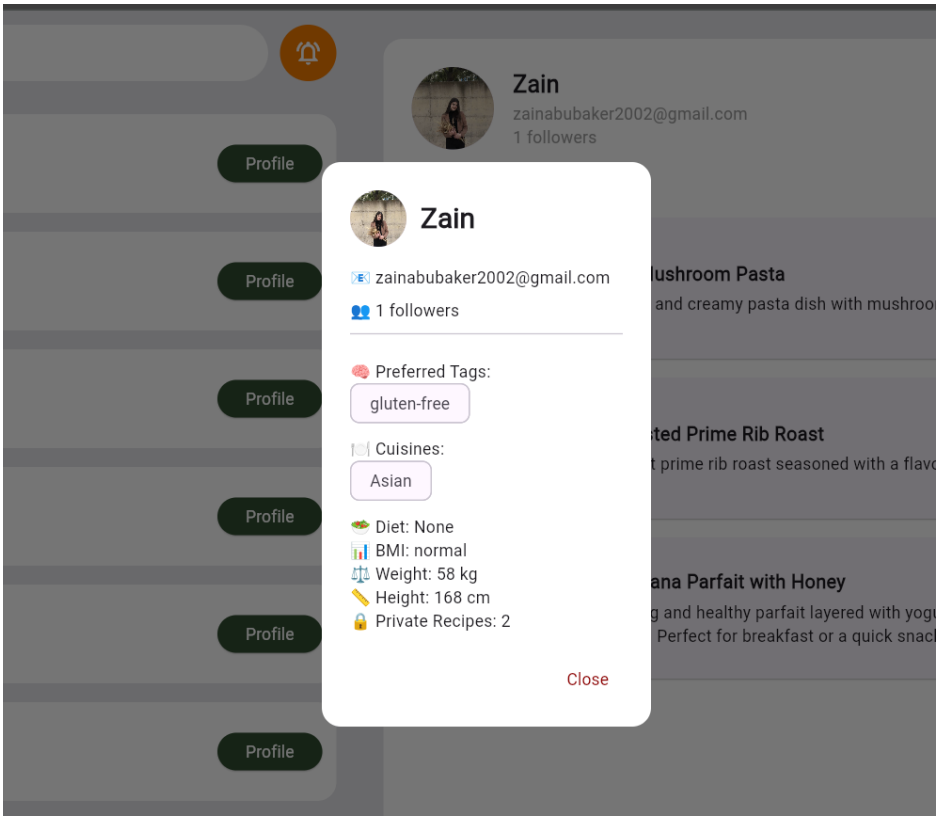

Figure 5.47: User's information

The Admin can manage system and users recipes, by deleting any, and creating new recipes.

