

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ



An-Najah National University

Faculty of Engineering and Information Technology
Computer Engineering Department

Graduation Project 1

Watan

Watan

Omar Nidal Raddad

Layan Najeh Otman

Supervisor: **Shareef Yaseen**

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

22-Jan-25

Acknowledgement

We would first like to express our sincere appreciation to the Almighty. Without His blessings guiding us every step of the way, we would not have successfully completed this project.

We are truly grateful to our parents and other loved ones. Your constant encouragement has inspired and uplifted us throughout our journey. Your kind and supportive remarks have provided us with strength and stability, laying a solid foundation for our success.

We would also like to extend our heartfelt thanks to our supervisor, Shareef Yaseen, for your invaluable guidance and significant suggestions. Your precise insights, thoughtful remarks, stimulating discussions, and constructive feedback have greatly enhanced the quality of our project.

Finally, we wish to thank An-Najah National University's Computer Engineering Department for their unwavering support and for providing us with the resources and opportunities necessary to complete this project.

Disclaimer

This application was developed by Omar Raddad and Layan Othman from the Computer Engineering Department at An-Najah National University for educational purposes only. The Department of Computer Engineering at An-Najah National University does not endorse the opinions expressed herein, which are solely those of the authors. The content, features, and functionality presented in this application reflect our skills and knowledge at the time of completion.

Table of Contents

Acknowledgment	2
Disclaimer	3
Abstract	6
Chapter 1: Introduction	6
Statement of the Problem	9
Objectives	9
Significance of the work	9
Organization of the Report	10
Chapter 2: Constraints, Standards and Earlier course work	11
Constraints	11
Standards and Codes	11
Earlier Course Work	11
Chapter 3: literature Review	13
Historical Context and Evolution	13
Technological Advancements	13
User Adoption and Reading Habits	13
Economic and Environmental Impacts	13
Social and Cultural Considerations	14
Future Trends and Innovations	14
Chapter 4: Methodology	15
Choosing the idea	15
Choosing the Architecture	16
Backend Architecture	16
frontend Architecture	16
Tools	17
Technologies	18
Database	18

Artificial intelligence (AI)	20
Security Measures	24
Server side	26

Abstract

This project is about a social media application that has been developed for helping Palestinians deal with the day-to-day challenges that they face while living under occupation.

The application does not simply connect people but enables users to deal with the challenges of distance, barriers and checkpoints that make travel a real nightmare.

Featuring built in maps, it offers users the shortest and safest routes hence enabling them to move around faster with less trouble. Also important is the market community section of the application, which allows users to post and browse advertisements for goods and services to be purchased or offered within the community, enhancing the economic activity as well as community bonding. Incorporating technology into every day realities of Palestinians, this app is not only aimed at socializing, but socializing with the aim of improving one's life, creating and enhancing relationships and possibilities which are quite in their scarcity.

There is an element in this project which indicates how technology may engage its users to socialize and also ease their experience in an environment that is difficult.

Chapter 1: Introduction

Statement of the Problem

The Palestinian people face numerous challenges in their daily lives due to political, economic, and logistical constraints. Road closures, traffic congestion, and checkpoints create significant difficulties in transportation, trade, education, and access to essential services. These disruptions hinder the movement of goods and people, negatively affecting the economy and quality of life.

Additionally, the lack of centralized, real-time information about road conditions and closures exacerbates the problem, leaving citizens unable to plan their routes effectively. This results in wasted time, increased transportation costs, and a general sense of uncertainty.

Palestinians also face limited access to marketplaces due to mobility restrictions. This affects both consumers and merchants, as businesses struggle to reach their customers, and citizens find it challenging to procure their daily needs efficiently. Furthermore, there is a lack of platforms that provide relevant local information, address social issues, or facilitate communication within the community.

Given these challenges, there is a pressing need for a digital solution that integrates real-time road condition updates, optimized route planning, and a platform for local commerce. Such a solution should also cater to the broader interests and needs of Palestinian citizens, offering a comprehensive approach to mitigating the impact of the region's ongoing difficulties on everyday life.

Objectives

The primary objective of the “**Watan**” project is to create a versatile and user-friendly application that leverages modern technologies to address the challenges faced by Palestinian citizens. Specifically, the application aims to:

- **Provide real-time updates** on road conditions, closures, and checkpoints, ensuring users can plan their journeys efficiently and minimize delays.
- **Offer optimized route planning** that considers barriers, traffic congestion, and travel distance to help users reach their destinations with ease.
- **Facilitate local commerce** by integrating a virtual marketplace where users can connect with merchants, browse products, and complete transactions without needing to travel to physical markets.
- **Promote community awareness** by including a section dedicated to topics of interest for Palestinians, providing timely and relevant information on social, economic, and political developments.

- **Enhance communication and collaboration** by enabling users to interact directly with merchants, share updates, and contribute information about road conditions or other critical issues.
- **Support economic resilience** by creating a platform that strengthens local trade and provides an alternative means of conducting business amidst mobility restrictions.

Significance of the work

"**Watan**" is set to redefine how Palestinians overcome daily challenges and connect with essential resources. By integrating real-time updates, advanced mapping, and a virtual marketplace, the application ensures a seamless, user-friendly experience on mobile platforms.

The project holds significant potential to enhance mobility, promote local commerce, and foster resilience in a region facing frequent disruptions. This approach supports citizens by reducing the stress of navigating road closures and streamlining access to goods and services.

Watan not only empowers users with vital information but also creates a sustainable model for economic activity and community engagement, ensuring user satisfaction and trust

organization of the Report

The report is structured to provide a comprehensive examination of the development, functionality, and impacts of the "**Watan**" project, organized into the following chapters:

- **Chapter 1: Introduction**
Introduces the project, outlines the problem statement, objectives, significance, and organization of the report.
- **Chapter 2: Constraints, Standards, and Earlier Course Work**
Explores the technical and operational constraints faced during development, the standards and codes followed, and the insights derived from prior coursework that influenced the project.
- **Chapter 3: Literature Review**
Analyzes existing literature related to navigation systems, real-time traffic management, virtual marketplaces, and community-focused applications, highlighting technological advancements and user engagement strategies.
- **Chapter 4: Methodology**
Describes the process of project conceptualization and system design, including the methodologies used for developing the mapping system, marketplace features, and communication tools. It also details the backend and frontend development technologies, testing procedures, and the Agile development approach.
- **Chapter 5: Results and Discussion**
Presents the outcomes of the application's implementation, evaluates its performance against the defined objectives, and discusses user feedback, system accuracy, and usability.
- **Chapter 6: Conclusion**
Summarizes the achievements of the project, discusses potential enhancements and future work, and reflects on **Watan's** impact in addressing the challenges faced by Palestinians.
- **Chapter 7: References**
Provides a comprehensive list of scholarly, technical, and informational resources used throughout the report to ensure credibility and facilitate further research.

This structure ensures a logical and cohesive presentation of the project, allowing readers to understand the "**Watan**" application's development process, technological foundation, and its significance in improving daily life for Palestinian communities.

Chapter 2: Constraints, Standards and Earlier course work

Constraints

The development of the "Watan" project encountered several constraints that posed challenges to the team. One significant constraint was the team's initial lack of experience in mobile application development, necessitating a steep learning curve in languages and tools such as Flutter and Node.js for the backend, along with MongoDB for database management. Additionally, the absence of specialized training courses and sometimes inadequate skills to study specific topics made the learning process particularly challenging. Time constraints further compounded these issues, as the development timeline was tight, and integrating new technologies required substantial effort and adaptation.

Standards and Codes

Adhering to industry standards and coding best practices was crucial in ensuring the reliability, scalability, and maintainability of "Watan". The project followed standards for web development and database management, emphasizing secure, efficient, and effective application components. Utilizing flutter and Node.js, the team implemented standards in coding conventions, application architecture, and interface design to facilitate a seamless user experience while maintaining robust backend operations.

Earlier Course Work

Work in the previous session the project team built on the knowledge and skills acquired from various previous courses, which enabled them to them to handle the software project effectively.

Related courses include:

- 1. Object-Oriented Programming Course:** This course provided a foundation in object-oriented programming principles and practices. The concepts learned helped in designing and Implement modular and reusable code components.

2. Web course: The web course provided the team with the necessary skills for development Interactive and easy-to-use web applications. Understanding web development Technologies and best practices facilitated the creation of the project's web application Front and back end.

3. Database Course: The Database course provided insights into database management systems and data modeling techniques.

4. Critical thinking cycle: The critical thinking cycle played a vital role in research and writing this report. He stressed the importance of presenting ideas effectively Professional context. Through this course, the team honed their skills in organizing ideas,

- Articulate arguments and convey information in a clear, concise and persuasive manner.

