Al-Azhar University

Faculty Of Engineering

Systems And Computers Engineering Department

DAWAM Platform for Everlasting documents security using blockchain technology- Waqf as an example

Proposed By:

Omnia Essam Elden Mohamed Kamal	**ID No: 904052**
Janna Ibrahim Moustafa Elsheshtawy	**ID No: 904114**
Doaa Amin Sami Ahmed Abdulrahim Elfawal	**ID No: 904126**
Salma Ahmed Ali Zaki	**ID No: 904172**
Mai Hassan Muhammad Abu Taleb	**ID No: 904260**
Hager Abobaker Hussein Mohamed	**ID No: 904275**

Supervised By:

Dr. Momtaz Alkholy

Date:

2022/2023

Acknowledgements

It has been a great opportunity to gain such experience like this in real-time projects followed by the knowledge of how to design and analyze real projects. For that, we want to thank all the people who made it possible for us. Special thanks to the graduation Project Unit for all the efforts that they have provided us with all valuable and useful information and for making a clear path for students like us to implement all the education functionality and periods in real-time project design and analysis. Furthermore, we thank our doctors for the interesting lectures they presented and taught us which were a great benefit for all of us. we would like to express our deepest gratitude to our graduation project supervisor:

Dr. Momtaz Alkholy

for all her patience and guidance, continuous encouragement, and supportiveness for us during the whole year. Moreover, I think it’s a duty for us to thank all testing committee members for their generous discussion and encouragement

Abstract

The increasing need for secure data storage has led to the development of Blockchain technology that gives us the ability to store data in an immutable, transparent, secure and decentralized form. Dawam is a platform that harnesses the power of Blockchain to provide everlasting document security. This project focuses on the development of a web and mobile application that allows users to explore and store their documents as Non-Fungible Tokens (NFTs) on the Blockchain.

Having the Blockchain power we decided to explore it in one of the fields that needs it the most witch is Waqf. As Waqfs are everlasting assets that tended to enrich Muslim communities in various aspects over the years. And as we discovered it was suffering from several problems, many of them was because of the document security which led to tedious systems and Waqf recession.

The project takes Waqf documents as an example to show how blockchain technology can be used to ensure the security, immutability, and longevity of important documents. The platform converts documents into NFTs and stores them on a Blockchain, such as Ethereum, making them tamper-proof and permanent.

Dawam also includes a web dashboard to enter documents and approve the applied documents. The platform aims to provide a secure and easy-to-use solution for document storage, making it accessible to everyone.

Dawam's web and mobile applications make it easy for users to explore and store their documents, while the web dashboard simplifies the document approval process all connected with the same API to make sure they are up to date all the time.

In summary, Dawam is a groundbreaking platform that utilizes blockchain technology to provide everlasting Waqfs document security. The project's web and

mobile applications, along with the web dashboard, make it an accessible and user-friendly solution for document storage and everlasting security. Hopping from God to make us among those who save his Waqfs

Table of Contents

Table of Figures

List of Tables

CHAPTER 1: Project Initiation

1.1 Introduction

Waqf documents are a crucial part of Islamic history, providing insights into the social, economic, and religious practices of past generations. However, the preservation of these documents has been a challenge for many years, with issues such as damage, loss, and forgery posing significant threats.

Blockchain technology, with its decentralized, transparent, and immutable nature, has emerged as a promising solution for preserving Waqf documents. Blockchain technology allows for the creation of a tamper-proof and secure environment for storing and managing these documents, ensuring their authenticity and accessibility for future generations.

Overall, this project aims to highlight the importance of preserving Waqf documents, and the potential of blockchain technology for achieving this goal. It is hoped that this work will inspire further research and development in this area and contribute to the preservation of this important part of Islamic heritage for generations to come.

1.1.1 Problem Definition

If we look at our society, we find that saving contracts are essential in all fields. Saving Contracts with the traditional way isn't secure because of that it's more vulnerable to hacking and change. There is no trusted way for saving contracts:

- Contracts could get lost.
- Contracts could be destroyed.
- Contracts could be changed.
- Some contract functions need to offer more trusted systems.

So, we decided to take Waqfs as an example, because Waqf facing lot of problems such as:

- There are no eternal contracts to save waqf documents which are eternal and some of them were lost across history due to different reasons like occupation and change of successions and more.
- Investing in waqf and ongoing charities had fallen back (when they are one of the most powerful aspects of the Islamic economy) due to the lack of trust and difficulty of accomplishment.
- It's very difficult to access waqf documents because they are archived under security protection at Waqfs Ministry.

Through our website and mobile application, we provide a way for saving waqf documents permanently by using Blockchain & decentralized Web 3 technologies. Waqf documents will be more reachable and encourage people to invest in waqf and the ongoing charities.

Therefore, the problem definition for this project is to explore the feasibility of using blockchain for preserving Waqf documents, and to develop a blockchain-based solution that can ensure the integrity and security of these important documents. The project aims to investigate the potential of blockchain in preserving Waqf documents, and to identify the challenges and limitations of implementing such a solution. The project will also develop a blockchain-based solution that considers the unique features and requirements of Waqf documents, and that can provide a reliable and secure system for preserving these important documents.

1.1.2 Project Objectives

Objective 1: Develop a functional web and mobile application for the storage of documents as NFTs on the Blockchain, with a user-friendly interface that allows users to explore and apply documents for storage.

Objective 2: Create a web dashboard that allows for the easy entry of documents and the approval of applied documents, with a clear and efficient workflow for document management.

Objective 3: Integrate Blockchain technology, such as Ethereum, into the platform to ensure the immutability and security of stored documents, measured by the absence of document tampering or unauthorized access.

Objective 4: Implement smart contract functionality to provide automated document storage and management, allowing for streamlined processes and efficient document tracking.

Objective 5: Ensure scalability of the platform, measured by its ability to handle an increasing number of users and stored documents without compromising performance or security.

Objective 6: Provide user support and documentation to ensure the platform is easy to use and accessible to a wide range of users and fulfills user satisfaction.

Objective 7: giving public Muslims easier access to waqf documents which would enrich waqf in Muslim’s communities and enable collaboration between different communities.

1.2 Current and existing systems

1.2.1 Current Systems

The current system for preserving Waqf documents varies across different regions and countries. In some areas, Waqf documents are stored in libraries or archives, while in others, they are kept in private collections or with Waqf institutions themselves. These documents are often handwritten on paper, and their preservation can be challenging due to their fragility and susceptibility to damage from natural disasters, wars, or simply from aging.

Several efforts have been made to preserve Waqf documents, including digitization and microfilming. Digitization involves scanning the documents and converting them into digital format, while microfilming involves photographing the documents and storing the images on microfilm. These methods have the advantage of creating a backup copy of the original documents, but they also have their limitations.

One of the main limitations of the current system for preserving Waqf documents is the risk of data corruption and manipulation. Digitized or microfilmed documents can be altered or deleted, either intentionally or accidentally, resulting in the loss or distortion of the information they contain. Additionally, these methods require significant resources and infrastructure to implement, which may not be available in all areas.

Another limitation of the current system is the lack of transparency and accessibility. Waqf documents are often stored in libraries or archives that may not be easily accessible to the public or researchers. This can limit the ability of stakeholders to monitor the preservation process and access the information they need.

In conclusion, the current system for preserving Waqf documents has limitations that can affect the integrity, accessibility, and transparency of these important documents. The use of blockchain technology has the potential to address these limitations by providing a decentralized, immutable, and transparent system for preserving Waqf documents.

1.2.2 Existing Systems

National unified portal (KSA's national source for government services and information)

A portal that provides electronic services, including a request to register a waqif, and it is an e-service provided by the General Authority for Endowments, which allows the receiving of endowment registration requests and processing them as necessary by the procedures enforced by the Authority, in addition to the service of the endowment or the beholders registration in endowment.

This service enables the issuance of an endowment certificate through the receipt of applications for issuance, renewal, and amending the endorsement in the Kingdom of Saudi Arabia and processing it as required by the procedures of GAA. Applications are received and followed up through the official and allocated channels.

Service implementation steps:

- Registration.
- Proccing Request.

- Return the request to the customer when there is feedback on the customer's request.
- Application Review
- Audit and certification

The website of the General Authority of Islamic Affairs and Endowments for the United Arab Emirates

The website that provides electronic waqf services, including:

- Taking care of waqf assets during endower's life of and after their death - request waqf report:

  The Awqaf Department takes care of the Waqf revenues in coordination with the donor or his/her heirs in addition to maintaining the Waqf assets. It also provides each donor with periodic financial reports relating to the endowment revenues (villa, building, farm, piece of land) and disbursement channel(s) selected in compliance with the request of donor or his/her heirs.

- Establishing new endowment:

  This service allows those who wish to donate or establish an endowment to register their Waqf.

  Service implementation steps:

  - Issue Waqf establishment from court.
  - Visit the nearest Awqaf branch to register the waqf.

E Donation:

This service allows individuals to make electronic donations through one of the following channels:

GAIAE's App on smartphone and tablets (AWQAF) where the donor can register his/her data and select the type of disbursement channel and make the payment either using dirham or credit card.

Online via GAIAE's website where the donor can register, select disbursement channel, and make payment either using dirham or credit card.

Waqf Coupons:

Waqf disbursement channels coupons are sold through sales representatives. Coupons are printed with the type of disbursement channel marked on them (like: Healthcare, orphans, holy Quran printing…etc.). The service allows the donor to choose the amount to be donated and the disbursement channel.

FINTERRA WAQF CHAIN

The Finterra ecosystem offers an inclusive platform to consumers through various verticals that support Islamic Social Finance. Finterra's flagship products, the Islamic Social Finance Suite has been developed to revitalize the Islamic Social Finance system for the digital age using Blockchain technology. With relevant regulatory compliance built into the products and services, it helps solve core challenges in unlocking and integrating options for Capital raise, Waqf management, and Asset management while offering robust reporting capabilities.

How it Works:

- Cash Waqf

  Their platform allows for anyone to make cash donations towards projects having social impact.

- Waqf Project

  Their platform gives equal weight to social and environmental factors vis-a-vis financial returns.

1.3 Literature Review

1.3.1 Background

The preservation of Waqf documents is a critical issue, as these documents provide evidence of endowments that have been established for centuries. Waqf documents can be damaged or lost due to natural disasters, wars, or simply from aging. Therefore, several efforts have been made to preserve Waqf documents, including digitization and microfilming. However, these methods have their limitations, including the risk of data corruption and the possibility of manipulation.

Blockchain technology has emerged as a potential solution for preserving Waqf documents. Blockchain is a decentralized and immutable database that can store data without the need for intermediaries. It uses cryptography to secure data and prevent unauthorized access. The use of blockchain for preserving Waqf documents has several advantages, including:

- Data immutability: Once data is stored on the blockchain, it cannot be altered or deleted, ensuring the integrity of the Waqf documents.

- Decentralization: Blockchain is a decentralized system, which means that there is no central authority that can manipulate or alter the data.
- Transparency: The use of blockchain provides transparency in the preservation process, allowing stakeholders to monitor the preservation process.
- Security: Blockchain technology uses cryptography to secure data, making it highly resistant to hacks and cyber-attacks.

1.3.2 Several studies have explored the potential of using blockchain for preserving Waqf documents.

- Use of Blockchain Applications in Developing Waqf: A Platform by Finterra Companies as a Modal Study (Sasse, 2019). The research tried to employ the experience of the Finterra Foundation in adopting the blockchain methodology in the endowment system, but it did not reveal the real relationship that this methodology could represent in the endowment performance, although the paper drew some points in general without detailing.

  In addition, the details of what can be demonstrated from the application of this blockchain methodology to the reality of the work of the Awqaf Foundation were not discussed from the legal point of view.

- Endowment and blockchain technology: Invest and Finance from the Sharia Perspective (Al-Salahat, 2021). This research aims to define blockchain methodology and highlight the blockchain relationships in the framework of waqf system and its main processes. It also aims to show how blockchain methodology can facilitate developing the work of the waqf institution further, and it clarifies the legal framework of interests and/or corruption.

1.3.3 Our Product Features

Through our website and mobile application, we provide a way for saving waqf documents permanently by using Blockchain & decentralized Web 3 technologies. Waqf documents will be more reachable and encourage people to invest in waqf and the ongoing charities.

1.4 Stakeholders List

Stakeholder	Interest	Importance
Dr. Momtaz Alkholy	Team supervisor and guide of the development process	High
Project Team Members	Responsible for requirement gathering and analysis, designing, developing, testing, and deploying the platform	High
People (Users)	The people who will be using the application	High
Waqf Institutions	Secure and everlasting storage of their documents through the platform	High

Table 1-1:Stakeholders

1.5 Proposed Scope & Process Model

1.5.1 Proposed Scope

1. Research and analysis of the requirements for the platform, including the needs of the users and stakeholders.
2. Design and development of the web and mobile application for exploring the stored documents and applying documents to be stored.
3. Design and development of a web dashboard for entering documents and approving the applied documents.
4. Design and development of app database and API to handle requests from web, mobile application, and web dashboard.
5. Integration of Blockchain technology (Ethereum as an example) for storing the documents as NFTs.
6. Implementation of security measures to ensure the safety and confidentiality of the documents.
7. Testing and validation of the system to ensure its functionality and usability.
8. Documentation of the system, and data gathering for real examples.

1.5.2 Process Model

The process model for the project will follow an agile approach, with iterative and incremental development cycles. The Agile approach is well suited to software development projects that require flexibility and adaptability to changing requirements and user feedback.

1.6 Project excluded and Project constraints

1.6.1 Project excluded

- The project does not involve the physical preservation of Waqf documents, such as restoration or repair of damaged documents.
- The project does not involve the development of a blockchain network from scratch, but rather the use of an existing blockchain technology.

1.6.2 Project constraints

- There was difficulty finding documents as they were at Waqfs ministry under high security protection. We could only find some little examples through the internet and some books.
- It was not easy to find resources to study blockchain.
- Time was too short for studying new technology through the educational year.
- Cost of the server.
- Integration with Waqf ministry will not be delivered by the end of this project due to security issues and time limitation.