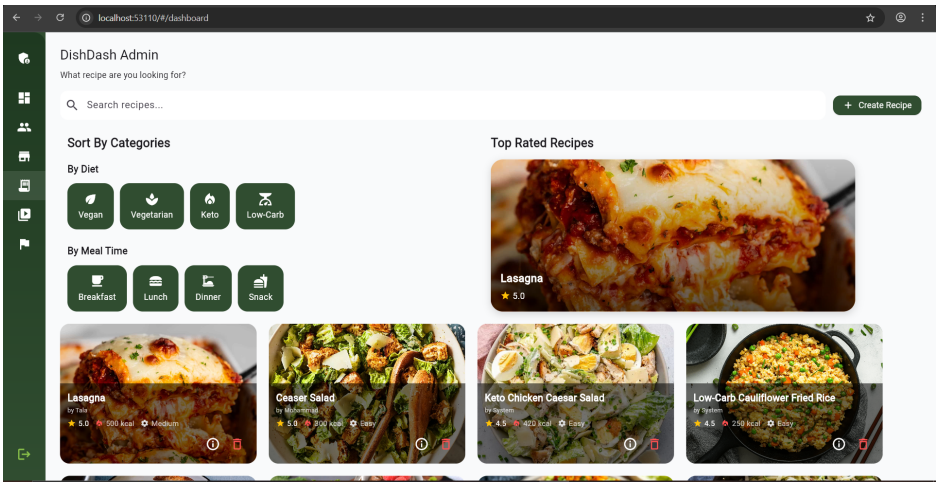

Figure 5.48: Recipes Screen in Admin's Dashboard

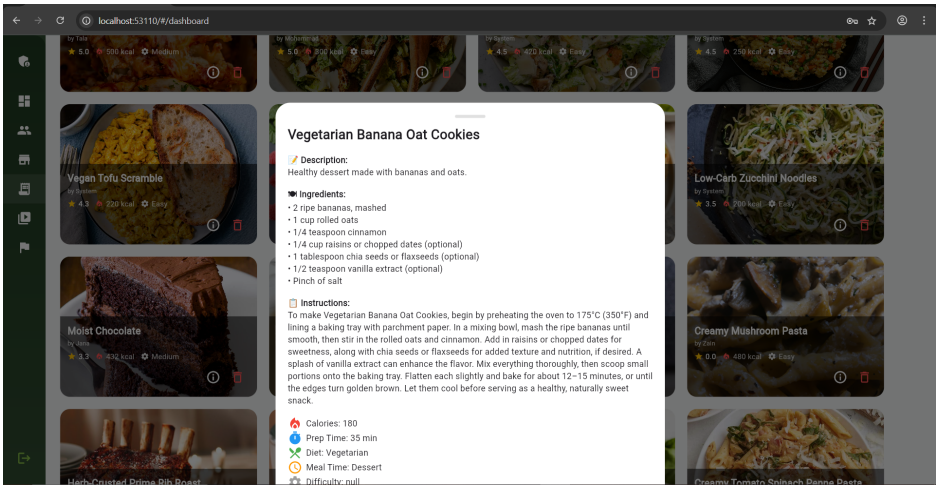

Figure 5.49: Recipes' Details

For challenges, The Admin is responsible of creating new challenges, assigning their dates and their type. After creating any challenge, the Admin can view its participants and submissions, score the submissions and choose winners