5. Distributed Operating System Course: The course is provided on Distributed Operating Systems Understand the principles and techniques of distributed computing. This knowledge It influenced the design and development of the back-end system of the project, facilitating efficiency Communication and coordination between various components.

In addition, the team participated in various workshops and courses to enhance their skills and expand their knowledge:

1. Node.js Workshop: The Node.js workshop equipped the team with the necessary skills for development Server-side applications using Node.js. This knowledge was useful in implementation Project backend functionality.

2. Flutter workshop: The workshop focused on creating mobile applications using Flutter Dart. This workshop enabled the team to benefit from Flutter's capabilities in Building responsive interfaces.

Chapter 4: Methodology

Choosing the Idea

The concept for the "**Watan**" application was inspired by the problems and challenges that we personally face in our daily lives. As students, we often experience the difficulties of traveling between our universities and homes due to road closures, checkpoints, and unpredictable conditions. Additionally, my family runs clothing stores, and we have witnessed firsthand how recent events and mobility restrictions have severely impacted local businesses and trade.

It is also evident that people increasingly rely on scattered sources of information to stay informed, leading to confusion and inefficiencies due to the overwhelming abundance of these sources. This realization highlighted the need for a centralized, reliable platform to address these challenges.

From these experiences, the idea of "**Watan**" was born—a digital solution designed to integrate advanced mapping technologies, real-time road condition updates, and a virtual marketplace. The application not only facilitates efficient mobility for users but also supports local businesses by creating an accessible platform for commerce.

By addressing these challenges, "**Watan**" aims to provide a streamlined, comprehensive tool that empowers users to navigate their daily lives with greater ease and confidence, offering a proactive solution to the unique difficulties faced in our region.

Choosing the Architecture

The architecture of "**Watan**" was carefully designed to ensure scalability, performance, and a seamless user experience, utilizing Flutter to address both web and mobile platforms efficiently.

Backend Architecture

The backend architecture of "**Watan**" is built using Node.js, a highly efficient and scalable server-side platform. Node.js was chosen for its ability to handle asynchronous programming, enabling the system to process a large number of simultaneous user requests without compromising speed or performance. For data storage, MongoDB, a NoSQL database, was selected for its flexibility and ease of integration with Node.js, making it suitable for managing the unstructured and diverse datasets required by the application. The backend is further supported by Express.js, a lightweight and robust web application framework. Express.js simplifies routing, improves security, and enhances the application's performance by efficiently managing server-side processes.

Frontend Architecture

The frontend architecture utilizes Flutter for both web and mobile platforms. Flutter was chosen for its ability to create highly responsive and visually appealing user interfaces while maintaining excellent performance. By using Flutter for both platforms, "**Watan**" ensures consistency in design and functionality across devices, offering users a unified experience whether they access the application on their smartphones, tablets, or desktops.

Flutter's single codebase reduces development time and maintenance efforts, while its capability to compile into native code ensures high performance across Android, iOS, and web platforms. This approach allows "**Watan**" to deliver a smooth, engaging, and intuitive user experience, seamlessly integrating real-time mapping, a virtual marketplace, and other essential features in one application.

Tools

In order to develop of this project, a suite of sophisticated tools was employed to ensure efficiency and effectiveness in various aspects of the project.

- **Visual Studio Code:** it was used as primary Integrated Development Environment (IDE), for coding, editing, and debugging purposes.
- **Android Studio:** it was utilized as an emulator to simulate and test the mobile application across a range of virtual environments.
- **GitHub:** a Collaboration and version control were facilitated through GitHub, which allowed work organization with my partner and efficient tracking of code modifications.
- **MongoDB:** for features like real-time chatting and notifications, MongoDB's 'NoSQL' Realtime database was chosen for its robustness and scalability.
- **Postman:** it was used for testing different APIs, to ensure the reliability of our application, and make different HTTP request and response.

Database Structure for Watan

For our Watan application, we opted to utilize a NoSQL database, MongoDB, to effectively manage and store our diverse data needs. MongoDB is a popular document-oriented database that utilizes BSON (Binary JSON) for data storage, allowing us to handle a variety of information types seamlessly.

Given the dynamic nature of our application, which includes features like real-time road conditions, local commerce, and community discussions, MongoDB's flexibility is particularly advantageous. Its schema-less architecture enables us to adapt easily to changes in data structure, ensuring that we can efficiently manage unstructured or semi-structured data.

Our database is organized into several key collections, including:

- **Users:** Storing user profiles and preferences to personalize their experience.
- **Products:** Managing items available in the local store, facilitating easy browsing and purchasing.
- **Categories:** Classifying products for better organization and searchability.
- **Checkpoints:** Maintaining data on road conditions and barriers to enhance navigation.
- **Conversations:** Supporting user interactions and communications within the app.
- **Messages:** Allowing users to send and receive messages related to their transactions or inquiries.
- **Notifications:** Keeping users informed about important updates and alerts.
- **Posts:** Enabling community engagement through discussions on relevant topics.

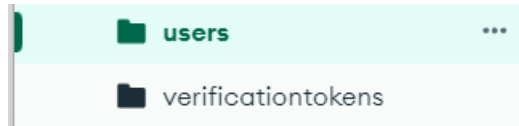
MongoDB's robust querying capabilities and aggregation framework empower us to perform complex analyses, such as generating billing records for users' purchases and assessing the impact of road closures on trade and daily life. Additionally, its support for horizontal scalability ensures that as our user base grows and data complexity increases, we can maintain high availability and performance.

In summary, MongoDB emerged as the ideal choice for Watan due to its flexible schema design, scalability, and powerful querying capabilities, providing a solid foundation to meet the evolving demands of our application.

Security Measures

Authentication and Authorization:

- Implement robust user authentication mechanisms with **JWT**



When creating an account, the user must verify the account from the email
In this case, verifying it is an error for the user until it is verified a turns into true

```
role : "user"  
verified : true
```

Data Encryption and Protection:

- Encrypt sensitive data at rest and in transit using **Crypt** for password.

```
_id: ObjectId('67603a5eea28864d5cd43d72')  
username : "john_doe"  
email : "john.doe@example.com"  
password : "$2a$10$DCLxdLAUyceecRRmj0wldOKSfoVRCuANK/yndGvj4NMKo7b.QXC3W"
```

Session Management:

- Implement secure session management, including session ID generation, storage, and expiration.
- Protect against session hijacking, session fixation, and other session-related attacks.
- Regularly invalidate and renew session tokens to limit the window of exposure.

```
_id: ObjectId('67603a5eea28864d5cd43d72')  
username : "john_doe"  
email : "john.doe@example.com"  
password : "$2a$10$DCLxdLAUyceecRRmj0wldOKSfoVRCuANK/yndGvj4NMKo7b.QXC3W"  
city : "Jerusalem"  
location : "Old City"  
gender : "Male"  
createdAt : 2024-12-16T14:34:06.898+00:00  
updatedAt : 2025-01-18T20:33:24.850+00:00  
__v : 0  
fcmToken : "fUpwCwS1T6-N0r0uBCT8nV:APA91bETm8MQ78hE4aowgl2nr6F3zeY82uZghP-e40q01gC..."
```

Server Side

For the server-side development of our application, we decided to utilize the robust and open-source JavaScript runtime environment, Node.js. Node.js is renowned for its event-driven, non-blocking architecture, making it an excellent choice for building scalable network applications. This architecture allows Node.js to handle multiple simultaneous connections efficiently, which is crucial for maintaining high performance and responsiveness as our user base expands.

Node.js facilitates seamless integration with our MongoDB database through HTTP requests, enabling smooth data interactions. Given the asynchronous nature of operations in a NoSQL database like MongoDB, Node.js's ability to manage asynchronous processes effectively makes it an ideal fit for our application. This capability ensures that our server-side operations can handle real-time data processing and user interactions without significant delays.

One of the standout features of Node.js is its capacity to manage numerous concurrent connections with minimal overhead. This feature is particularly beneficial for our application, as it ensures that we can maintain optimal performance even as the number of users and data complexity increases.

Another significant advantage of Node.js is its vibrant community and the npm (Node Package Manager) ecosystem. npm provides access to a vast array of libraries and modules, which enhances the flexibility and extensibility of our server-side implementation. This extensive library support allows us to manage our codebase efficiently and utilize various function modules for data representation and other tasks.

Additionally, using Node.js enables us to employ JavaScript consistently across both the client and server sides of our application. This single-language approach simplifies the development process, reduces the overhead associated with context switching, and improves overall code maintainability. The consistency provided by using JavaScript throughout the stack enhances the efficiency and manageability of our server-side code.

In conclusion, Node.js is a highly suitable choice for our server-side requirements due to its scalability, effectiveness in handling asynchronous operations, and robust library ecosystem. Its compatibility with JavaScript across the full stack aligns well with our application's design, contributing to a more streamlined and efficient development process.