CHAPTER 2: Planning and Requirements

2.1 Planning

2.1.1 Scope Initiation (WBS)

TASK TITLE	DURATION	START DATE	Finish	Predecessors	Resource Names
DAWAM					
Project Conception & Initiation	60	01/11/2022	31/12/2022		
Deep Research in Waqf System	60	01/11/2022	31/12/2022		
Current and existing systems	14	07/11/2022	21/11/2022		
Literature Review	14	14/11/2022	28/11/2022		
Stakeholders	8	29/11/2022	07/12/2022		
Problem Definition	22	22/11/2022	14/12/2022		
Project Objectives	6	15/12/2022	21/12/2022		
Project Definition & Planning	9	22/12/2022	31/12/2022		
Scope Initiation	5	22/12/2022	27/12/2022		
Resource Planning	5	22/12/2022	27/12/2022		
Risk List Analysis	5	22/12/2022	27/12/2022		
Information Gathering	5	22/12/2022	27/12/2022		
Requirement Analysis	6	22/12/2022	28/12/2022		
Project time Plan	3	28/12/2022	31/12/2022		
Project Launch & Execution	228	15/11/2022	01/07/2023		
Studying Blockchain and other needed technologies	151	15/11/2022	15/4/2023		
UI and UX Design	62	15/12/2022	15/2/2023		
Choosing Color Pallet	6	15/12/2022	21/12/2022		
Creating Logo	6	15/12/2022	21/12/2022		
Web UI and UX	55	22/12/2022	15/2/2023	3.2.1	
Design Home Page	55	22/12/2022	15/2/2023		
Design About Page	55	22/12/2022	15/2/2023		
Design Waqfs Explore Page	55	22/12/2022	15/2/2023		
Design Single Waqf Page	55	22/12/2022	15/2/2023		
Design Search Page	55	22/12/2022	15/2/2023		
Design Contact Page	55	22/12/2022	15/2/2023		
Design Login Page	55	22/12/2022	15/2/2023		
Design Dashboard Page	55	22/12/2022	15/2/2023		

Design Add Waqf Page	55	22/12/2022	15/2/2023		
Mobile UI and UX	55	22/12/2022	15/2/2023		
Design Splash Screen	55	22/12/2022	15/2/2023		
Design Landing Screen	55	22/12/2022	15/2/2023		
Design About Screens	55	22/12/2022	15/2/2023		
Design Waqfs Explore Screen	55	22/12/2022	15/2/2023		
Design Single Waqf Screen	55	22/12/2022	15/2/2023		
Design Search Screen	55	22/12/2022	15/2/2023		
Design Contact Screen	55	22/12/2022	15/2/2023		
Design Add Waqf Screen	55	22/12/2022	15/2/2023		
Web Front-end Implementation	59	15/2/2023	15/4/2023	3.2	Omnia Essam, Doaa Amin,Mai Hassan
Project Architecture	6	15/2/2023	21/2/2023		
Implementing Shared Components	6	15/2/2023	21/2/2023		
Implementing Different Pages Components	37	22/2/2023	31/3/2023		
Implementing Routing	37	22/2/2023	31/3/2023		
Implementing Login Components	37	22/2/2023	31/3/2023		
Implementing Dashboard Components	37	22/2/2023	31/3/2023		
Implementing Add/View/Edit Waqf Components	37	22/2/2023	31/3/2023		
Connecting Website with API	101	04/01/2023	15/4/2023	3.4	
Connection Dashboard With API	101	04/01/2023	15/4/2023		
Mobile Implementation	28/02/1900	15/2/2023	15/4/2023		Salma Ahmed, Hager Abobaker
Project Architecture	6	15/2/2023	21/2/2023		
Implementing Shared Components	6	15/2/2023	21/2/2023		
Implementing Different Screens Components	37	22/2/2023	31/3/2023		
Implementing navigation	37	22/2/2023	31/3/2023		
Implementing Add Waqf Components	37	22/2/2023	31/3/2023	3.4	
Connecting Mobile With API	14	01/04/2023	15/4/2023		

Back-end Implementation	20/02/1900	15/2/2023	07/04/2023		Janna Ibrahim
Project Architecture	6	15/2/2023	21/2/2023		
Project Design Patterns	6	15/2/2023	21/2/2023		
Database Design	6	15/2/2023	21/2/2023		
Database Implementation	6	22/2/2023	28/2/2023		
Implementing Different Get Requests	30	01/03/2023	31/3/2023		
Implementing Post Requests	30	01/03/2023	31/3/2023		
Implementing Put Requests	30	01/03/2023	31/3/2023		
Implementing Delete Requests	30	01/03/2023	31/3/2023		
Creating API Documentation	6	01/04/2023	07/04/2023		
Blockchain Smart Contracts Development	74	15/2/2023	30/4/2023		
Creating ERC721 NFT creating contract	74	15/2/2023	30/4/2023		
Designing NFT metadata format	74	15/2/2023	30/4/2023		
Developing contract functions for exploring the NFTs and their Metadata	74	15/2/2023	30/4/2023		
Choosing Decentralized Storage to store the Waqf Documents	74	15/2/2023	30/4/2023		
Connecting with the website	74	15/2/2023	30/4/2023		
Project Testing and Refinement	61	01/05/2023	01/07/2023		
Project Launch	91	01/04/2023	01/07/2023		
Choosing server and uploading backend APIs	6	01/04/2023	07/04/2023		
Adding mobile app to app store	6	25/6/2023	01/07/2023		
Uploading web frontend to the server	6	25/6/2023	01/07/2023		
Adding Blockchain contracts to Test/main nets	6	25/6/2023	01/07/2023		
Project Closing	36	25/5/2023	30/6/2023		
Project Reviewing	29	01/06/2023	30/6/2023		
Project Documenting	31	25/5/2023	25/6/2023		

Table 2-1:WBS

2.1.2 Gantt Chart

	11		12		1		2		3		4		5	6
Deep Research in Waqf system	🟧	🟧	🟧	🟧										
Study Blockchain Technology and other needed technologies		🟥	🟥	🟥			🟥	🟥	🟥	🟥	🟥			
Design project diagram and UI/ UX				🟪			🟪							
Blockcahin Development				🟧			🟧	🟧	🟧	🟧	🟧			
Front-End Development								🟥	🟥	🟥	🟥			
Back-End Development								🟪	🟪	🟪	🟪	⬛	⬛	
Testing and modifing									🟧	🟧	🟧	🟧	🟧	
English Project Report		🟥	🟥	🟥			🟥	🟥	🟥	🟥	🟥	🟥	🟥	

Table 2-2:Gantt Chart

2.1.3 Resource Sheet

Resource Name	Resource Type	Cost
Omnia Essam	Front-End	Free
Janna Ibrahim	Back-End	Free
Doaa Amin	Front-End	Free
Salma Ahmed	Mobile	Free
Mai Hassan	Front-End	Free
Hager Abubaker	Mobile	Free
Figma	Material	Free
Remix	Material	Free
VS Code	Material	Free
Visual studio	Material	Free
SQL Server Management Studio	Material	Free
Postman	Material	Free
Swagger	Material	Free
Android Studio	Material	Free
Microsoft Word	Material	Free
Laptop	Material	Free
Smartphone (Android Based)	Material	Free

Table 2-3:Resource Sheet

2.1.4 System Development Requirements

The following table will demonstrate the resources that will be required throughout the development process of this project, including Human Resources, Software, and Hardware, in addition to a cost overview of the project.

Resource Type	Resources
Human Resources	•Front-End Developer •Back-End Developer •Blockchain Developer •UI/UX Designer
Software	• Figma • Remix • VS Code • Visual studio • Postman • Swagger • SQL Server Management Studio • Android Studio • Microsoft Word
Hardware	• Laptop • Smartphone (Android based)

Table 2-4: System Development Requirements

2.1.5 Cost Estimation and Budgeting

The following table will contain cost estimations for the different resources that will be used in the development process of this project:

Materials		
	Name	Cost
1	Figma	Free

2	Remix	Free
3	VS Code	Free
4	Visual studio	Free
5	SQL Server Management Studio	Free
6	Postman	Free
7	Swagger	Free
8	Android Studio	Free
9	Microsoft Word	Free
10	Laptop	Free
11	Smartphone (Android Based)	Free
	Total Cost	0

Table 2-5:Cost Estimation

2.1.6 Risk List

Technical Risks

1. Blockchain technology is still relatively new and evolving, which may lead to unexpected technical challenges and difficulties in development and implementation.
2. Integration with Ethereum blockchain may result in technical issues that could delay the project timeline.
3. The use of NFTs may result in unexpected technical challenges and can be difficult to implement.

Time Risks

1. The project timeline may be impacted by unforeseen technical challenges, which could result in delays.

2. A delay in one task may have a cascading effect on other tasks, which could result in a delay in the overall project timeline.
3. Changes in requirements or scope could result in delays in development and implementation.

Resource Risks

1. Limited availability of skilled blockchain developers and designers could result in a shortage of resources, which may impact project timeline.
2. Lack of funding may impact the project timeline and resources available for development and implementation.

2.2 Requirements

2.2.1 Information Gathering

Brainstorming

In this project, we decided to start information gathering with simple brainstorming sessions aiming to get to a better understanding of the problem we are set out to deal with in this project and the requirements that we need to implement during the development process, reaching a solid base of requirement specification that we could build on as we go.

Research

We reviewed academic papers, industry reports, and online resources to understand the challenges faced by the Waqf industry in managing its documents. We also researched the features and functionalities of existing Blockchain-based document management platforms to understand the best practices in the industry.

Stakeholders Interviews

We conducted interviews with stakeholders involved in the Waqf industry and Blockchain technology. These interviews aimed to gather insights into their needs, expectations, and specific requirements related to a platform that provides everlasting documents security using Blockchain technology.

Expert Consultations

Discussions and consultations were held with experts in Awqaf, Blockchain and UI/UX designer. These experts provided insights, recommendations, and best practices related to the technologies and methodologies to be employed in the project.

2.2.2 Functional Requirements

1. **Document storage:** The system should provide a secure and decentralized environment for storing Waqf documents. The system should allow for the storage of various document types, including text, images, and audio.
2. **Document management:** The system should allow for the management of Waqf documents, including adding, editing, and deleting documents. It should also allow for the tracking of document changes and the creation of a document history.
3. **Document authentication:** The system should provide a method for authenticating the Waqf documents stored in the system, ensuring that they are not tampered with or altered in any way. This can be achieved using digital signatures or other cryptographic methods.
4. **Access management:** The system should provide access management features, allowing for the control of who can access the Waqf documents

stored in the system. This can be achieved using permissions and access controls.

5. **Transparency:** The system should be transparent, allowing for the tracking of changes to the Waqf documents stored in the system. This can be achieved using the transparency features of blockchain technology.
6. **Interoperability:** The system should be interoperable, allowing for the integration of various blockchain protocols and smart contracts. This will allow for customization of the system to meet the unique needs of Waqf documents.
7. **User interface:** The system should provide a user-friendly interface for accessing and managing the Waqf documents stored in the system. The interface should be intuitive and easy to use for both technical and non-technical users.

2.2.3 Non-functional Requirements

1. **Security:** The system should provide a high level of security to ensure the confidentiality, integrity, and availability of the Waqf documents stored in the system. This can be achieved using encryption, access controls, and other security measures.
2. **Scalability:** The system should be scalable, allowing for the addition of new Waqf documents and the expansion of the system to accommodate growing storage needs.
3. **Performance:** The system should have high performance, allowing for fast and efficient access to the Waqf documents stored in the system. This can be achieved using optimized data structures and algorithms.

4. **Reliability:** The system should be reliable, ensuring that the Waqf documents stored in the system are always available and accessible. This can be achieved using redundancy and failover mechanisms.
5. **Compatibility:** The system should be compatible with existing technologies, allowing for the integration of existing Waqf document databases and other systems.
6. **Usability:** The system should be easy to use, with a user-friendly interface and intuitive features for accessing and managing the Waqf documents stored in the system.
7. **Compliance:** The system should comply with relevant legal and regulatory requirements, including data protection and privacy laws.

2.3 Use cases

2.3.1 AddWaqf

When a user wants to endow something in the Ministry of Endowments, he must go to the Ministry and then meet the Admin, and then the Admin adds the Waqf to the system.

Use Case Name	Add Waqf
Precondition	The person must fulfill all the requirements of the Ministry of Awqaf
Flow of Events	Open system Enter the add Waqf in order to add the required endowment

	Fill in all required data(waqf name , founder Name , date , description , ….) Click save to send data
Alternatives	-
Post Condition	Complete the conditions of the Ministry of Awqaf
Exceptions	Wrong data entry (missing fields, short password, incorrect email format).

Table 2-6:Add Waqf Use Case

2.3.2 EditWaqf

Use Case Name	EditWaqf
Precondition	_
Flow of Events	Open system Enter the edit Waqf in order to edit the required Admin makes the modification he wants Click save to send data
Alternatives	Enter id for waqf and edit waqf
Post Condition	Id Waqf
Exceptions	Wrong the request to send data for Waqf ID

Table 2-7:Edit Waqf Use Case

2.3.3 ViewWaqf

Use Case Name	ViewWaqf
Precondition	_
Flow of Events	Open system Enter the view Waqf in order to view the required
Alternatives	Enter id for waqf and view waqf
Post Condition	Id Waqf
Exceptions	Wrong the request to send data for Waqf ID

Table 2-8:View Waqf Use Case

2.3.4 Sign-up

Use Case Name	User Sign-up
Actor	Users, System
Precondition	_
Flow of Events	1. The user clicks on the signup page. 2. The user enters his/her data (name, date of birth, profile picture, email, and password). 3. The user clicks the signup button.

	4. The system verifies the data entered (no empty fields, the password is not too short, the email format is correct and the email does not already exist in the database). 5. The system creates the account. 6. The system redirects the user to the sign-in page.
Alternatives	The user chooses the “sign up via Google” option. The user picks the Google account he/she wants to use. The user is redirected to the signup page with some fields already filled. The user clicks the signup button. 5. The system creates the account.
Post Condition	The system creates the user account and displays the Search page.
Exceptions	Wrong data entry (missing fields, short password, incorrect email format).

Table 2-9:Sign Up Use Case

2.3.5 Sign-in

Use Case Name	User Sign-in
Precondition	The user already has an account
Flow of Events	1. The system displays the sign-in page. 2. The user enters his/her email and password. 3. The system displays the sign-in page. 4. The user enters his/her email and password.
Alternatives	_
Post Condition	The system greets the user and transfers him/her to the SearchPage.
Exceptions	The email or password entered is incorrect or does not match the database.