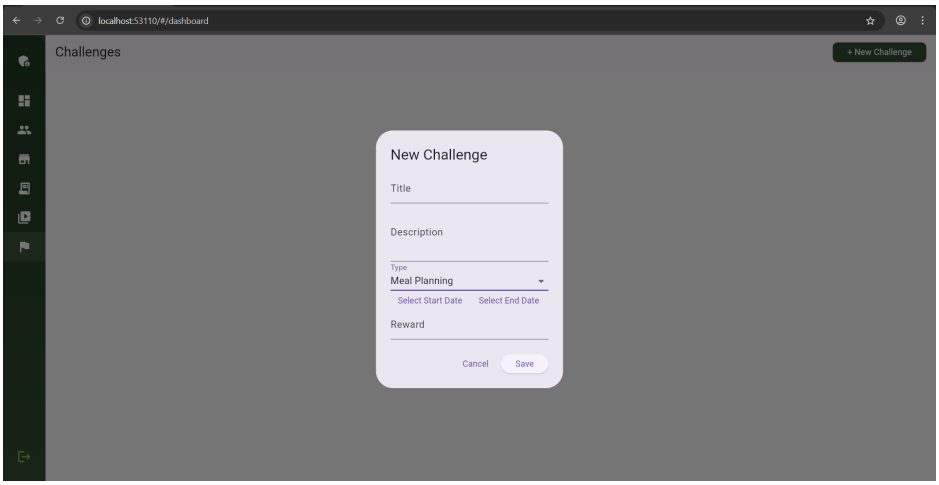

Figure 5.50: Admin Creating Challenges

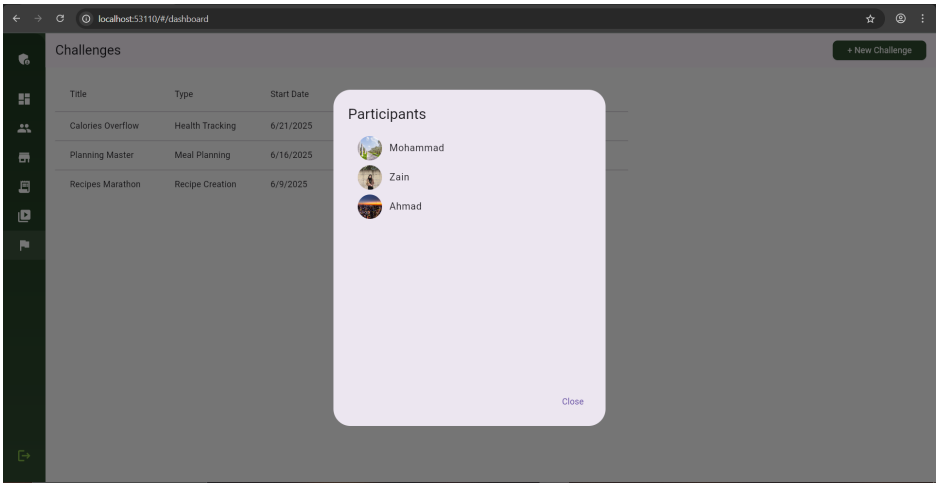

Figure 5.51: Participants in Challenges

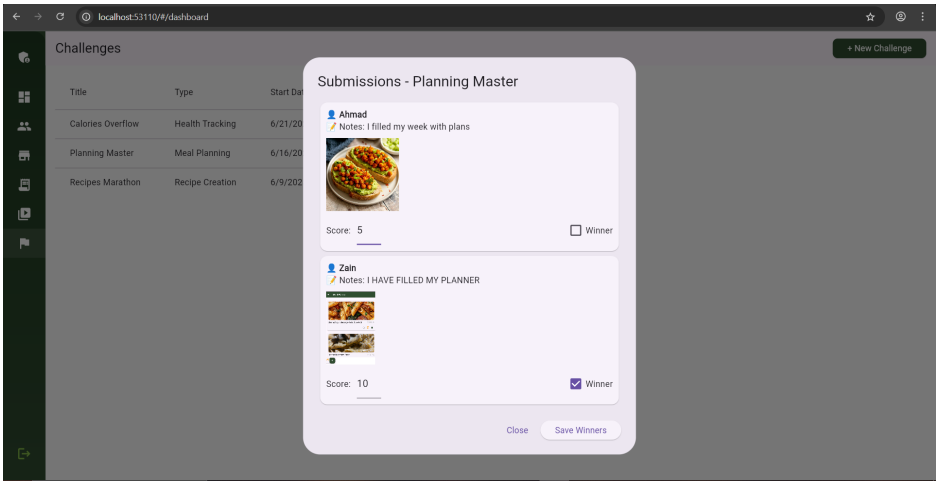

Figure 5.52: Viewing Submissions and Assigning Winners

Stores registered in DishDash are visible on the Admin's dashboard, allowing the Admin to view detailed information for each store and delete them if necessary.

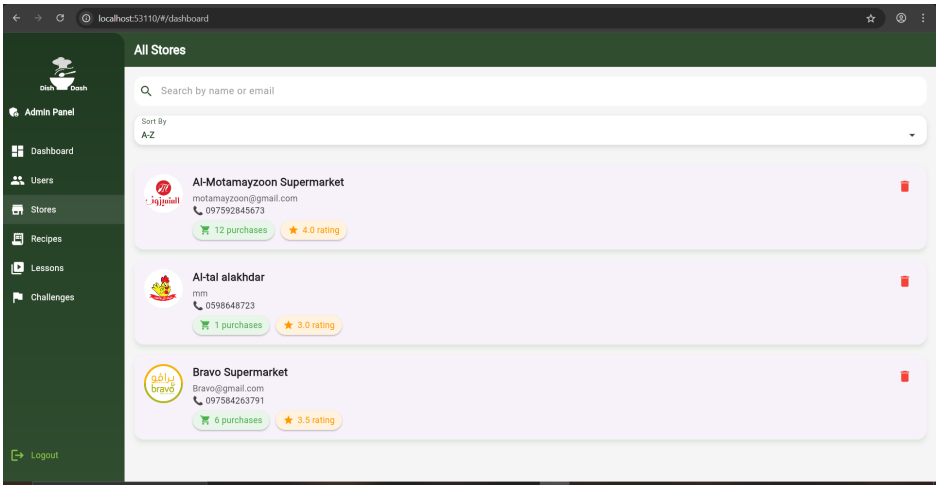

Figure 5.53: Stores Page in Admin's Dashboard

Admin can import videos for cooking lessons, assign their chef, Title, description, and view their rates.