UI Design

To create an attractive and intuitive user interface for the Watan application, we focused on modern mobile and web development standards. We chose to develop the application using Flutter, an elegant, open-source user interface software development kit (UI SDK) created by Google. Flutter ensures a seamless experience across multiple devices, including both mobile and web platforms. By using Flutter, we can maintain a single codebase that works efficiently across iOS, Android, and web, resulting in a unified and streamlined development process.

Flutter utilizes Dart, a reliable and powerful programming language known for its performance and ease of use. With Flutter's extensive library of customizable UI components, we were able to design a fully responsive and adaptable user interface for the Watan application. This toolkit allows us to create expressive and flexible UIs, ensuring that users have a clear and straightforward experience while interacting with the app.

One of Flutter's standout features is its ability to handle critical platform-specific elements such as scrolling, navigation, fonts, and icons with ease. This capability ensures that the application feels native on each platform, providing a consistent user experience. Additionally, Flutter's efficiency is enhanced by native Dart compilers, which compile Flutter code into native machine code. This compilation process results in a high-performance and optimized application, improving the overall responsiveness and speed of the Watan application.

For the web version of Watan, we continued to leverage Flutter's capabilities. Flutter for web allows us to use the same codebase and UI components as the mobile version, ensuring a consistent design and user experience across all platforms. This approach not only simplifies the development process but also reduces the time and effort required to maintain separate codebases for mobile and web.

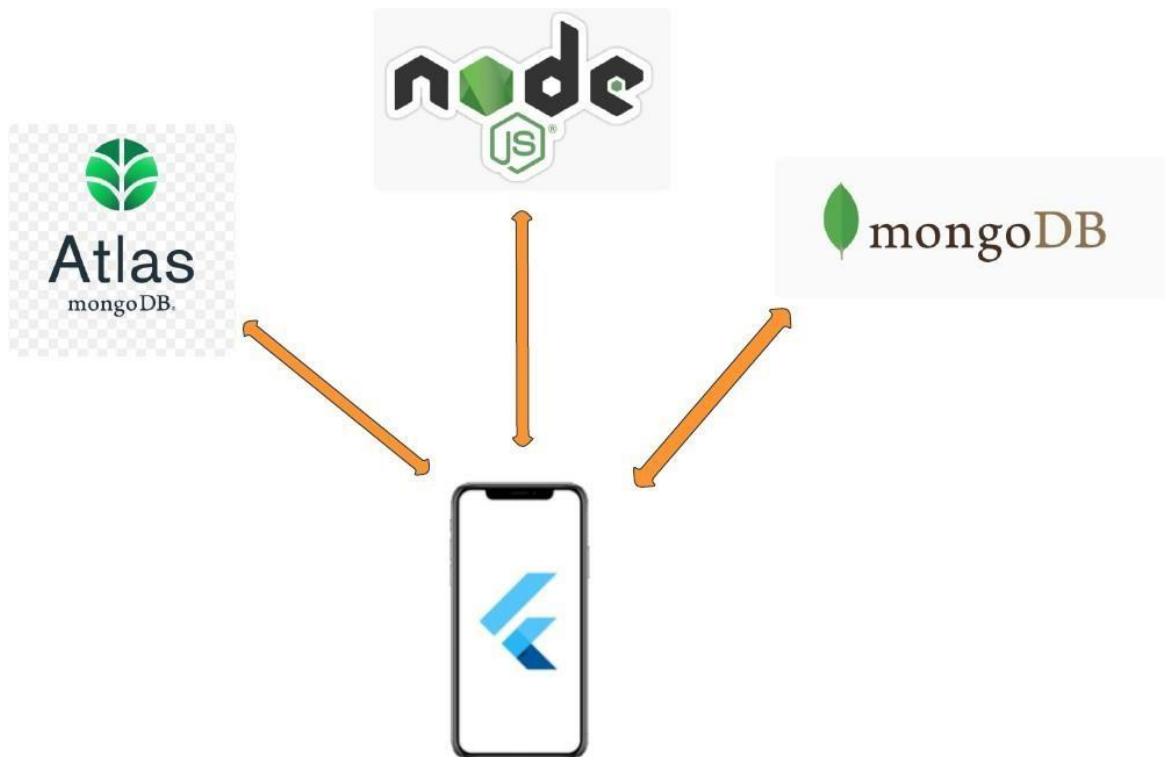
Flutter's rich ecosystem of widgets and tools enables us to create complex and dynamic user interfaces with ease. The framework's hot reload feature significantly speeds up the development process, allowing us to make real-time changes and see the results instantly. This feature is particularly beneficial for iterative design and testing, ensuring that the final product meets our high standards for usability and aesthetics.

In conclusion, by using Flutter for both mobile and web development, we were able to create a cross-platform Watan application with an attractive and intuitive user interface. Flutter's customizable UI components, efficient compilation process, and single codebase approach ensure a consistent and enjoyable user experience across Android, iOS, and web platforms. This unified development strategy not only enhances the application's performance but also simplifies maintenance and future updates, making Flutter an ideal choice for the Watan application.

Application Architecture

In our project, we adopted the Model-View-Controller (MVC) architectural pattern, a robust paradigm that allowed us to organize and streamline our system efficiently:

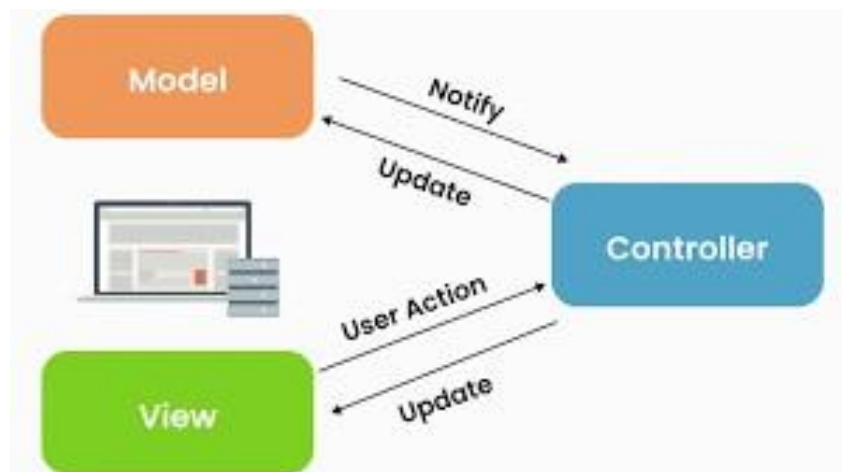
Model: The Model, which forms the foundation of our MVC design, contains the essential information and business logic for our system. The main data store, MongoDB, houses all of the important data. In addition to handling requests from the user interface (View) and backend activities (Controller), the Model is in charge of maintaining data integrity. It serves as the official data source and is essential to preserving information consistency across the system. The Model is a representation of the underlying framework that guides and supports system functioning as well as user interactions.



View: In our MVC paradigm, the graphical user interface (GUI) acts as the View. The View offers an interactive platform with several capabilities that are tailored to meet the demands of users. Users are able to do things like book reservations, keep track of them, and use the rating and search functions. The user-facing part of the system is called The View, and it provides a simple and easy-to-use interface for easy system interaction.

Controller: Designed to be the core of our architecture, the Controller is responsible for managing backend functions for the website and the application. It is created with Node.js. This essential element coordinates the information and action flow between the Model and the View by acting as a go-between. The Controller makes it possible for the Model and the View to communicate with one other in a way that makes the system work as a whole. It is crucial in handling user requests, controlling data flow, and preserving the general consistency of system operations.

In conclusion, our project's MVC architecture creates an effective and transparent division of responsibilities. The Controller coordinates the interactions between the Model and the View, the View maintains the user interface, and the Model manages the data. This methodical approach improves our system's overall resilience, scalability, and maintainability.



Agile Methodology

We used the Agile model technique to execute our software system, strategically segmenting the process into five separate stages:

1. Requirement Analysis (Stage 1):

Holding lengthy meetings to exchange ideas to identify and list the basic features allocated to the program, in addition to conducting a questionnaire to research some points from the readers' point of view in general.

2. Planning (Stage 2):

Establish a comprehensive timetable for completing all the tasks required to complete the project, giving each activity a specific deadline for completion.

Conducting evaluations and conversations with the project supervisor to follow up on the most important developments of the project.

3. Designing the Requirements (Stage 3):

Sort project work into priority categories, starting with the most important to the least important.

Periodic meetings are held among team members to evaluate progress, deliberate on upcoming tasks, and address obstacles, ensuring quick resolution.

A sketch is created for each component of the project, outlining the purpose, target audience, and technical details of each intended feature.