Table 2-10:Sign in Use Case

2.4 Domain Diagram

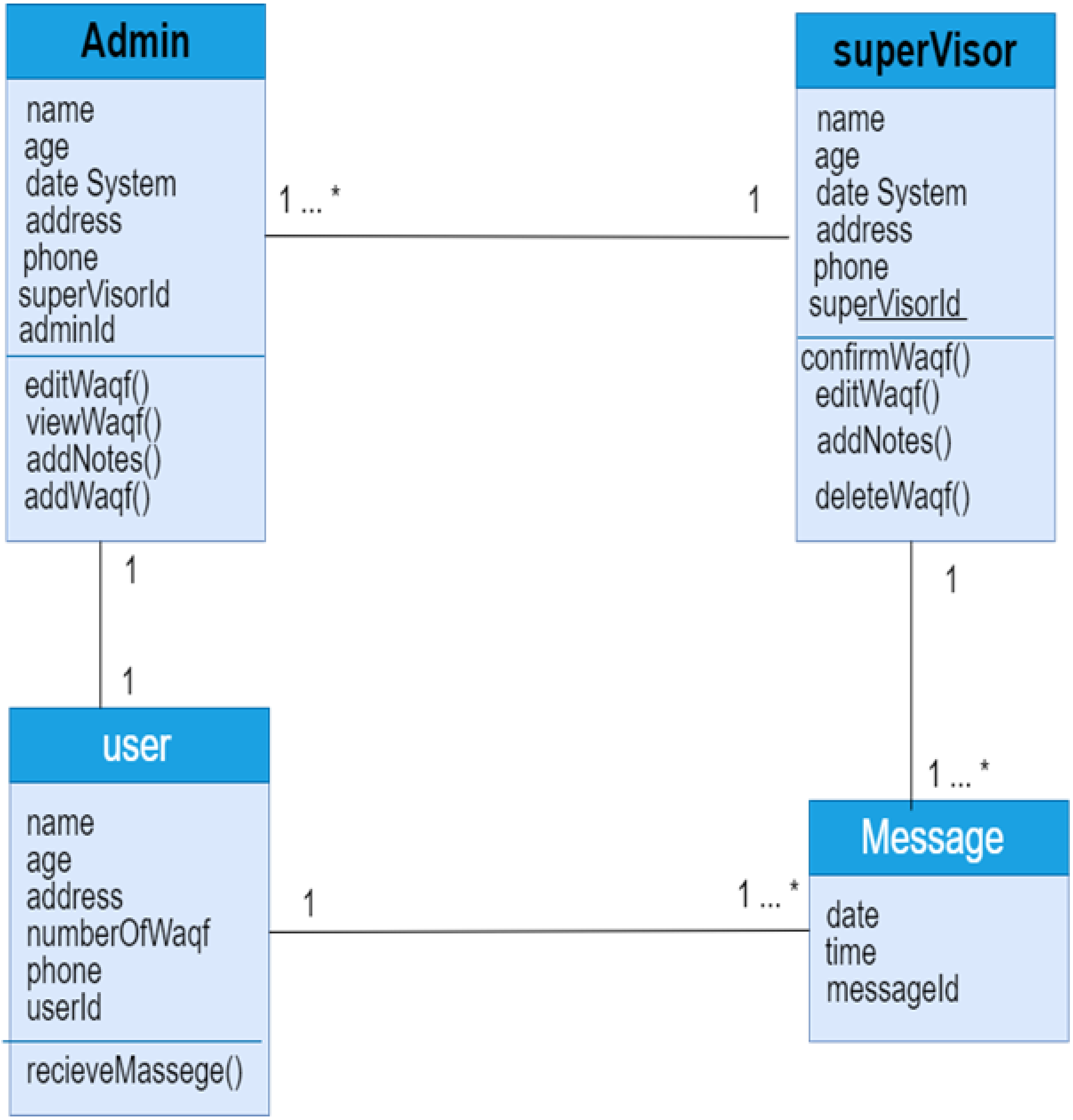


Figure 2-1:Domain Diagram

CHAPTER 3: Project Analysis and Design

3.1 Actor-goal List

Actor	Goal
Admin	○ **AddWaqf** ○ **EditWaqf** ○ **ViewWaqf** ○ **AddNotes**
Supervisor	○ **ConfirmWaqf** ○ **DeleteWaqf** ○ **EditWaqf** ○ **AddNotes**
Researcher	○ **Login** ○ **SignUp** ○ **FilterInWaqf**
Client	○ **RecieveMassege**

Table 3-1:Actor Goal List

3.2 Use Case Diagram

The diagram shows the actors and use cases of the system extracted from the functional list in section ().

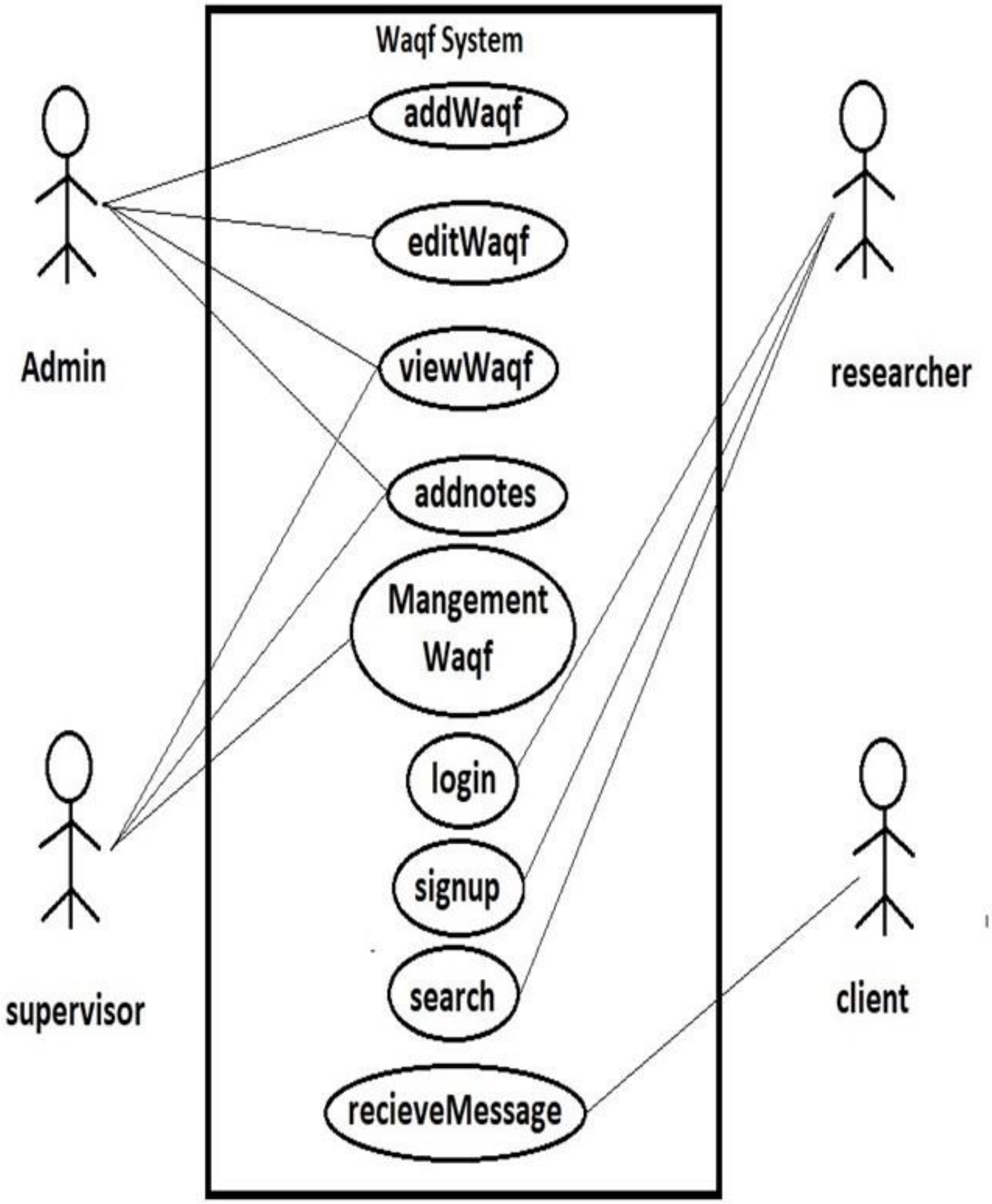
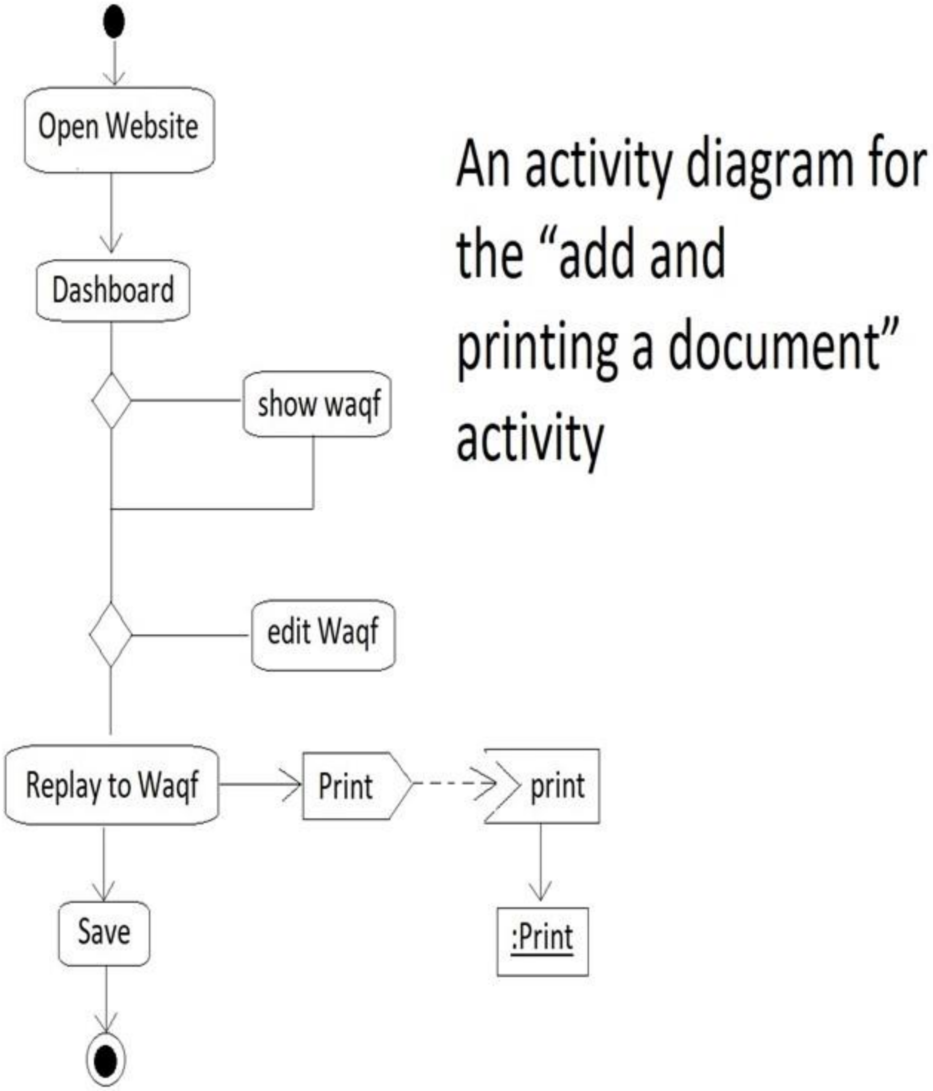