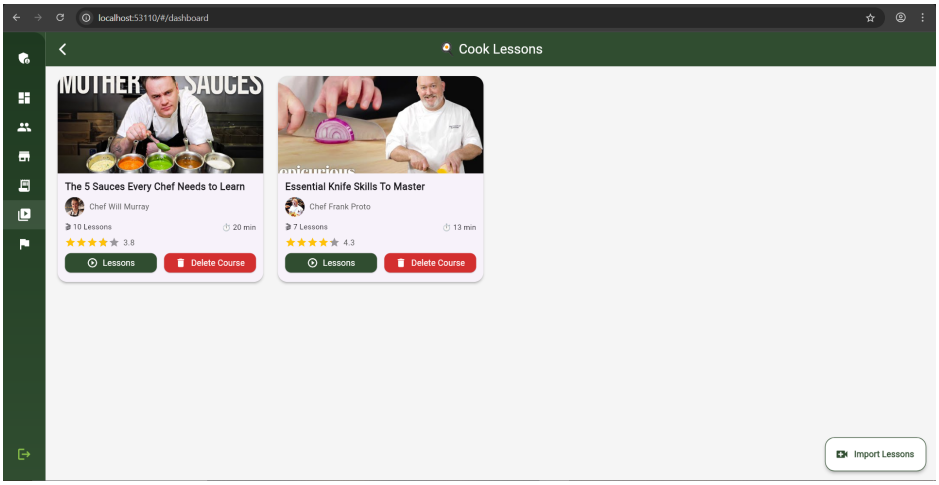

Figure 5.54: Courses Screen in Admin's Dashboard

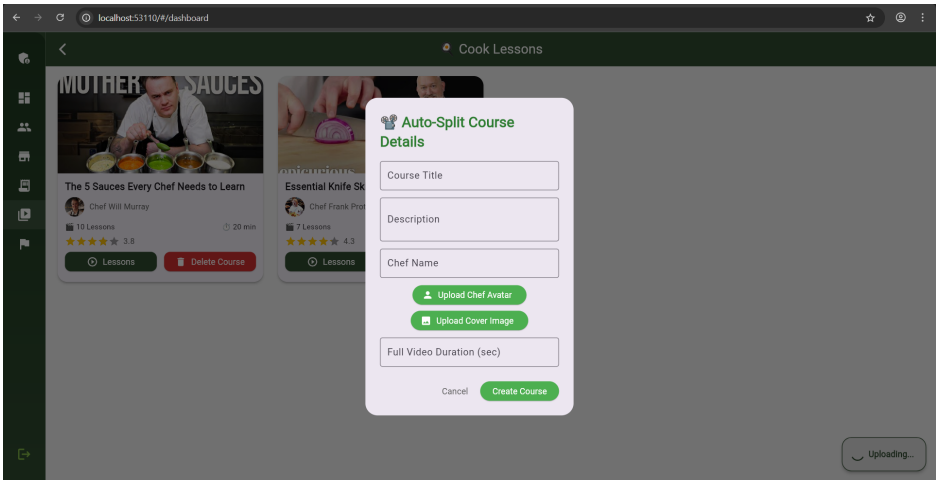

Figure 5.55: Importing Videos

Chapter 6

Results and Discussion

6.1 Results

DishDash has effectively developed a cross-platform application that integrates community features, recipe recommendations, grocery management, and meal planning. Important features like real-time chat, AI-generated recipes, personalized recommendations, and store integration were efficient and satisfied user needs. The recommendation system, community posts, and interactive grocery lists were well-received by users. Real-world usability was improved by store features like order tracking, WhatsApp contact, and location filtering. Managing image uploads across platforms and synchronizing orders with the cart were among the difficulties, but overall, the app operated dependably and provided a seamless user experience. The project showed excellent technical execution and obvious room for expansion.

6.2 Discussion

The creation of DishDash brought to light the difficulties and possibilities of creating a comprehensive food and nutrition platform. The individualized experience provided to users via survey-based dietary customization and AI-driven recipe recommendations was among the most influential features. An improvement over generic food apps, the app was able to recommend meals that were both relevant and health-conscious by incorporating user preferences, allergies, meal timing, and even BMI.

Using Socket.IO to integrate real-time features was another powerful element. Interactivity and user satisfaction were greatly increased by implementing live chat, notifications, and order updates. Following people, leaving comments on posts, and interacting with store profiles gave the platform a community-driven element that transformed it from a simple tool into a food-focused social hub.

Another distinctive feature was the incorporation of store functionality. Real-world value was

increased by enabling users to browse nearby stores, filter by rating or distance, view available items, and place orders. Map-based directions and WhatsApp contact options helped users and local businesses by bridging the gap between online planning and in-person shopping.

But there were difficulties with the project. To maintain consistency, additional handling was needed to manage image uploads on both web and mobile platforms (using Base64 and Cloudinary). It was also challenging to keep the cart updated with orders and to display real-time updates on screens. Additionally, careful design and testing were required to balance complex UI layouts in Flutter, particularly when adapting them to both mobile and web.