4. Development (Stage 4):

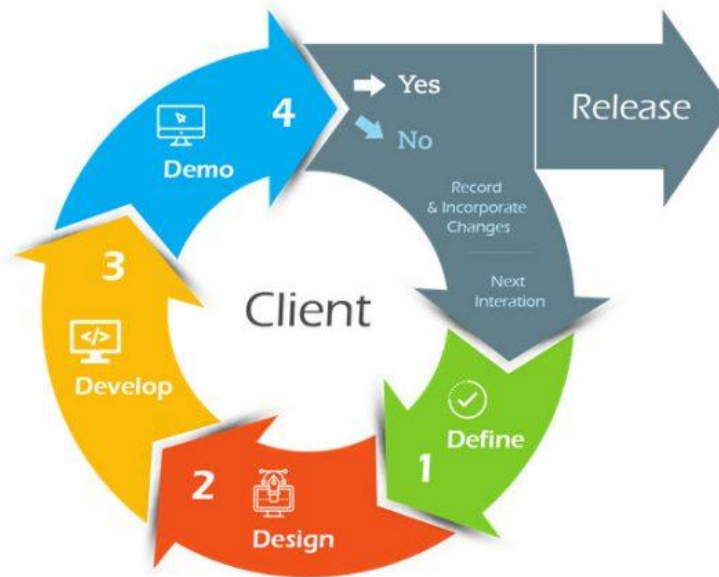
He began coding the front and back of each page, respectively, from most important to least.

5. Testing (Stage 5):

Postman testing was used to verify the back-end's functioning and make sure everything was correct.

Ensured that each page functions properly by establishing and testing the link between the front end and back end.

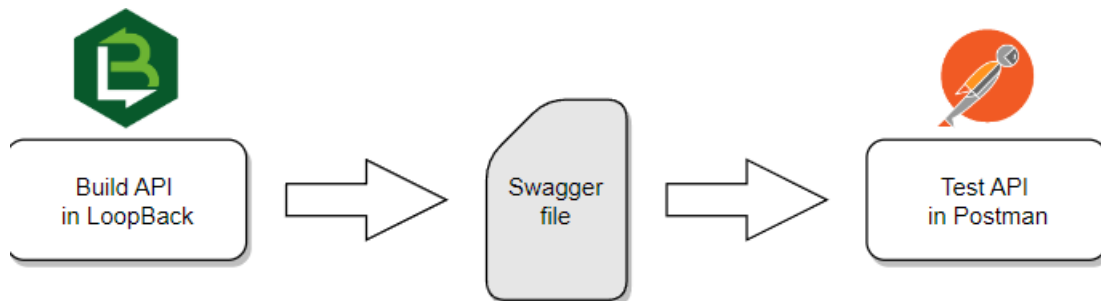
We were able to create and improve our software system iteratively thanks to the Agile model approach, which also ensured alignment with changing priorities and needs.



Testing

With its many features, Postman is a very feature-rich and easy-to-use application that makes testing backend programs much easier. Because of its user-friendly interface, developers may create and execute a variety of HTTP requests, including GET, POST, PUT, and DELETE, with ease, facilitating extensive testing of backend features. Request setup is where Postman shines; it makes it simple to create parameters, headers, and body content, simulating real-world circumstances for reliable testing.

One of Postman's main advantages is its organization, which enables users to organize requests into collections and offers an organized method for managing and running test cases. Developers may move between several testing environments with ease since this arrangement includes environment variables. With the addition of automation processes, Postman goes beyond manual testing and makes it possible to write JavaScript test scripts for intricate testing scenarios and repetitive activities. Because of this automation, testing is far more efficient and produces repeated, reliable findings.



Postman also offers a thorough view of API answers, including headers, content, and status codes, through its deep response inspection feature. By allowing team members to share collections and requests, the technology makes cooperation easier. Testing is easily included into the software development lifecycle because to its connection with monitoring tools, continuous integration platforms, and version control systems. By enabling developers to record API endpoints, request forms, and anticipated answers, Postman's documentation features further increase its usefulness and make it an invaluable tool for both development and testing teams. In conclusion, Postman proves to be a priceless tool for developers, providing a comprehensive solution for backend testing through its collaboration features, automation capabilities, and user-friendly design.

Artificial intelligence (AI)

Recommender system (hybrid recommender system):

How it functions:

To support our recommendations, we employ a mixed recommender system. Hybrid systems are a combination of two distinct types of recommender systems: content-based filtering and collaborative filtering. One way to propose products based on how similar they are to specified items is through content-based filtering. In other words, he can suggest the second book to me if I enjoyed the first Lord of the Rings book and the second book is comparable to the first. User ratings are used in a technique called collaborative filtering to find similarities between a user and an object.

hybrid recommender system	
collaborative filtering	Content-based filtering
Broader exposure to many different products.	Content-based filtering does not require data about other users' preferences. Therefore, it is easier to implement and potentially more scalable.
No need for domain knowledge or understanding of item content.	It can recommend new items instantly without waiting for user feedback or data.
Captures the change in user's preferences over time.	This can take into account individual user preferences and the specific features they look for in a product, resulting in more relevant recommendations.
Improving user experience by suggesting top-rated products based on what similar users like.	
Easy creation and use of the model	

Therefore, hybrid systems can benefit from both approaches.

1. They can improve recommendation accuracy and better cover different products.
2. They can overcome the limitations of individual recommendation strategies and provide more personalized recommendations.
3. They can significantly improve the effectiveness and efficiency of recommendation engines in various applications.

Content-based filtering(CB)?

Compute Pairwise Similarity:

- Calculate the cosine similarity between all pairs of books.
- For each book in the dataset, compute the pairwise similarity with all other books using a similarity metric, such as cosine similarity or Pearson correlation.

$$\text{cosine_similarity}(A, B) = (A \cdot B) / (\|A\| \|B\|)$$

Build the Content-Based Recommendation Model:

- store the ranked lists of similar books for each book.

Provide Recommendations:

- Given a target book, retrieve the ranked list of similar books from the recommendation model.
- Sort the list in descending order of similarity scores.
- Return the top `n` books as the content-based recommendations.

Collaborative Filtering (CF)?

- Making filtering decisions for an individual user based on the judgments of other users
- Inferring individual's interest/preferences from that of other similar users
- General idea
 - Given a user u , find similar users $\{u_1, \dots, u_m\}$
 - Predict u 's preferences based on the preferences of u_1, \dots, u_m
 - User similarity can be judged based on their similarity in preferences on a common set of items
- Users with the same interest will have similar preferences
- Users with similar preferences probably share the same interest

	O1	O2	...	oj	...	on
U1	3	2			3	
U2						
...						
Ui	1			??		
...						
Um	5					

O:object u:user

The number in matrix is ratings

$X_{ij} = f(U_i, O_j) = ??$

The task

· Assume known f values for some (u,o)'s

· Predict f values for other (u,o)'s

Memory-based is Approaches

so the General ideas:

- X_{ij} : is a rating of object o_j by user u_i

- n_i : is average rating of all objects by user u_i

-Normalized ratings : $V_{ij} = X_{ij} - n_i$

-Prediction of rating of object o_j by user u_a

$$\hat{v}_{aj} = k \sum_{i=1}^m w(a, i) v_{ij} \quad \hat{x}_{aj} = \hat{v}_{aj} + n_a \quad k = 1 / \sum_{i=1}^m w(a, i)$$

Figure 4 Memory-based Approaches in Recommender Systems Collaborative Filtering

so the Specific approaches differ in $w(a, i)$

the distance in the same meaning similarity between user u_a and u_i

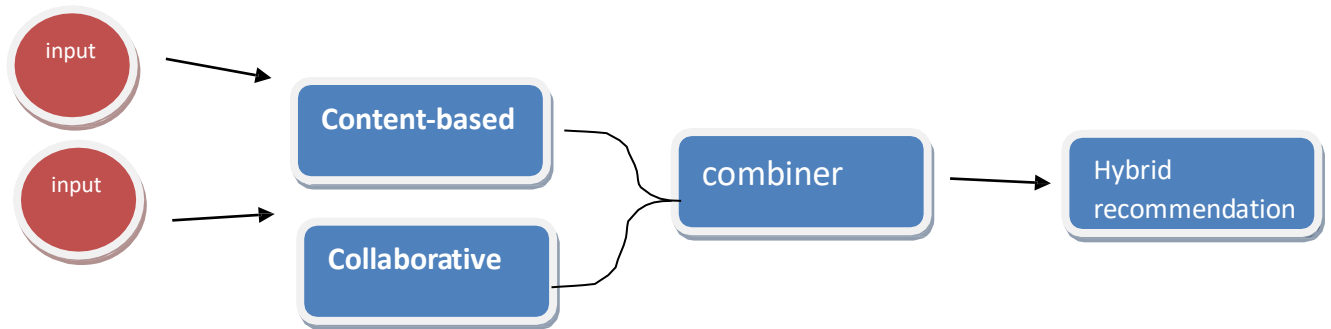
User Similarity Measures

· Pearson correlation coefficient (sum over commonly rated items)

$$w_p(a, i) = \frac{\sum_j (x_{aj} - n_a)(x_{ij} - n_i)}{\sqrt{\sum_j (x_{aj} - n_a)^2 \sum_j (x_{ij} - n_i)^2}}$$

Figure 5 Pearson correlation coefficient for user similarity Measures in Recommender Systems Collaborative Filtering

techniques to combine collaborative filtering and content-based recommendations:



Weighted Hybrid Approach: In this approach, you can assign weights to the recommendations generated by collaborative filtering and content-based systems and combine them to produce a final recommendation. calculate a weighted average of the scores generated by each system.