Figure 3-1:Use Case Diagram

3.3 Activity Diagram



An activity diagram for the “add and printing a document” activity

Figure 3-2:Activity Diagram

3.4 Sequence Diagrams

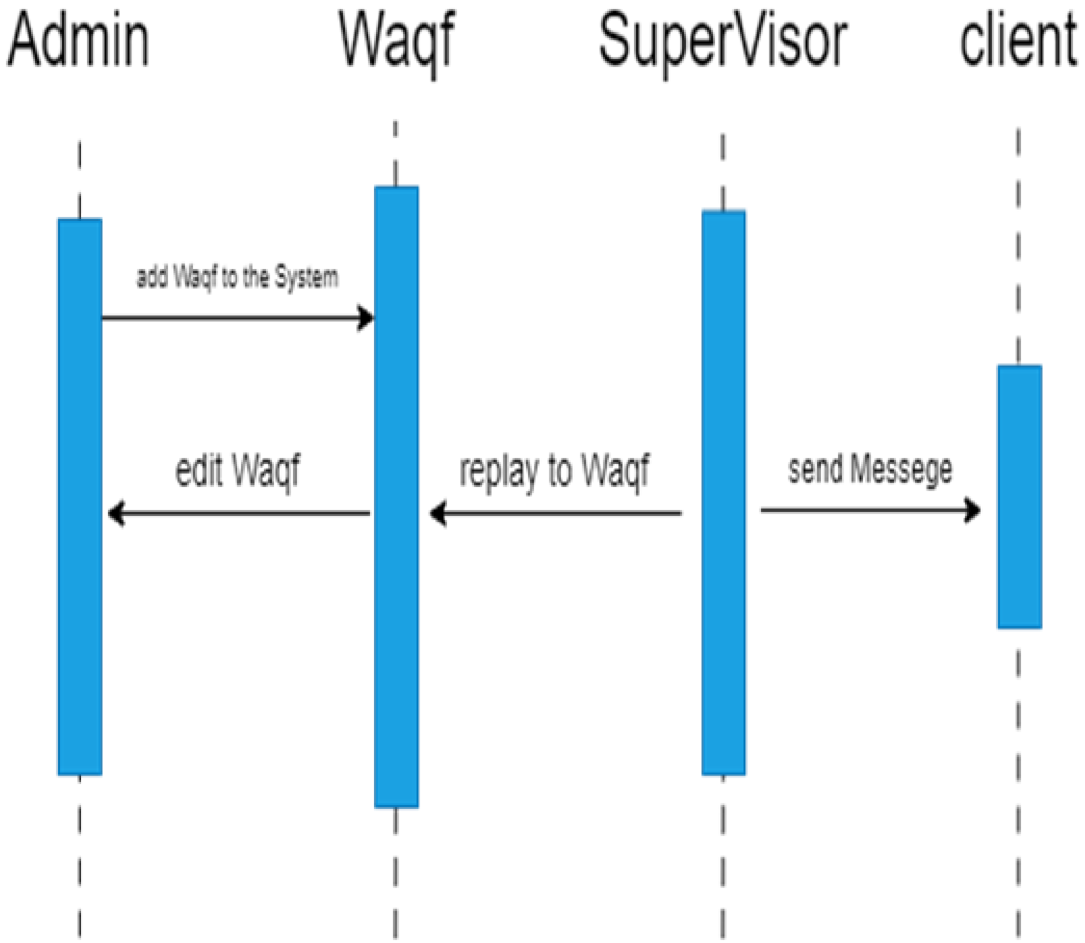


Figure 3-3:Sequence Diagram

3.5 State Diagram

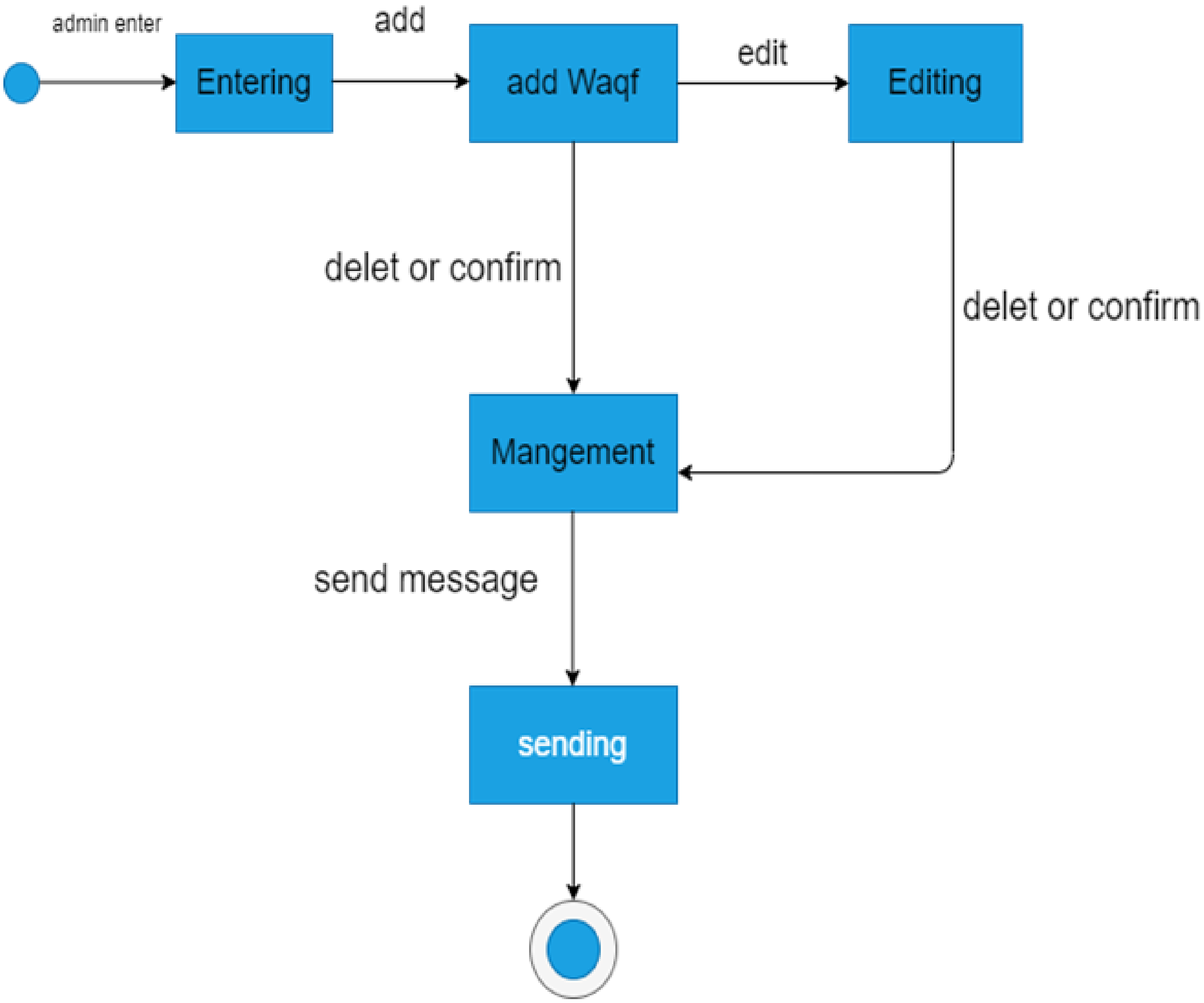


Figure 3-4:State Diagram

3.6 Design Class Diagram

Design Class Diagram: Show's all classes in our system, which a

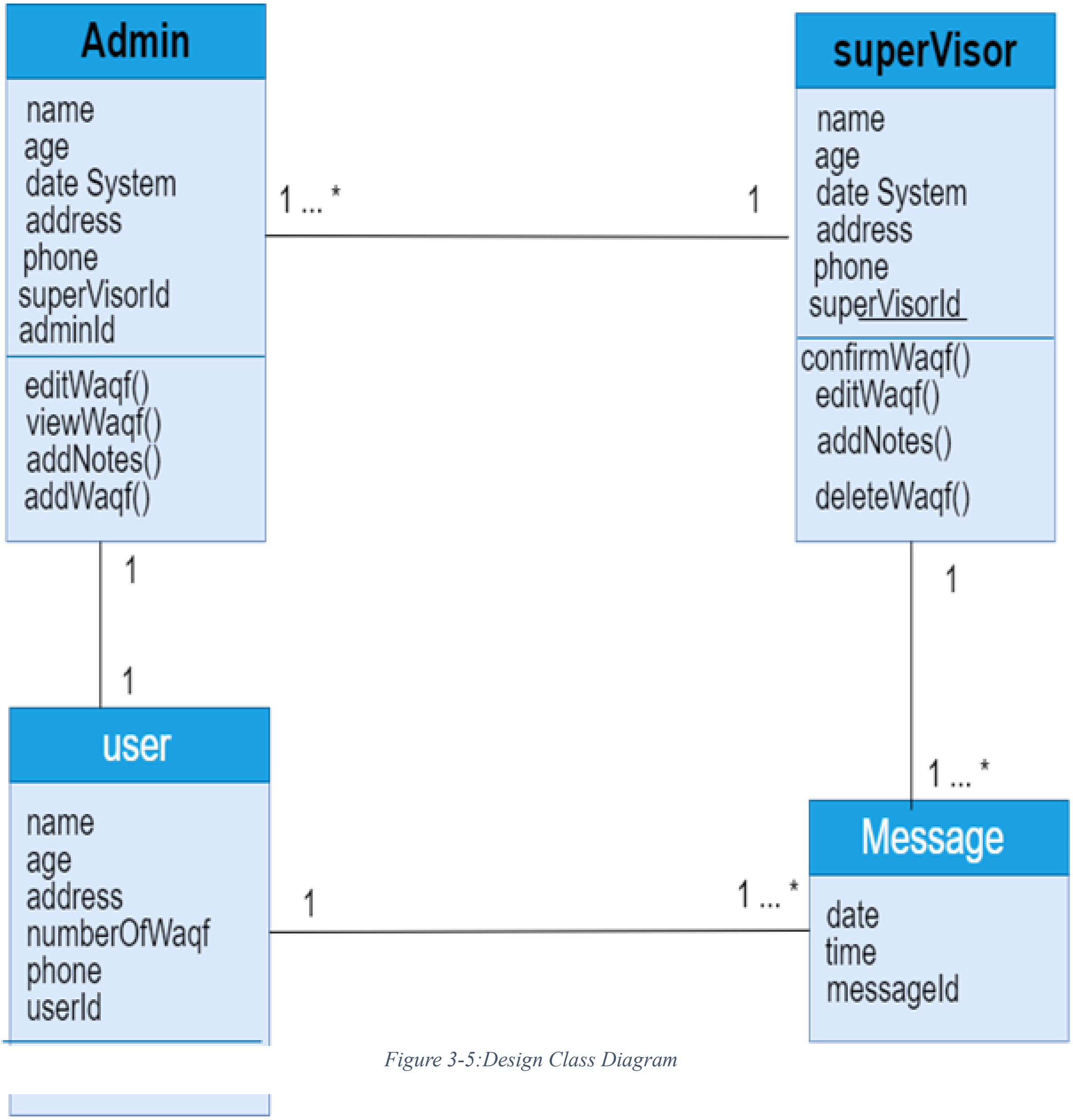


Figure 3-5:Design Class Diagram

3.7 Deployment Diagram

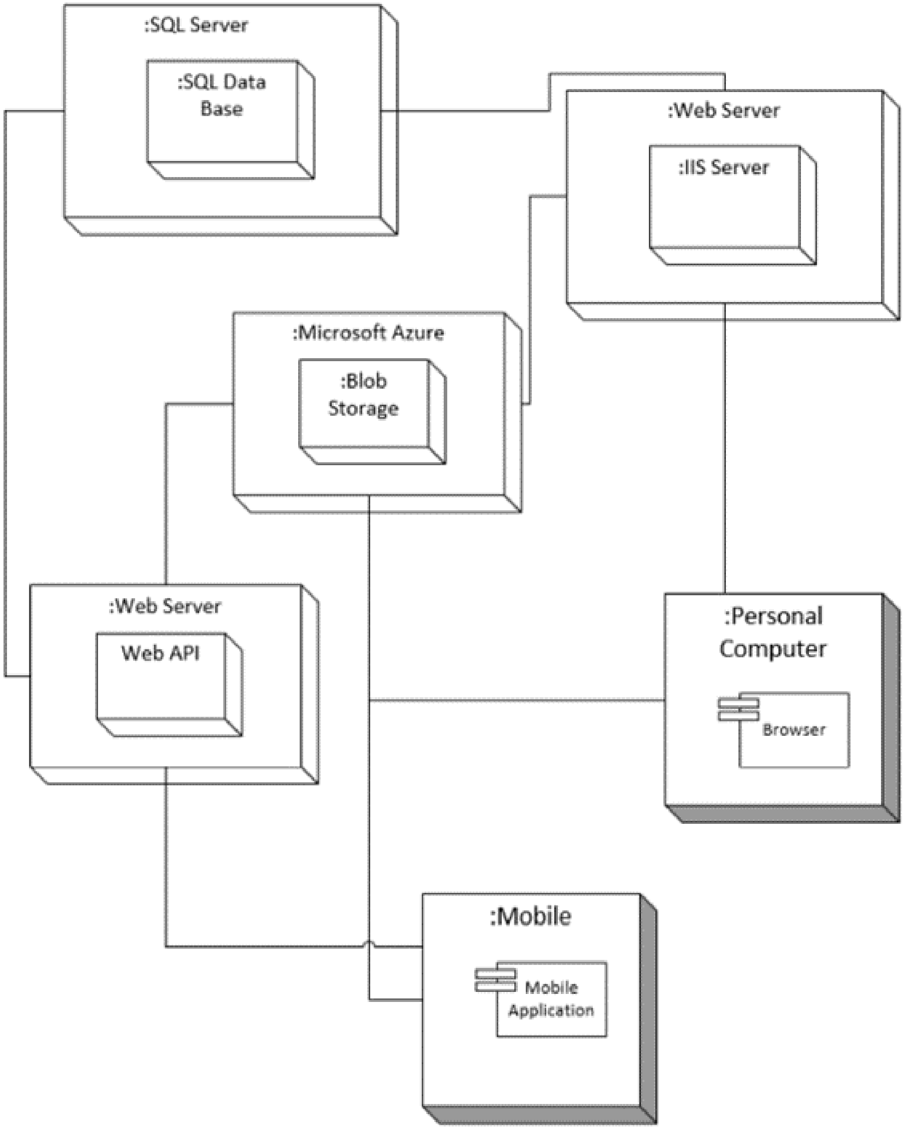


Figure 3-6:Deployment Diagram

3.8 Output & Input Design (Screens)

The screen below shows adding Waqf Page

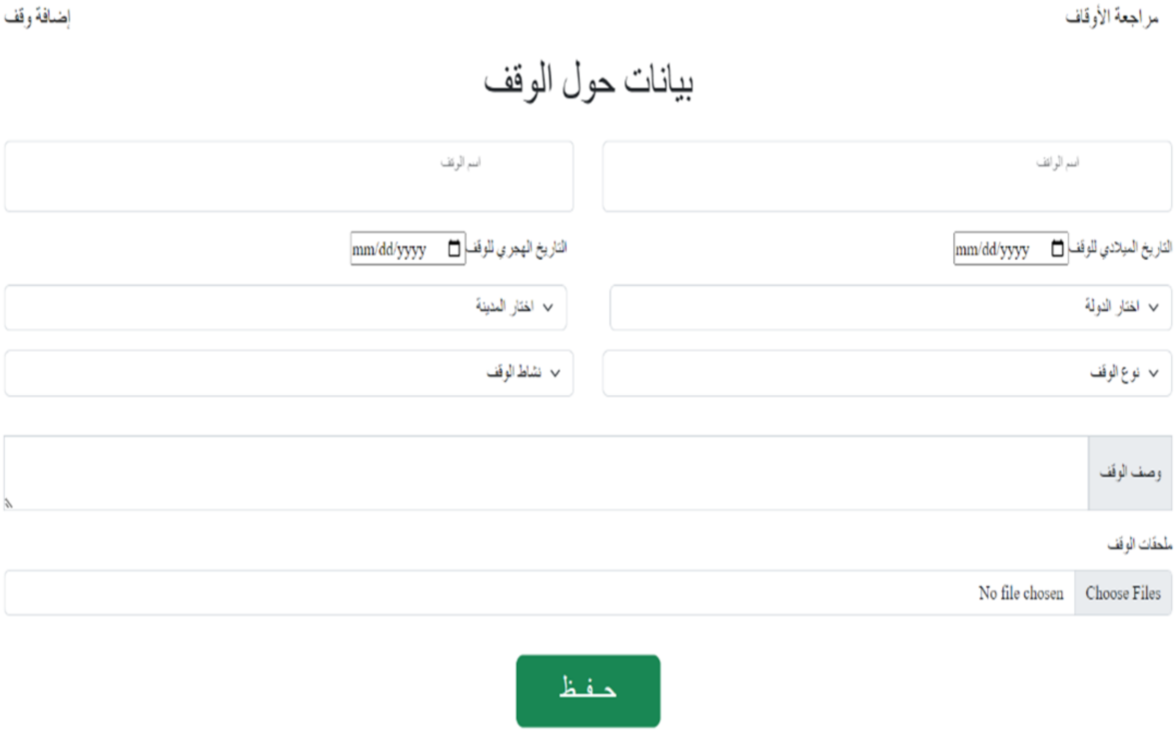


Figure 3-7:Adding Waqf

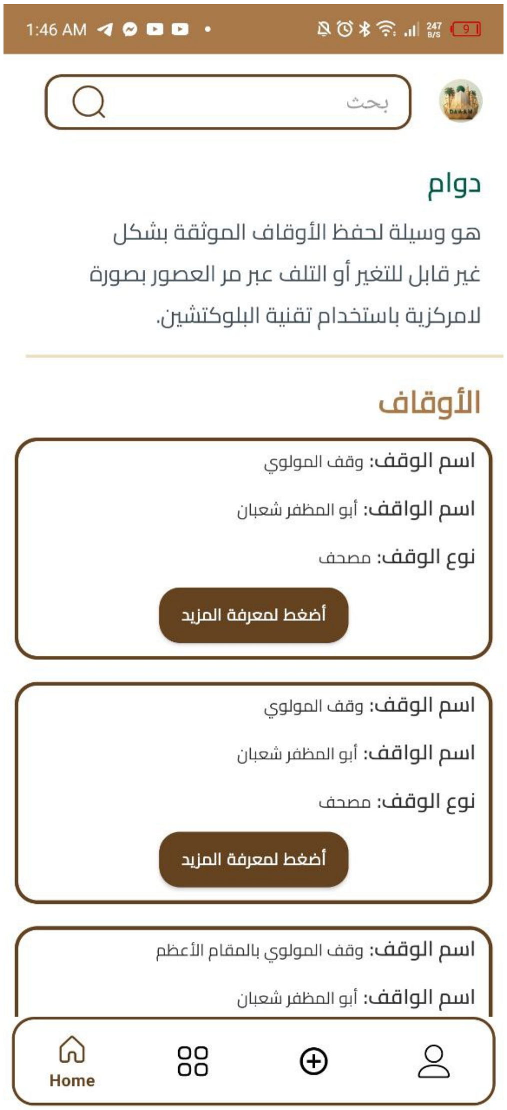
CHAPTER 4: Implementation and Testing

Our Project is divided into three parts:

- Mobile App
- Website For Users
- Dashboard

4.1 configuration code for the Mobile App

Home screen (show all Awqaf)



دوام

هو وسيلة لحفظ الأوقاف الموثقة بشكل غير قابل للتغير أو التلف عبر مر العصور بصورة لامركزية باستخدام تقنية البلوكتشين.