During testing, DishDash showed good performance and usability in spite of these problems. Scalable features were made possible by the modular design, and user feedback was largely favorable. The project's technical and practical viability was demonstrated by the successful integration of AI, real-time systems, and health-focused tools into a single platform, laying a solid basis for upcoming enhancements and features.

Chapter 7

Conclusion and Recommendation

7.1 Conclusion

The DishDash project was successful in achieving its objective of developing a feature-rich, user-focused platform that integrates social interaction, grocery management, AI-powered recipe generation, and personalized meal planning. The application provided a seamless and captivating experience on mobile and web platforms thanks to a well-organized backend and an easy-to-use frontend developed with Flutter and Node.js. Although there were issues with image processing, cross-platform consistency, and real-time synchronization, the system was scalable, stable, and successful in meeting user demands for convenience, health, and food discovery.

7.2 Recommendation

It is advised that the AI recommendation engine be improved in the future by adding more sophisticated health data and user feedback loops. Usability would be enhanced by better image handling, particularly for large files and different network conditions. The platform might be strengthened even more by adding delivery choices, order status tracking, and payment integration to the store's features. Future improvements to accessibility, multilingual support, and deeper analytics will also help DishDash reach a wider audience and have a greater overall impact.

7.3 Future Work

Looking ahead, there are several areas of future work for further development of DishDash. In order to improve it, we can:

- **Multilingual Support:** Expand the app interface and content to support multiple languages, including Arabic, English, and potentially French or Spanish, making DishDash

accessible to a broader user base.

- **Smarter AI Recommendations:** Enhance the AI engine to adapt over time using user behavior, seasonal trends, and regional preferences—making suggestions even more relevant and context-aware.
- **Voice Interaction:** Integrate voice-based search and navigation to improve accessibility and hands-free use during cooking or grocery planning.
- **Augmented Reality (AR) Recipes:** Explore AR features that guide users step-by-step through recipes using real-time camera overlays.
- **Dynamic Nutrition Planning:** Adjust meal recommendations based on changing health goals (e.g., weight loss, muscle gain, diabetes-friendly plans).
- **Recipe Scanner:** Allow users to scan handwritten or printed recipes and convert them into structured, digital formats within the app.
- **AI Meal Generator by Mood or Culture:** Let users generate meals based on mood (comfort food, light meals) or cultural themes (Mediterranean, Asian, etc.).
- **Advanced Analytics for Users:** Offer users personal insights into their eating habits, nutritional balance, and most-cooked meals.
- **Global Community Feed:** Introduce region-based filtering or trending tabs for discovering what's popular in other countries or cultures.
- **Offline Mode:** Enable key features like viewing saved recipes, meal plans, and grocery lists without an internet connection.

Chapter 8

References

[1] J. Chen, J. E. Cade, and M. Allman-Farinelli. "The Most Popular Smartphone Apps for Weight Loss: A Quality Assessment". In: *JMIR mHealth and uHealth* 5.3 (2017), e16. DOI: 10.2196/mhealth.5849.

[2] E. Brindal et al. "Design and pilot results of a mobile phone weight-loss application for women starting a meal replacement programme". In: *Journal of Telemedicine and Telecare* 19.3 (2013), pp. 166–174. DOI: 10.1177/1357633X13479702.

[3] C. Pagliari et al. "What Is eHealth (4): A Scoping Exercise to Map the Field". In: *Journal of Medical Internet Research* 7.1 (2005), e9. DOI: 10.2196/jmir.7.1.e9.

[4] Google Developers. *Flutter Documentation*. https://docs.flutter.dev. 2024.

[5] OpenJS Foundation. *Node.js Documentation*. https://nodejs.org/en/docs. 2024.

[6] Express Team. *Express.js - Node.js web application framework*. https://expressjs.com/. 2024.

[7] OpenAI. *OpenAI API Documentation*. https://platform.openai.com/docs. 2024.

[8] Edamam. *Edamam Nutrition Analysis API*. https://developer.edamam.com/edamam-nutrition-api. 2024.

[9] Cloudinary Ltd. *Cloudinary Documentation*. https://cloudinary.com/documentation. 2024.

[10] Google Cloud. *Google Maps Platform Documentation*. https://developers.google.com/maps. 2024.

[11] MongoDB, Inc. *MongoDB Documentation*. https://www.mongodb.com/docs/. 2024.

[12] Unsplash. *Unsplash API Documentation*. https://unsplash.com/developers. 2024.