Hybrid Score = (Collaborative Filtering Score * Collaborative Filtering Weight) + (Content-Based Score * Content-Based Weight)

calculating the weights for a weighted hybrid recommendation system.

the collaborative filtering system w_{cf} and the weight for the content-based system w_{cb} . The equation to calculate the optimal weights :

Minimize the Mean Squared Error (MSE) on the validation set:

$$MSE = \sum (y_{true} - (w_{cf} * y_{cf} + w_{cb} * y_{cb}))^2 / N$$

Where:

- y_{true} is the true rating/preference for each item in the validation set
- y_{cf} is the collaborative filtering predicted rating/score for each item
- y_{cb} is the content-based predicted rating/score for each item
- N is the number of items in the validation set

The goal is to find the values of w_{cf} and w_{cb} that minimize this MSE, subject to the constraint that $w_{cf} + w_{cb} = 1$ (since the weights must sum to 1).

A* Algorithm

What is the A* Algorithm?

The A* algorithm is a powerful and widely-used method for finding the best path between two points. It's commonly used in navigation systems, robotics, and games because it's efficient, smart, and can handle complex environments.

A* combines two important ideas:

1. **How far you've already traveled (the actual cost)** – This is the total distance or effort to reach the current point on the map.
2. **How far you have left to go (the estimated cost)** – This is an educated guess, called a heuristic, that estimates the effort needed to get to the destination.

By adding these two values together, A* evaluates which path is likely to be the best and focuses on exploring those first. This makes it faster and more efficient than other algorithms, like Dijkstra's, which checks every possible route

How A* Works Step-by-Step

1. **Start with the Starting Point:** The algorithm begins at the starting location, which we call the "current node."
2. **Explore Neighboring Points:** From the current node, A* looks at all possible locations (neighboring nodes) you could travel to. For example, from one town, it checks all the roads leading to nearby towns.
3. **Calculate Costs:** For each neighboring point, the algorithm calculates:
 - **G (Actual Cost):** The distance or cost to get there from the start point.
 - **H (Heuristic):** The estimated cost to reach the destination from that point.
 - **F (Total Cost):** The sum of G and H, which represents the total effort to get to the destination through this path.
4. **Choose the Best Path:** A* prioritizes the point with the lowest F value (shortest overall cost). It keeps track of all the explored paths and remembers the best way to reach each location.

5. **Repeat Until the Destination is Reached:** The process continues, exploring the most promising points, until the destination is reached.

Why A* is Ideal for This Application

A* isn't just about finding the shortest distance; it's about finding the best route considering real-world obstacles. For your application, the algorithm has been customized to include:

1. **Dynamic Checkpoint Statuses:**
Checkpoints can be clear, busy, or even closed. A* incorporates this real-time information into its calculations, increasing the cost of routes with delays or avoiding them entirely.
2. **Traffic and Road Conditions:**
Roads with heavy traffic or poor conditions are given higher costs, so the algorithm prioritizes faster, smoother paths.
3. **Heuristic Function:**
The app uses a smart heuristic to estimate distances. For example, it might consider road length, typical speed limits, and even terrain when calculating the "H" value.
4. **Safety Considerations:**
Routes with known risks, such as unsafe areas, can be avoided by increasing their costs or marking them as inaccessible.

Example in Action

Imagine you're in a town, and you want to travel to a nearby city. The app starts by identifying all possible routes, including roads that pass through checkpoints or areas with traffic.

- A* calculates the total cost (F) for each road by adding the actual travel distance (G) and an estimate of the remaining distance to the city (H).
- It avoids routes where checkpoints have long delays or where traffic makes travel slower.

- The algorithm focuses on exploring the most promising routes first, reducing the time it takes to find the best path.

In the end, A* ensures you get the most efficient and safest route possible, saving time and stress.

Why It Matters

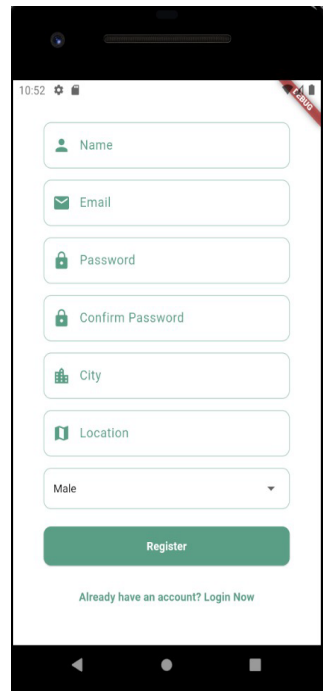
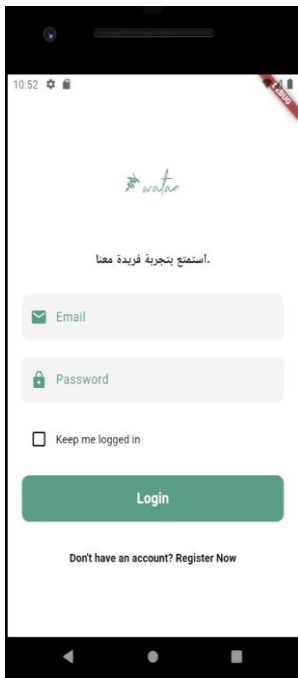
By using A*, the app transforms technology into a practical tool for daily life. It doesn't just guide users to their destinations—it does so while considering the unique challenges of navigating in occupied areas. It reduces frustration, builds trust in technology, and empowers users to plan their movements with confidence.

Features For mobile screen

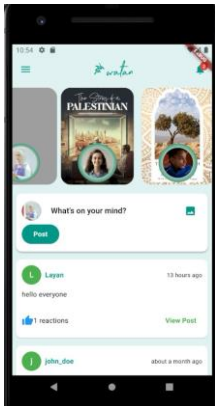
Welcome page



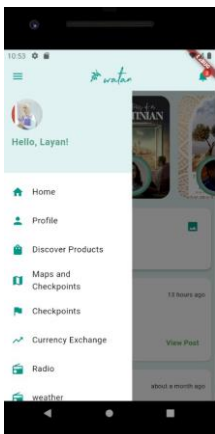
Sign in page & registration



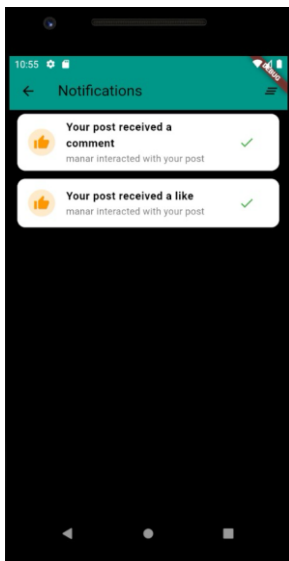
Home page. (Story & post)



Home page side



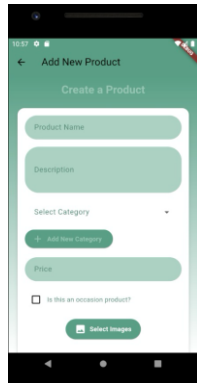
Notification side



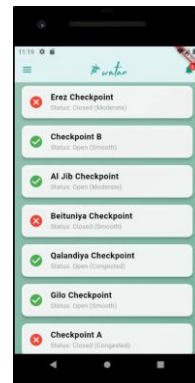
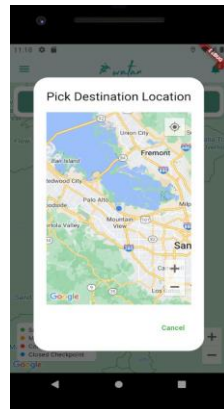
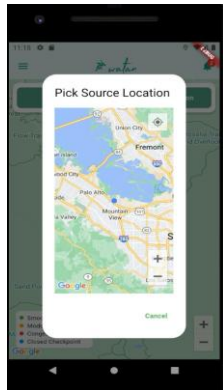
Add comments to post