الأوقاف

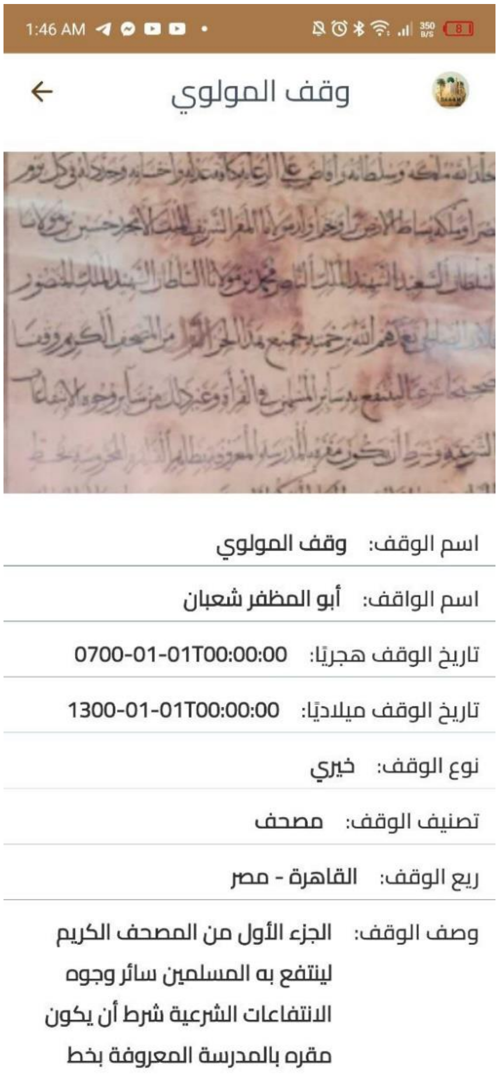
اسم الوقف: وقف المولوي
اسم الواقف: أبو المظفر شعبان
نوع الوقف: مصحف

اسم الوقف: وقف المولوي
اسم الواقف: أبو المظفر شعبان
نوع الوقف: مصحف

اسم الوقف: وقف المولوي بالمقام الأعظم
اسم الواقف: أبو المظفر شعبان

Figure 4-2:Home screen (show all Awqaf)

Waqf details (description)

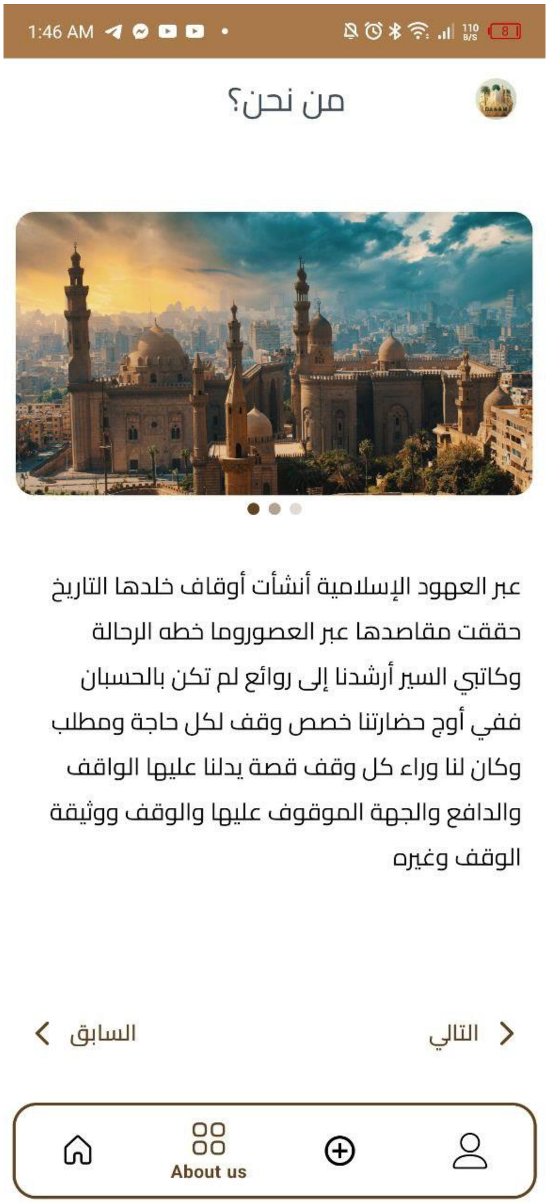


وقف المولوي

اسم الوقف: وقف المولوي
اسم الواقف: أبو المظفر شعبان
تاريخ الوقف هجريًا: 0700-01-01T00:00:00
تاريخ الوقف ميلاديًا: 1300-01-01T00:00:00
نوع الوقف: خيري
تصنيف الوقف: مصحف
ريع الوقف: القاهرة - مصر
وصف الوقف: الجزء الأول من المصحف الكريم لينتفع به المسلمين سائر وجوه الانتفاعات الشرعية شرط أن يكون مقره بالمدرسة المعروفة بخط

Figure 4-1:Waqf details (description)

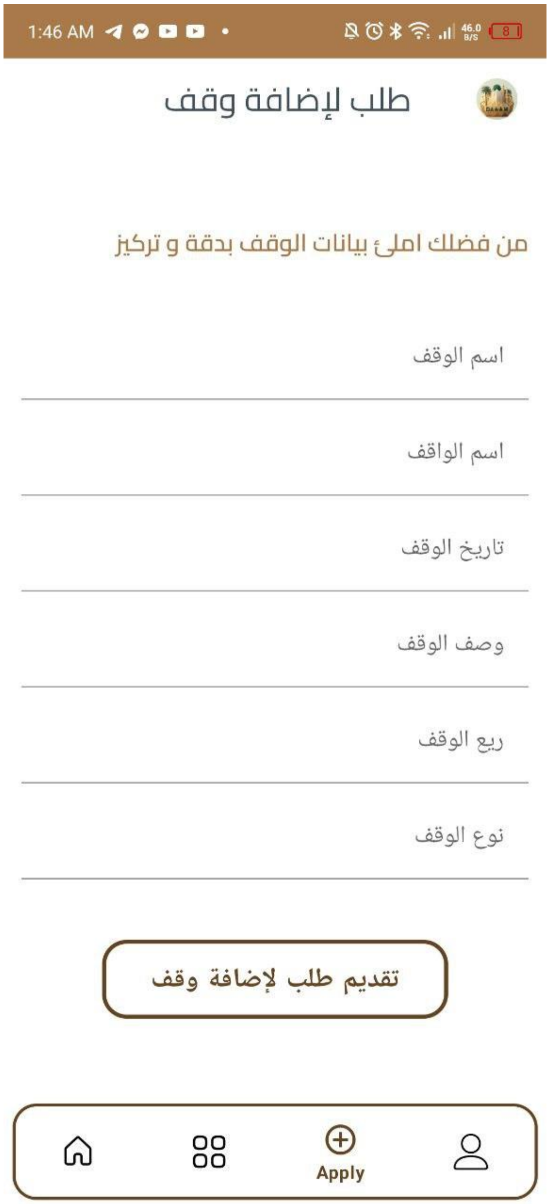
About us screens



عبر العهود الإسلامية أنشأت أوقاف خلدها التاريخ
حققت مقاصدها عبر العصوروما خطه الرحالة
وكاتبي السير أرشدنا إلى روائع لم تكن بالحسبان
ففي أوج حضارتنا خصص وقف لكل حاجة ومطلب
وكان لنا وراء كل وقف قصة يدلنا عليها الواقف
والدافع والجهة الموقوف عليها والوقف ووثيقة
الوقف وغيره

Figure 4-4:About us screens

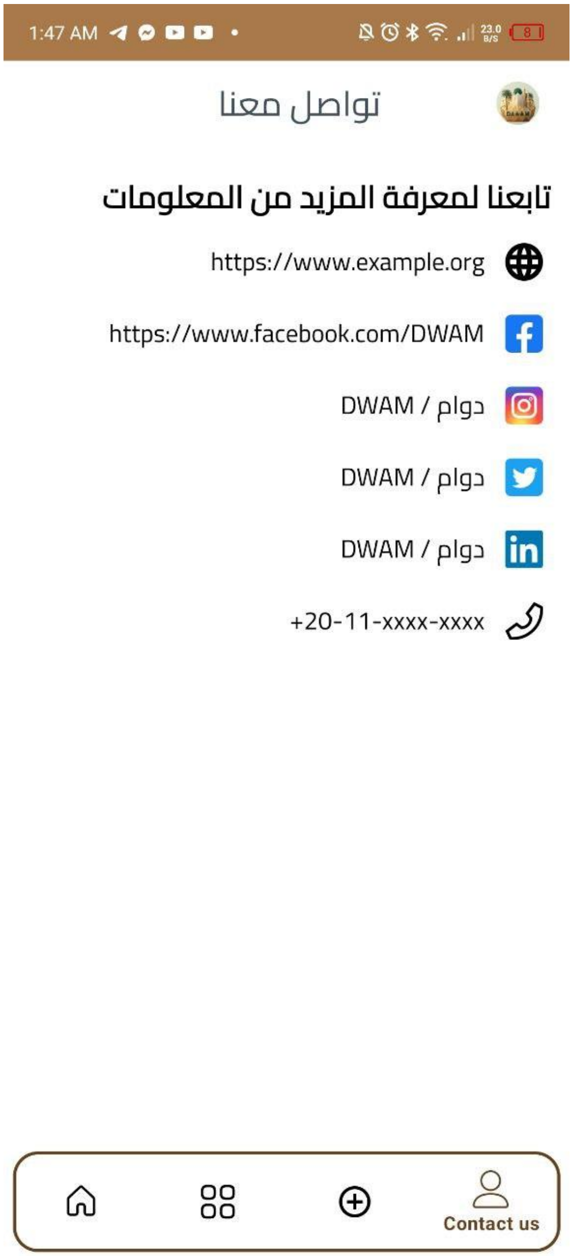
Add Waqf request



من فضلك املئ بيانات الوقف بدقة و تركيز

Figure 4-3:Add Waqf request

Contact us.



تابعنا لمعرفة المزيد من المعلومات

https://www.example.org

https://www.facebook.com/DWAM

DWAM / دوام

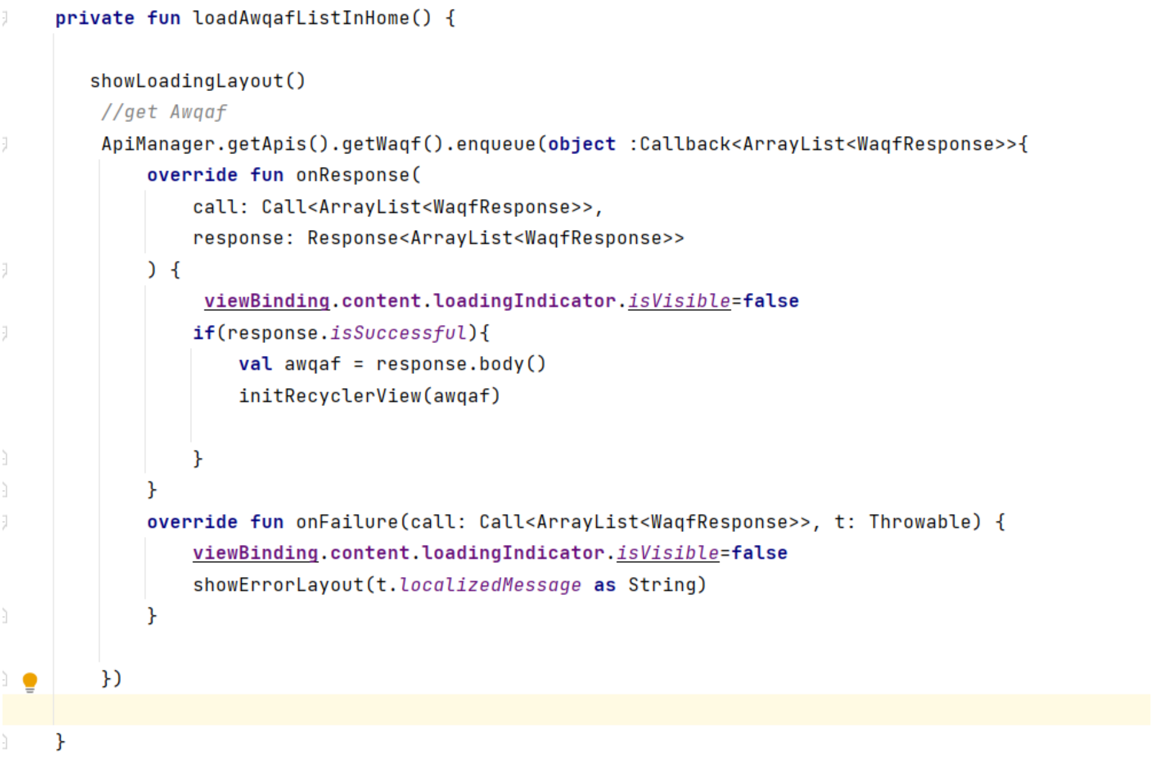
DWAM / دوام

DWAM / دوام

+20-11-xxxx-xxxx

Figure 4-5:Contact us

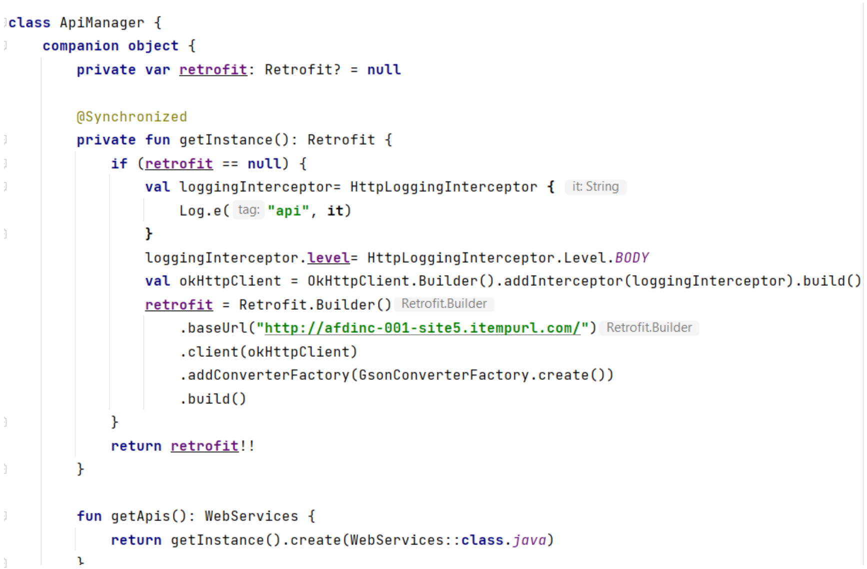
Implement API by retrofit library.



```
private fun loadAwqafListInHome() {

    showLoadingLayout()
     //get Awqaf
     ApiManager.getApis().getWaqf().enqueue(object :Callback<ArrayList<WaqfResponse>>{
         override fun onResponse(
             call: Call<ArrayList<WaqfResponse>>,
             response: Response<ArrayList<WaqfResponse>>
         ) {
              viewBinding.content.loadingIndicator.isVisible=false
             if(response.isSuccessful){
                 val awqaf = response.body()
                 initRecyclerView(awqaf)

             }
         }
         override fun onFailure(call: Call<ArrayList<WaqfResponse>>, t: Throwable) {
             viewBinding.content.loadingIndicator.isVisible=false
             showErrorLayout(t.localizedMessage as String)
         }

     })

}
```

Figure 4-6:Implement API by retrofit library

API Manager class to summons base URL once.