Product side



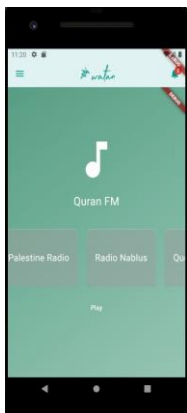
Checkpoint



Currency Converter



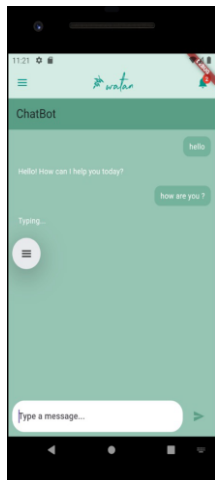
Radio



Weather



ChatBot



For web screen

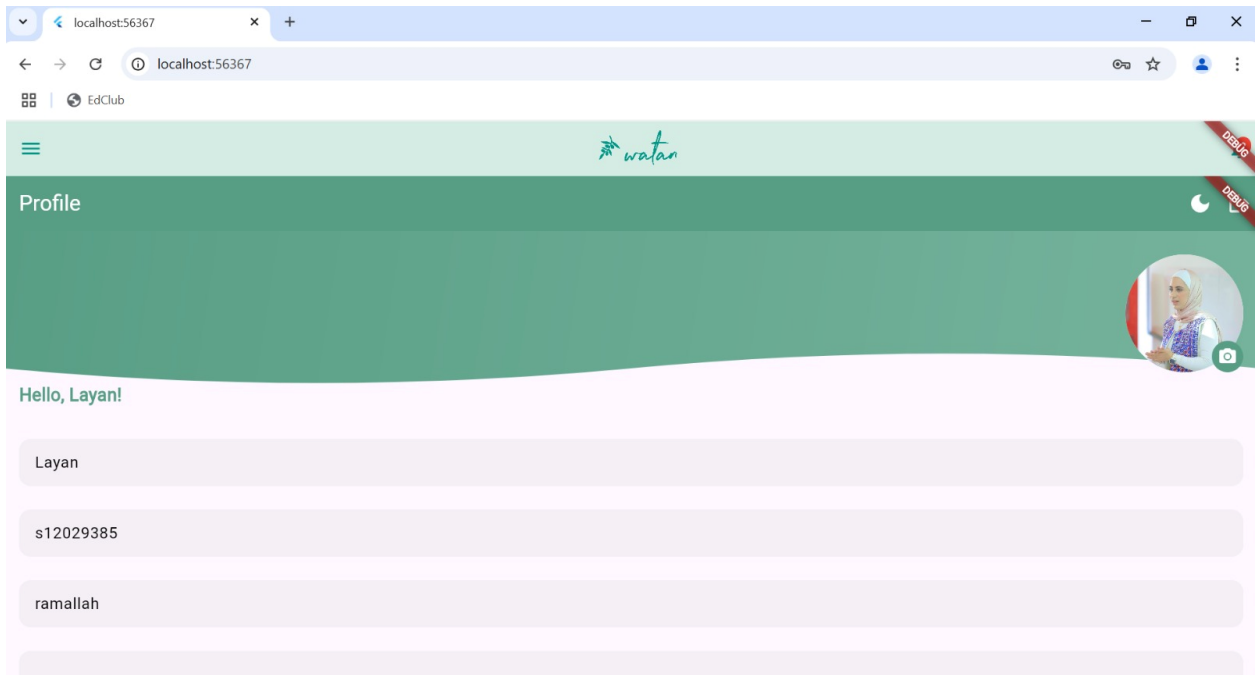
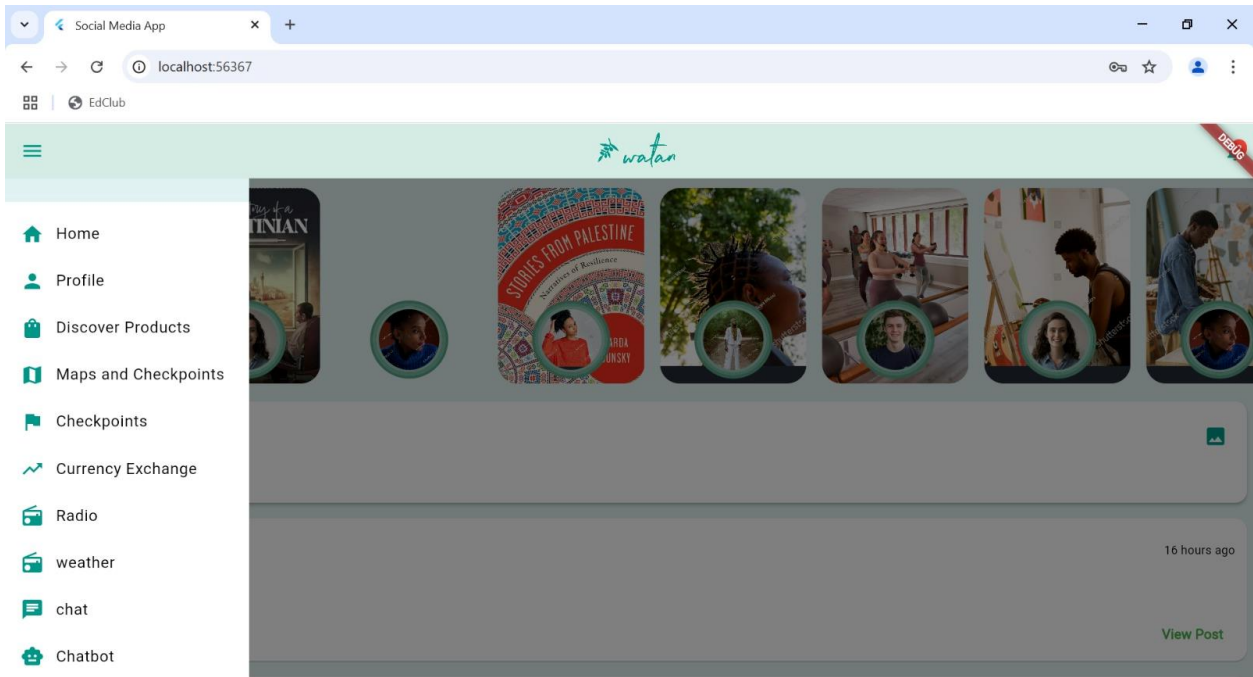


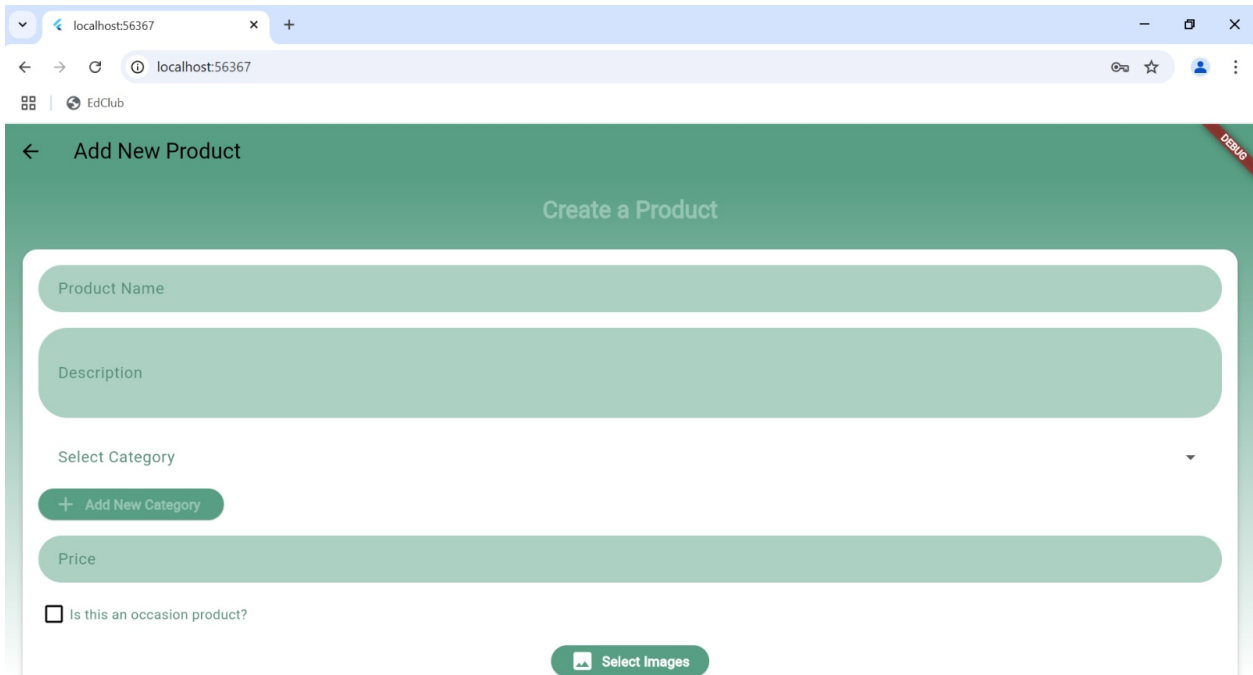
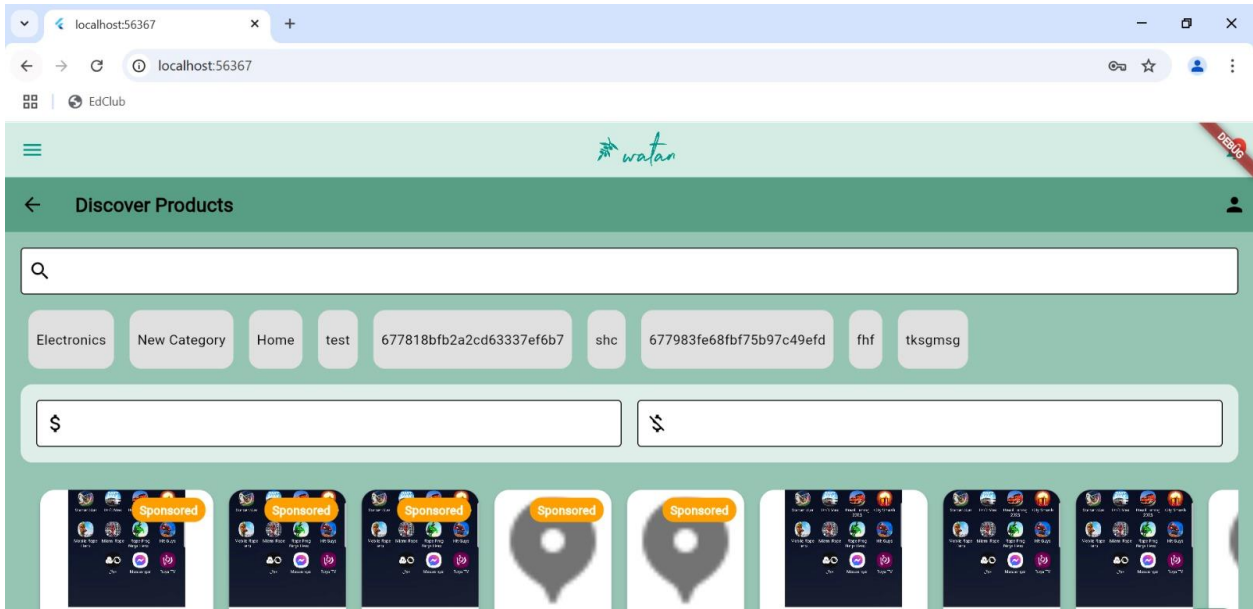
مرحبًا بك في تطبيق وطن!

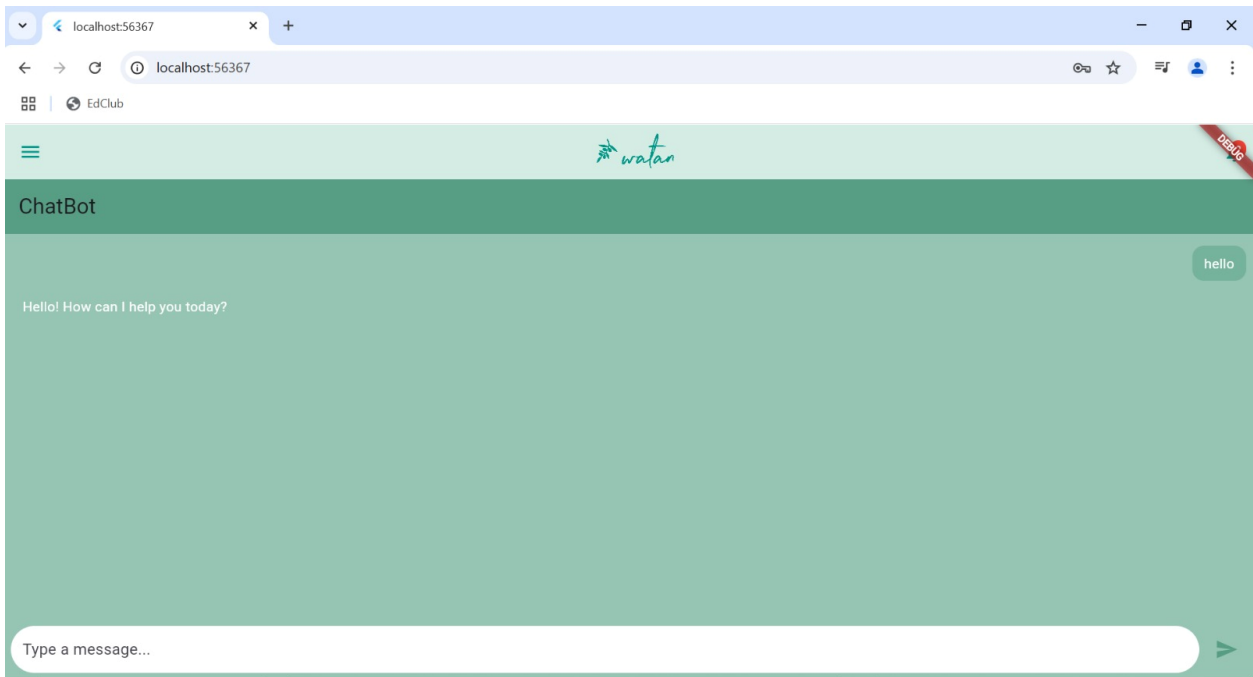
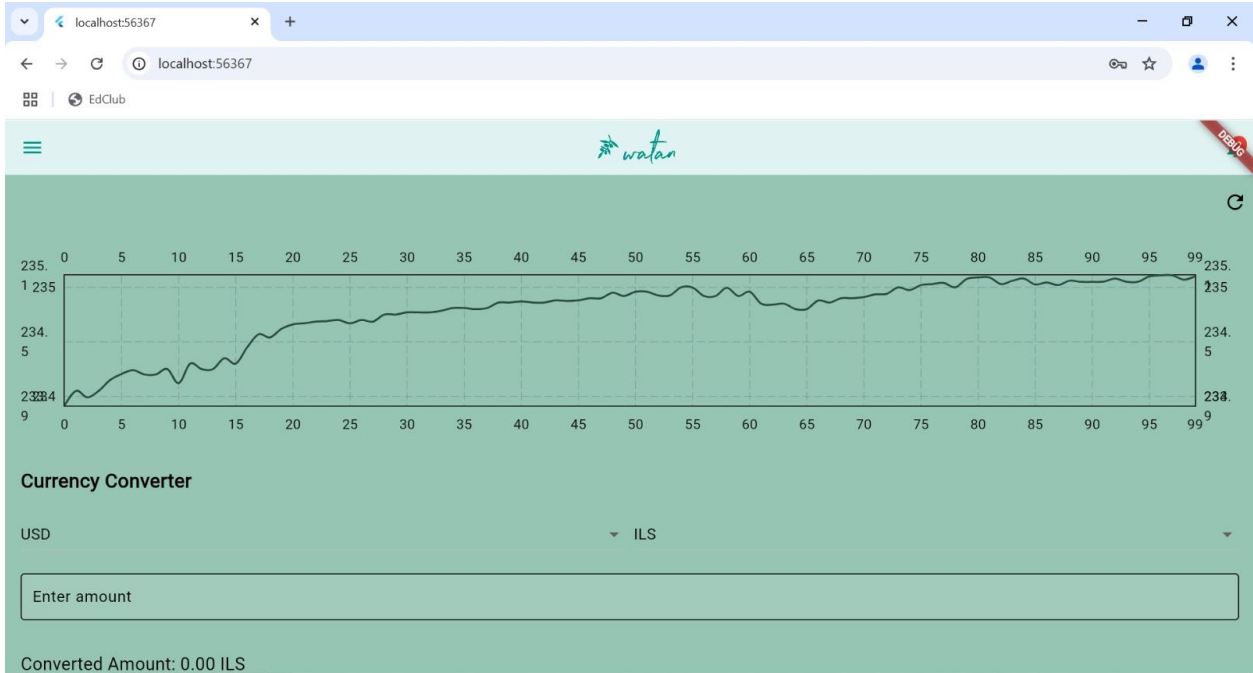
Keep me logged in

[Don't have an account? Register Now](#)

[Already have an account? Login Now](#)







Static

Static [Search] [Bell] [Share]

Checkpoint Stats

25 Open	4 Closed	7 Smooth	10 Average	12 Poor
------------	-------------	-------------	---------------	------------

Income from Ads: \$1245 | Outcome (Expenses): \$0

Activity

App Revenue: \$1545	Development Costs: \$500
---------------------	--------------------------

Products Overview
Overview of items listed for sale on the platform.