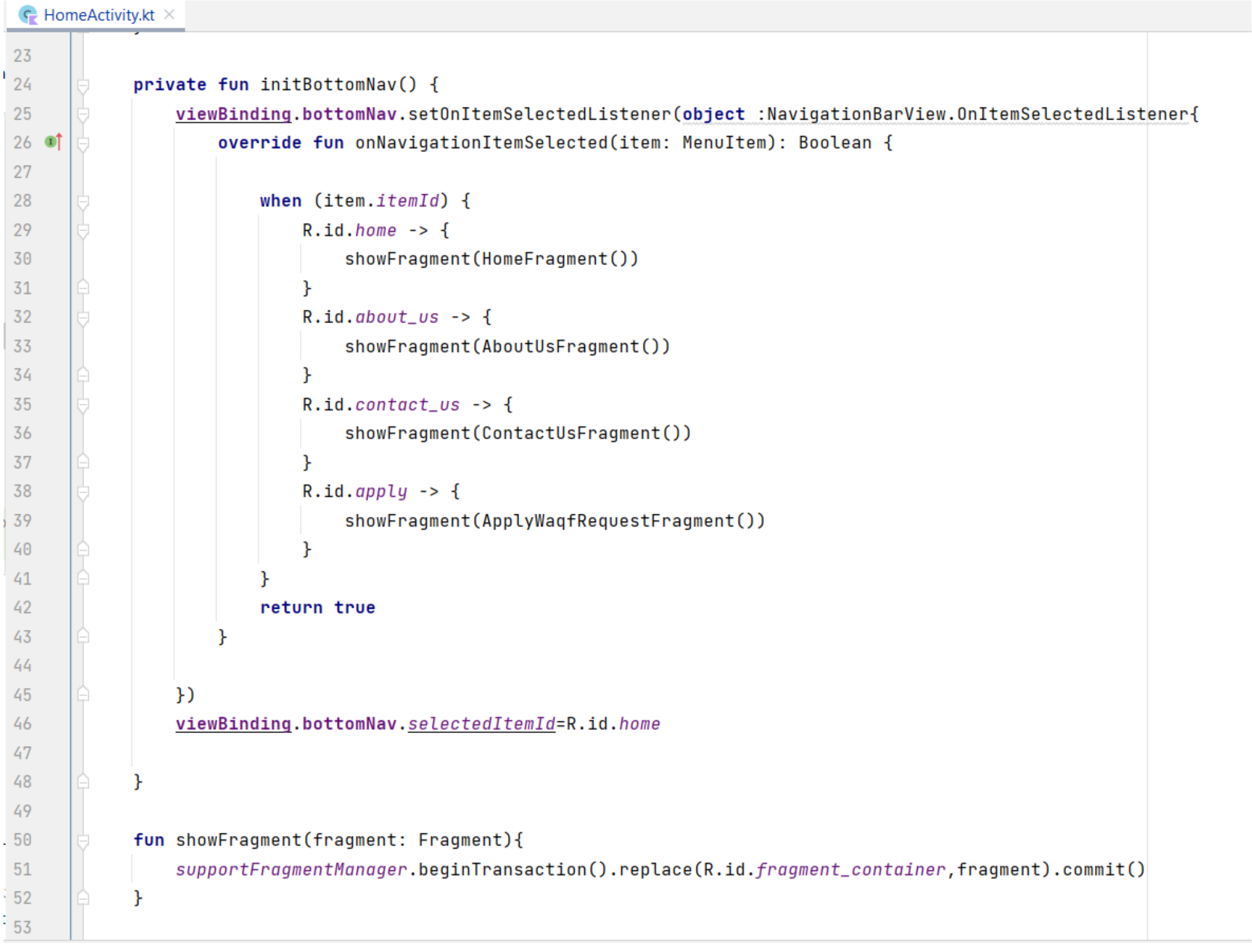


```
class ApiManager {
    companion object {
        private var retrofit: Retrofit? = null

        @Synchronized
        private fun getInstance(): Retrofit {
            if (retrofit == null) {
                val loggingInterceptor= HttpLoggingInterceptor { it: String
                    Log.e( tag: "api", it)
                }
                loggingInterceptor.level= HttpLoggingInterceptor.Level.BODY
                val okHttpClient = OkHttpClient.Builder().addInterceptor(loggingInterceptor).build()
                retrofit = Retrofit.Builder() Retrofit.Builder
                    .baseUrl("http://afdinc-001-site5.itempurl.com/") Retrofit.Builder
                    .client(okHttpClient)
                    .addConverterFactory(GsonConverterFactory.create())
                    .build()
            }
            return retrofit!!
        }

        fun getApis(): WebServices {
            return getInstance().create(WebServices::class.java)
        }
```

Figure 4-7:API Manager class to summons base URL once.

The code uses view binding to inflate layout.

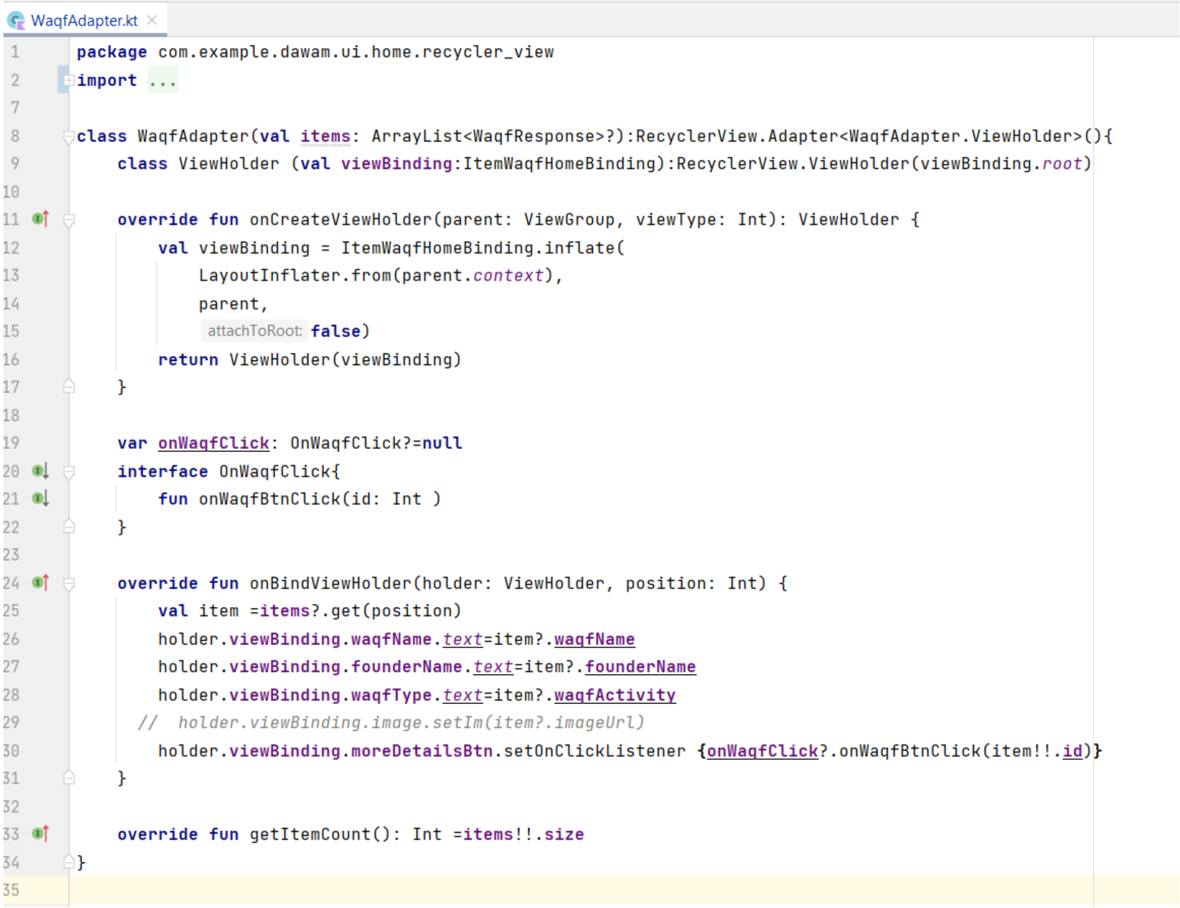
Implement bottom navigation to navigate all screens.



```kotlin
    private fun initBottomNav() {
        viewBinding.bottomNav.setOnItemSelectedListener(object :NavigationBarView.OnItemSelectedListener{
            override fun onNavigationItemSelected(item: MenuItem): Boolean {

                when (item.itemId) {
                    R.id.home -> {
                        showFragment(HomeFragment())
                    }
                    R.id.about_us -> {
                        showFragment(AboutUsFragment())
                    }
                    R.id.contact_us -> {
                        showFragment(ContactUsFragment())
                    }
                    R.id.apply -> {
                        showFragment(ApplyWaqfRequestFragment())
                    }
                }
                return true
            }

        })
        viewBinding.bottomNav.selectedItemId=R.id.home

    }

    fun showFragment(fragment: Fragment){
        supportFragmentManager.beginTransaction().replace(R.id.fragment_container,fragment).commit()
    }
```

Figure 4-8:Implement bottom navigation to navigate all screens

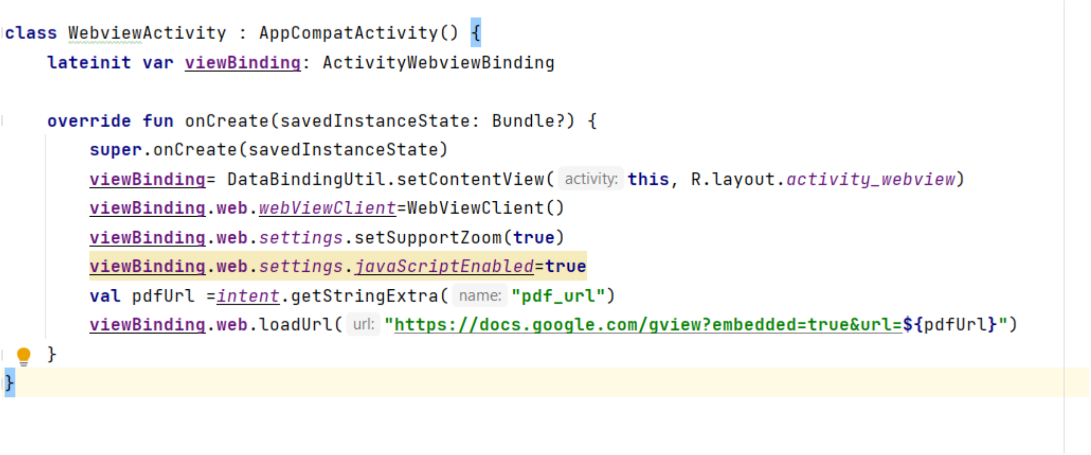
Recycler view to show all Awqaf list in Home page.



```
package com.example.dawam.ui.home.recycler_view
import ...

class WaqfAdapter(val items: ArrayList<WaqfResponse>?):RecyclerView.Adapter<WaqfAdapter.ViewHolder>(){
    class ViewHolder (val viewBinding:ItemWaqfHomeBinding):RecyclerView.ViewHolder(viewBinding.root)

    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): ViewHolder {
        val viewBinding = ItemWaqfHomeBinding.inflate(
            LayoutInflater.from(parent.context),
            parent,
             attachToRoot: false)
        return ViewHolder(viewBinding)
    }

    var onWaqfClick: OnWaqfClick?=null
    interface OnWaqfClick{
        fun onWaqfBtnClick(id: Int )
    }

    override fun onBindViewHolder(holder: ViewHolder, position: Int) {
        val item =items?.get(position)
        holder.viewBinding.waqfName.text=item?.waqfName
        holder.viewBinding.founderName.text=item?.founderName
        holder.viewBinding.waqfType.text=item?.waqfActivity
      //  holder.viewBinding.image.setIm(item?.imageUrl)
        holder.viewBinding.moreDetailsBtn.setOnClickListener {onWaqfClick?.onWaqfBtnClick(item!!.id)}
    }

    override fun getItemCount(): Int =items!!.size
}
```

Figure 4-9:Recycler view to show all Awqaf list in Home page

WebView class to show the document of Waqf



```
class WebviewActivity : AppCompatActivity() {
    lateinit var viewBinding: ActivityWebviewBinding

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        viewBinding= DataBindingUtil.setContentView( activity: this, R.layout.activity_webview)
        viewBinding.web.webViewClient=WebViewClient()
        viewBinding.web.settings.setSupportZoom(true)
        viewBinding.web.settings.javaScriptEnabled=true
        val pdfUrl =intent.getStringExtra( name: "pdf_url")
        viewBinding.web.loadUrl( url: "https://docs.google.com/gview?embedded=true&url=${pdfUrl}")
    }
}
```

Figure 4-10:WebView class to show the document of Waqf

Front-End Part

Our Project is divided into two separate websites one for All users to look at past waqfs and search about it.

Technologies used at front-end:

HTML

CSS

JS

React JS

We have used react JS framework to make website working with Single Page Application (SPA) to prevent Loading.