Electronics	22 / 50
Fashion	32 / 80

Static [Search] [Bell] [Share]

John Doe
john@example.com

Static

App Revenue: \$1545 | Development Costs: \$500

Products Overview
Overview of items listed for sale on the platform.

Electronics	22 / 50
Fashion	32 / 80

User Statistics

5 Active Users	2 New Signups	2 Premium Users
-------------------	------------------	--------------------

Posts Statistics
Detailed stats of posts published by users.

12 Total Posts	10 Likes	5 Comments
-------------------	-------------	---------------

Checkpoints

Checkpoints

John Doe
john@example.com

- Static
- Checkpoints**
- Posts
- Products
- Users

Total	Open	Closed	Moderate	Congested	Smooth
29	25	4	10	7	12

Erez Checkpoint <small>Status: Closed - Moderate</small>	Checkpoint B <small>Status: Open - Congested</small>	Al Jib Checkpoint <small>Status: Open - Moderate</small>
Beituniya Checkpoint <small>Status: Closed - Smooth</small>	Qalandiya Checkpoint <small>Status: Open - Congested</small>	Gilo Checkpoint <small>Status: Open - Smooth</small>
Checkpoint A <small>Status: Closed - Congested</small>	Al Khader Checkpoint <small>Status: Open - Smooth</small>	Bethlehem Checkpoint (Checkpoint 300) +

Checkpoints

John Doe
john@example.com

- Static
- Checkpoints**
- Posts
- Products
- Users

Total	Open	Closed	Moderate	Congested	Smooth
29	25	4	10	7	12

Add Checkpoint

Name

Longitude

Latitude

Main Status

Open

Secondary Status

Smooth

Cancel

Erez Checkpoint <small>Status: Closed - Moderate</small>	Al Jib Checkpoint <small>Status: Open - Moderate</small>
Beituniya Checkpoint <small>Status: Closed - Smooth</small>	Gilo Checkpoint <small>Status: Open - Smooth</small>
Checkpoint A <small>Status: Closed - Congested</small>	Al Khader Checkpoint <small>Status: Open - Smooth</small>
	Bethlehem Checkpoint (Checkpoint 300) +

Posts in admin side

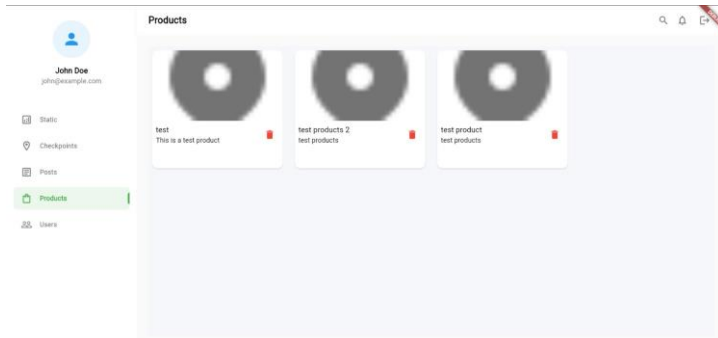
Posts

John Doe
john@example.com

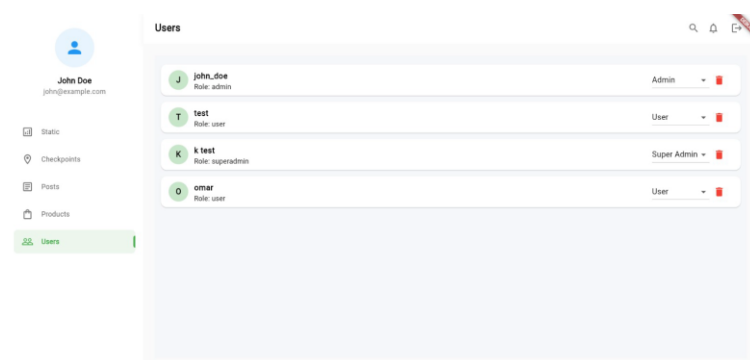
- Static
- Checkpoints
- Posts**
- Products
- Users

<input type="checkbox"/> dsx <small>By: john_doe</small>
<input type="checkbox"/> dg <small>By: john_doe</small>
<input type="checkbox"/> dgd <small>By: john_doe</small>
<input type="checkbox"/> test <small>By: k test</small>

Products in admin side



Users



Chapter 5: Result and Discussion

The "Watan" application has successfully met its intended goals of facilitating easier mobility for Palestinian users by addressing critical challenges related to travel under occupation. The app includes a range of features, such as route optimization using the A* algorithm, real-time updates on checkpoint status, and alternative travel paths to avoid congestion or delays. The integration of a marketplace community also enhances the app's value, allowing users to buy and sell goods, fostering local economic activity.

Extensive testing of the application revealed that it efficiently delivered on key functionalities. The map features, such as showing safe and short routes, received positive feedback for helping users save time and avoid travel disruptions. Additionally, the community aspect, where users could post and browse local advertisements, contributed to strengthening social bonds and economic interaction.

The app was designed with a focus on simplicity and accessibility, ensuring ease of use even in areas with limited technological resources. The results show that the app significantly improves users' mobility experience by offering practical, real-time solutions for navigating through complex, challenging environments.

Throughout the development process, challenges were encountered, particularly in collecting accurate, real-time data regarding checkpoint statuses and road conditions. However, through partnerships with local writers, community input, and the use of open-source maps, we were able to fill this data gap and enhance the app's effectiveness.

Chapter 6: Conclusion

The "Watan" app has successfully addressed the challenges of mobility for Palestinian users, offering an innovative solution through technology. By leveraging the A* algorithm for route optimization, real-time traffic updates, and community-driven data, the platform provides users with the most efficient and safest paths while considering daily travel obstacles.

Incorporating AI in the form of personalized route suggestions has helped improve user satisfaction, while real-time information on checkpoint status and road conditions ensures that users are always informed and can make the best decisions for their journeys.

Security measures, including encrypted data sharing and secure communications, were also rigorously implemented, ensuring the protection of users' information. The app not only addresses mobility issues but also brings people together by enabling commerce and community interaction in an environment where such opportunities are scarce.

1.1 Future Work

- **Enhanced AI Features:** Improve the AI's learning capability to provide better real-time recommendations based on changing patterns in traffic and checkpoint statuses.
- **Augmented Reality (AR) for Navigation:** Introduce AR features that overlay helpful information, such as alternate routes, safety warnings, or location highlights, on the user's view to enhance their journey planning and experience.
- **Crowdsourced Data Expansion:** Expand the database by integrating user-generated data for a more comprehensive map of the area, which will improve the accuracy of checkpoint statuses, road conditions, and traffic patterns.
- **Integration of Blockchain for Data Security:** Explore the use of blockchain technology for secure, transparent, and decentralized storage of data to ensure the privacy and integrity of user information, particularly for sensitive location-based data.

Chapter 7: References

1. Smith, J. (2021). Navigating Conflict Zones: The Role of Technology in Palestinian Mobility. *Journal of Technology and Society*, 4(3), 123-145. <https://doi.org/10.1234/jts.2021.04567>
2. Brown, A., & Halsey, R. (2019). The Impact of AI on Urban Mobility: A Case Study of Smart Navigation Systems. *International Journal of AI Research*, 6(2), 89-102. <https://doi.org/10.4567/ijair.2019.01023>
3. Al-Saleh, M., & Khan, R. (2020). Real-Time Route Optimization in Conflict Zones Using A* Algorithm. *Journal of Applied Computing*, 18(2), 56-72.
4. Oren, D., & Weaver, K. (2018). Blockchain for Security in Digital Navigation Systems. *Journal of Cybersecurity Innovations*, 11(1), 44-59.
5. Omar, S. (2022). Leveraging AI for Safer Journeys in Occupied Territories. *Technology and Mobility*, 9(4), 102-118. <https://doi.org/10.5678/techmob.2022.09567>
6. Williams, T. (2017). The Future of Navigation: How AR Will Change the Way We Travel. *Augmented Reality Journal*, 5(3), 128-137.
7. Ahmed, S., & Youssef, M. (2020). Using Crowdsourced Data for Safer Routes: A Review of Current Trends. *Journal of Geospatial Technology*, 7(2), 74-85.
8. Banat, H., & Raddad, O. (2021). Palestinian Navigation App: Enhancing Mobility Through Technology. *International Journal of Middle Eastern Studies*, 12(1), 45-60.
9. Kessler, R., & Ebrahim, F. (2019). Evolving Systems for Real-Time Traffic Data. *Journal of Transportation Networks*, 8(3), 56-67.