Users Website

Through this website users can reach out to Waqfs and Search about it

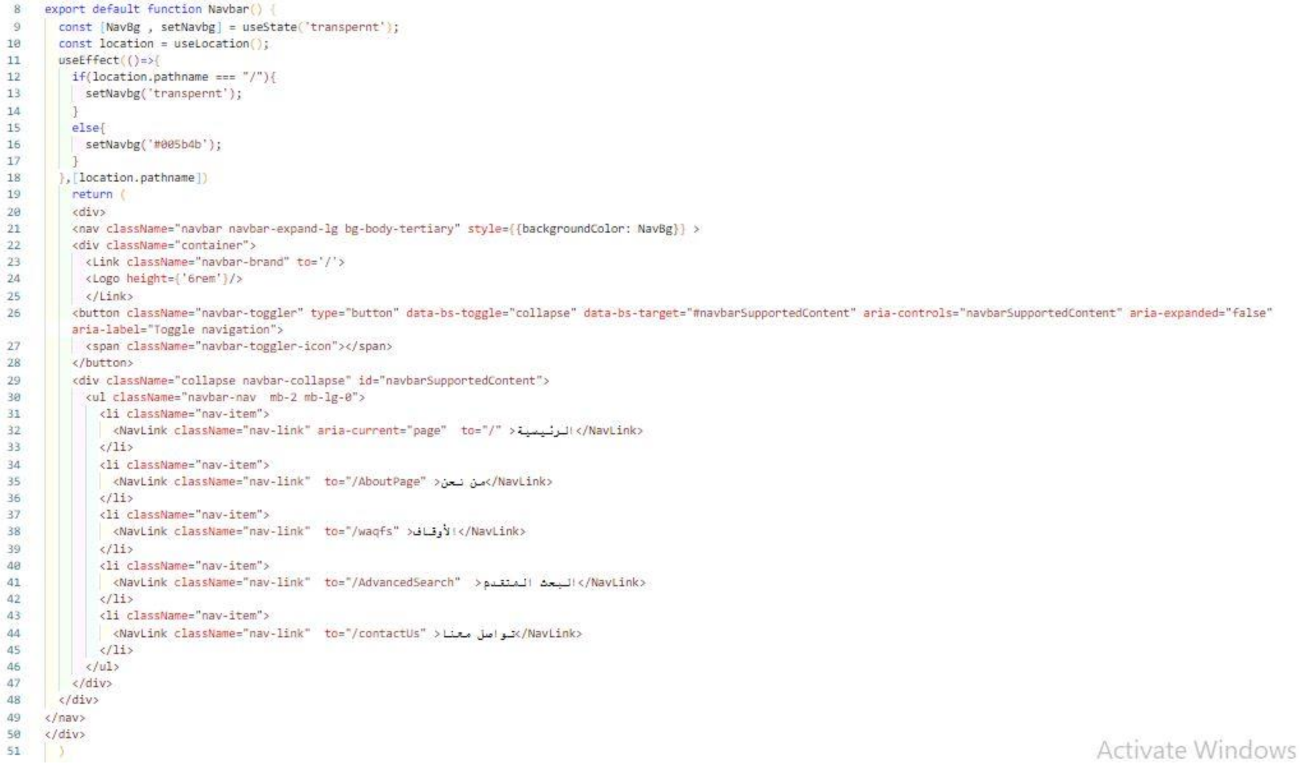
Users website consists of pages (Home-About-Awqaf-Search-Contac

4.2 configuration code for the website (Front-end):

4.2.1 Users Website

The SW shall illustrate: the navbar to show different options of pages that site include, and header show the name and description of website.

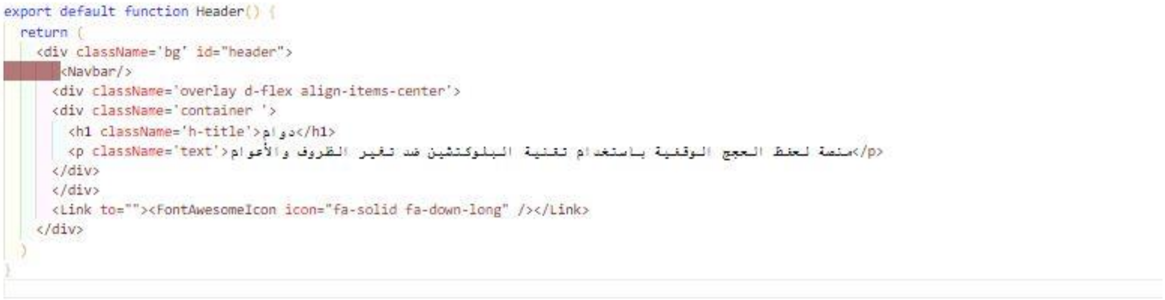
Navbar Code:



```
export default function Navbar() {
  const [NavBg , setNavbg] = useState('transpernt');
  const location = useLocation();
  useEffect(()=>{
    if(location.pathname === "/"){
      setNavbg('transpernt');
    }
    else{
      setNavbg('#005b4b');
    }
  },[location.pathname])
    return (
    <div>
    <nav className="navbar navbar-expand-lg bg-body-tertiary" style={{backgroundColor: NavBg}} >
    <div className="container">
      <Link className="navbar-brand" to='/'>
      <Logo height={'6rem'}/>
      </Link>
    <button className="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false"
    aria-label="Toggle navigation">
      <span className="navbar-toggler-icon"></span>
    </button>
    <div className="collapse navbar-collapse" id="navbarSupportedContent">
      <ul className="navbar-nav  mb-2 mb-lg-0">
        <li className="nav-item">
          <NavLink className="nav-link" aria-current="page"  to="/" >الرئيسية</NavLink>
        </li>
        <li className="nav-item">
          <NavLink className="nav-link"  to="/AboutPage" >من نحن</NavLink>
        </li>
        <li className="nav-item">
          <NavLink className="nav-link"  to="/waqfs" >الأوقاف</NavLink>
        </li>
        <li className="nav-item">
          <NavLink className="nav-link"  to="/AdvancedSearch"  >البحث المتقدم</NavLink>
        </li>
        <li className="nav-item">
          <NavLink className="nav-link"  to="/contactUs" >تواصل معنا</NavLink>
        </li>
      </ul>
    </div>
  </div>
</nav>
</div>
  )
```

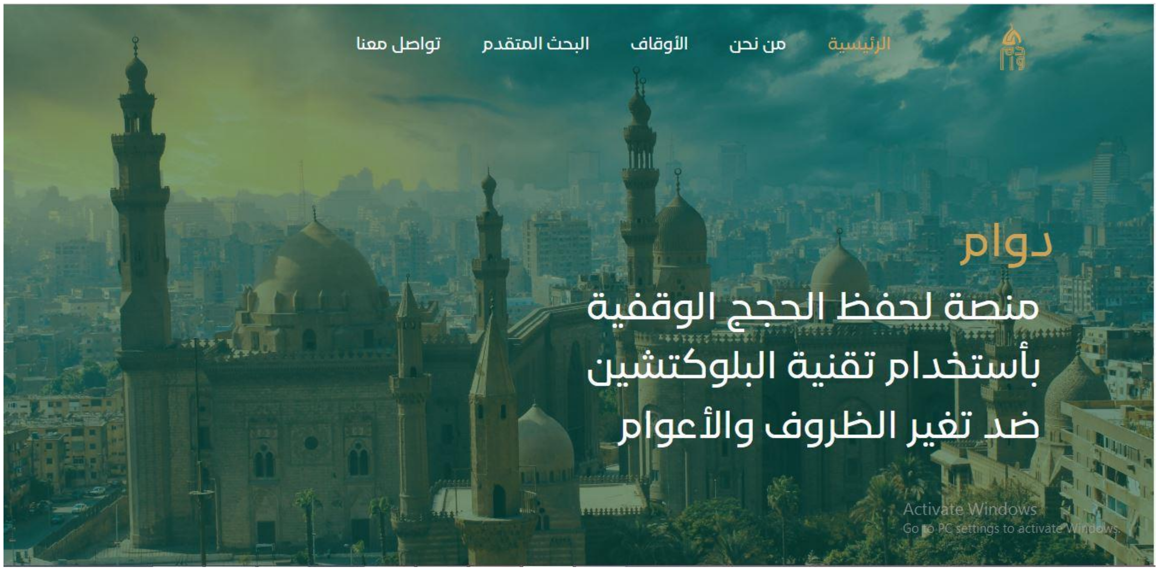
Figure 4-11:Navbar Code

Header:



```
export default function Header() {
  return (
    <div className='bg' id="header">
      <Navbar/>
      <div className='overlay d-flex align-items-center'>
      <div className='container '>
        <h1 className='h-title'>دوام</h1>
        <p className='text'>منصة لحفظ الحجج الوقفية باستخدام تقنية البلوكتشين ضد تغير الظروف والأعوام</p>
      </div>
      </div>
      <Link to=""><FontAwesomeIcon icon="fa-solid fa-down-long" /></Link>
    </div>
  )
}
```

4-12:Header Code

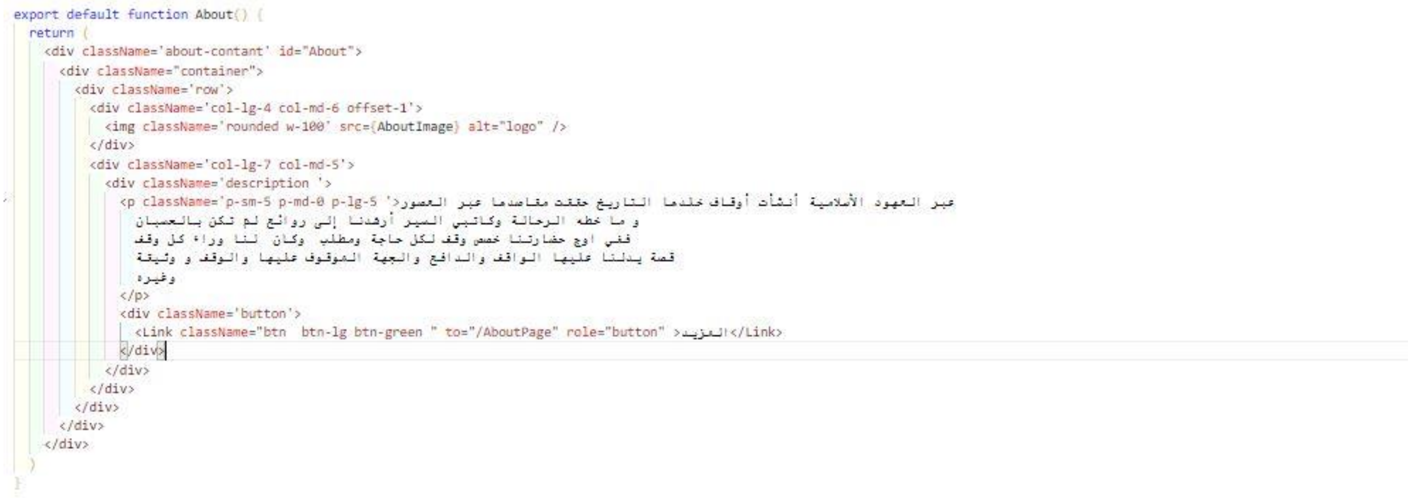


دوام

منصة لحفظ الحجج الوقفية
بأستخدام تقنية البلوكتشين
ضد تغير الظروف والأعوام

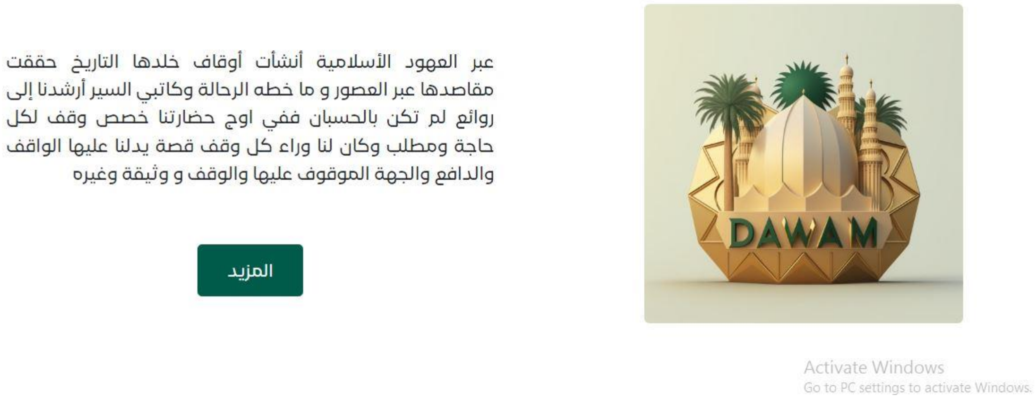
Figure 4-13:showing navbar and header in website

The SW shall illustrate: about section which contains an overview of the function performed by the site and some concepts of endowments and Blockchain, it include button to show more details.



```
export default function About() {
  return (
    <div className='about-contant' id="About">
      <div className="container">
        <div className='row'>
          <div className='col-lg-4 col-md-6 offset-1'>
            <img className='rounded w-100' src={AboutImage} alt="logo" />
          </div>
          <div className='col-lg-7 col-md-5'>
            <div className='description '>
              <p className='p-sm-5 p-md-0 p-lg-5 '>عبر العهود الأسلامية أنشأت أوقاف خلدها التاريخ حققت مقاصدها عبر العصور
                و ما خطه الرحالة وكاتبي السير أرشدنا إلى روائع لم تكن بالحسبان
                ففي اوج حضارتنا خصص وقف لكل حاجة ومطلب  وكان  لنا وراء كل وقف
                قصة يدلنا عليها الواقف والدافع والجهة الموقوف عليها والوقف و وثيقة
                وغيره
              </p>
              <div className='button'>
                <Link className="btn  btn-lg btn-green " to="/AboutPage" role="button" >المزيد</Link>
              </div>
            </div>
          </div>
        </div>
      </div>
    </div>
  )
}
```

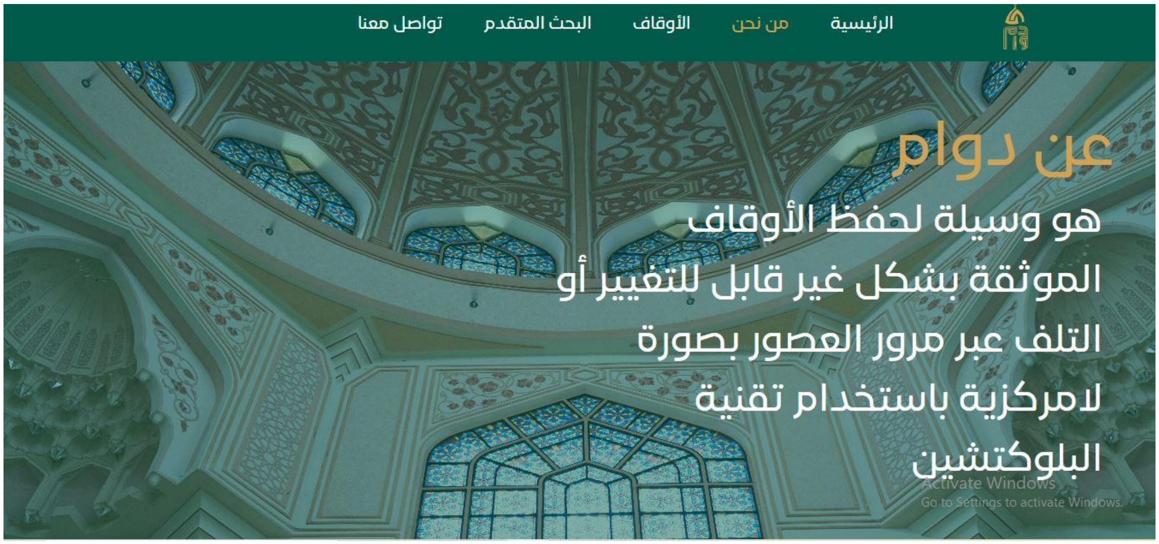
Figure 4-14:about code



عبر العهود الأسلامية أنشأت أوقاف خلدها التاريخ حققت
مقاصدها عبر العصور و ما خطه الرحالة وكاتبي السير أرشدنا إلى
روائع لم تكن بالحسبان ففي اوج حضارتنا خصص وقف لكل
حاجة ومطلب وكان لنا وراء كل وقف قصة يدلنا عليها الواقف
والدافع والجهة الموقوف عليها والوقف و وثيقة وغيره

Figure 4-15:showing about section in home page

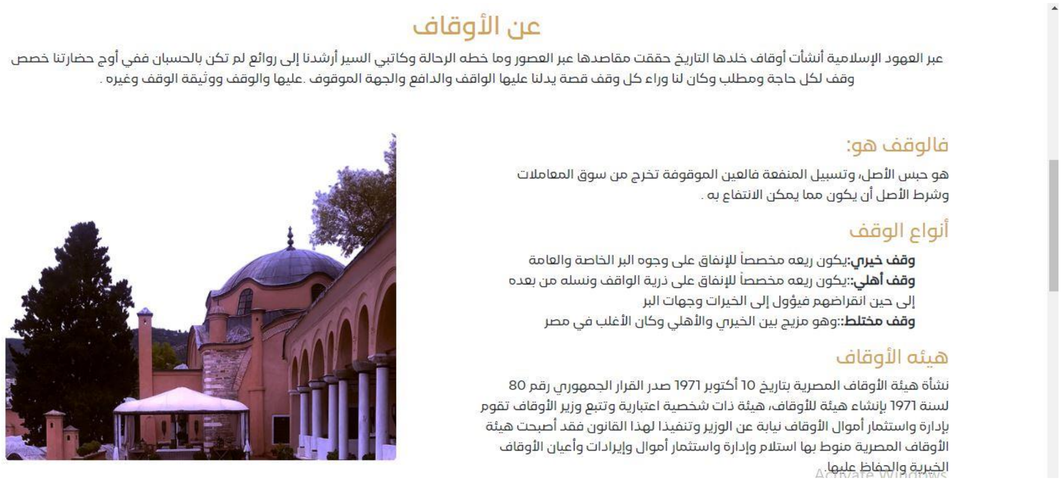
More button for switching to about page to show more information about platform DAWAM.



عن دوام

هو وسيلة لحفظ الأوقاف
الموثقة بشكل غير قابل للتغيير أو
التلف عبر مرور العصور بصورة
لامركزية باستخدام تقنية
البلوكتشين

Figure 4-16:Showing About page



عن الأوقاف

عبر العهود الإسلامية أنشأت أوقاف خلدها التاريخ حققت مقاصدها عبر العصور وما خطه الرحالة وكاتبي السير أرشدنا إلى روائع لم تكن بالحسبان ففي أوج حضارتنا خصص
وقف لكل حاجة ومطلب وكان لنا وراء كل وقف قصة يدلنا عليها الواقف والدافع والجهة الموقوف .عليها والوقف ووثيقة الوقف وغيره .

فالوقف هو:

هو حبس الأصل، وتسبيل المنفعة فالعين الموقوفة تخرج من سوق المعاملات
وشرط الأصل أن يكون مما يمكن الانتفاع به .

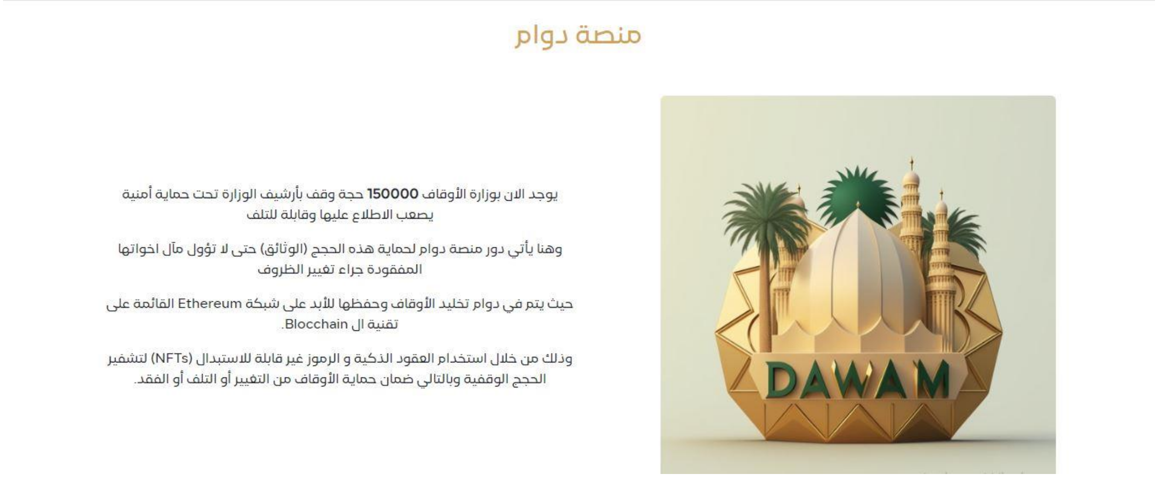
أنواع الوقف

وقف خيري:يكون ريعه مخصصاً للإنفاق على وجوه البر الخاصة والعامة
وقف أهلي::يكون ريعه مخصصاً للإنفاق على ذرية الواقف ونسله من بعده
إلى حين انقراضهم فيؤول إلى الخيرات وجهات البر
وقف مختلط::وهو مزيج بين الخيري والأهلي وكان الأغلب في مصر

هيئه الأوقاف

نشأة هيئة الأوقاف المصرية بتاريخ 10 أكتوبر 1971 صدر القرار الجمهوري رقم 80
لسنة 1971 بإنشاء هيئة للأوقاف، هيئة ذات شخصية اعتبارية وتتبع وزير الأوقاف تقوم
بإدارة واستثمار أموال الأوقاف نيابة عن الوزير وتنفيذا لهذا القانون فقد أصبحت هيئة
الأوقاف المصرية منوط بها استلام وإدارة واستثمار أموال وإيرادات وأعيان الأوقاف
الخيرية والحفاظ عليها.

Figure 4-18:About Awqaf



منصة دوام

يوجد الان بوزارة الأوقاف **150000** حجة وقف بأرشيف الوزارة تحت حماية أمنية
يصعب الاطلاع عليها وقابلة للتلف

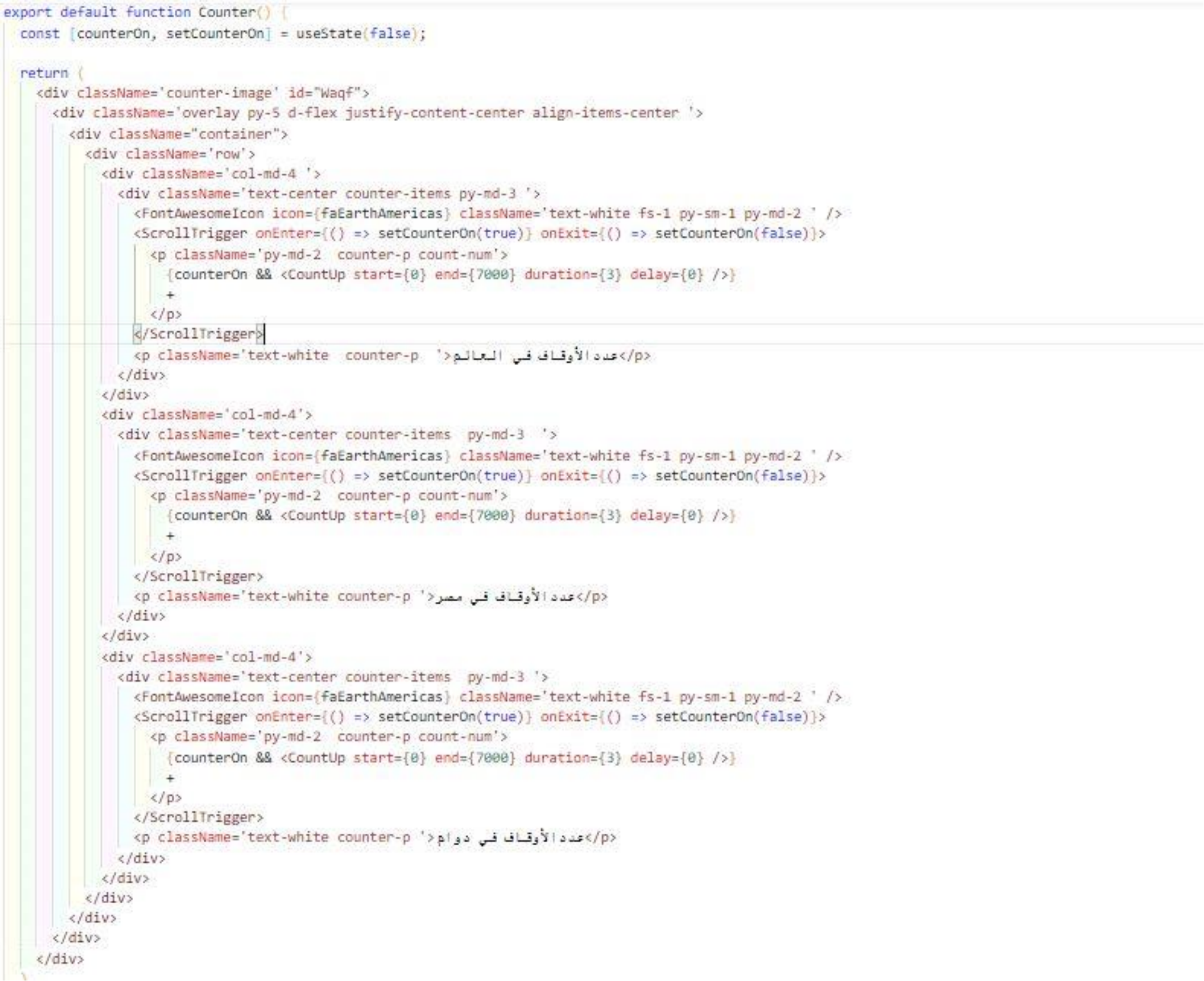
وهنا يأتي دور منصة دوام لحماية هذه الحجج (الوثائق) حتى لا تؤول مآل اخواتها
المفقودة جراء تغيير الظروف

حيث يتم في دوام تخليد الأوقاف وحفظها للأبد على شبكة Ethereum القائمة على
تقنية ال Blocchain.

وذلك من خلال استخدام العقود الذكية و الرموز غير قابلة للاستبدال (NFTs) لتشفير
الحجج الوقفية وبالتالي ضمان حماية الأوقاف من التغيير أو التلف أو الفقد.

Figure 4-17:About Dawam

The SW shall illustrate: counter section which count number of awqfs in world, Egypt and DAWAM platform.



```
export default function Counter() {
  const [counterOn, setCounterOn] = useState(false);

  return (
    <div className='counter-image' id="Waqf">
      <div className='overlay py-5 d-flex justify-content-center align-items-center '>
        <div className="container">
          <div className='row'>
            <div className='col-md-4 '>
              <div className='text-center counter-items py-md-3 '>
                <FontAwesomeIcon icon={faEarthAmericas} className='text-white fs-1 py-sm-1 py-md-2 ' />
                <ScrollTrigger onEnter={() => setCounterOn(true)} onExit={() => setCounterOn(false)}>
                  <p className='py-md-2  counter-p count-num'>
                    {counterOn && <CountUp start={0} end={7000} duration={3} delay={0} />}
                    +
                  </p>
                </ScrollTrigger>
                <p className='text-white  counter-p  '>عددالأوقاف في العالم</p>
              </div>
            </div>
            <div className='col-md-4'>
              <div className='text-center counter-items  py-md-3  '>
                <FontAwesomeIcon icon={faEarthAmericas} className='text-white fs-1 py-sm-1 py-md-2 ' />
                <ScrollTrigger onEnter={() => setCounterOn(true)} onExit={() => setCounterOn(false)}>
                  <p className='py-md-2  counter-p count-num'>
                    {counterOn && <CountUp start={0} end={7000} duration={3} delay={0} />}
                    +
                  </p>
                </ScrollTrigger>
                <p className='text-white counter-p '>عددالأوقاف في مصر</p>
              </div>
            </div>
            <div className='col-md-4'>
              <div className='text-center counter-items  py-md-3 '>
                <FontAwesomeIcon icon={faEarthAmericas} className='text-white fs-1 py-sm-1 py-md-2 ' />
                <ScrollTrigger onEnter={() => setCounterOn(true)} onExit={() => setCounterOn(false)}>
                  <p className='py-md-2  counter-p count-num'>
                    {counterOn && <CountUp start={0} end={7000} duration={3} delay={0} />}
                    +
                  </p>
                </ScrollTrigger>
                <p className='text-white counter-p '>عددالأوقاف في دوام</p>
              </div>
            </div>
          </div>
        </div>
      </div>
    </div>
  )
```

Figure 4-20:showing counter code

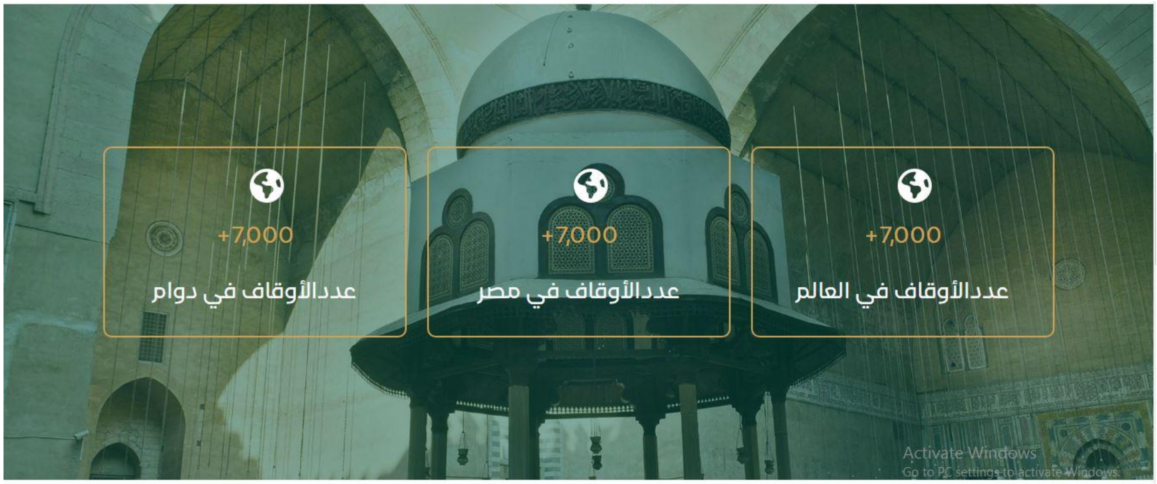
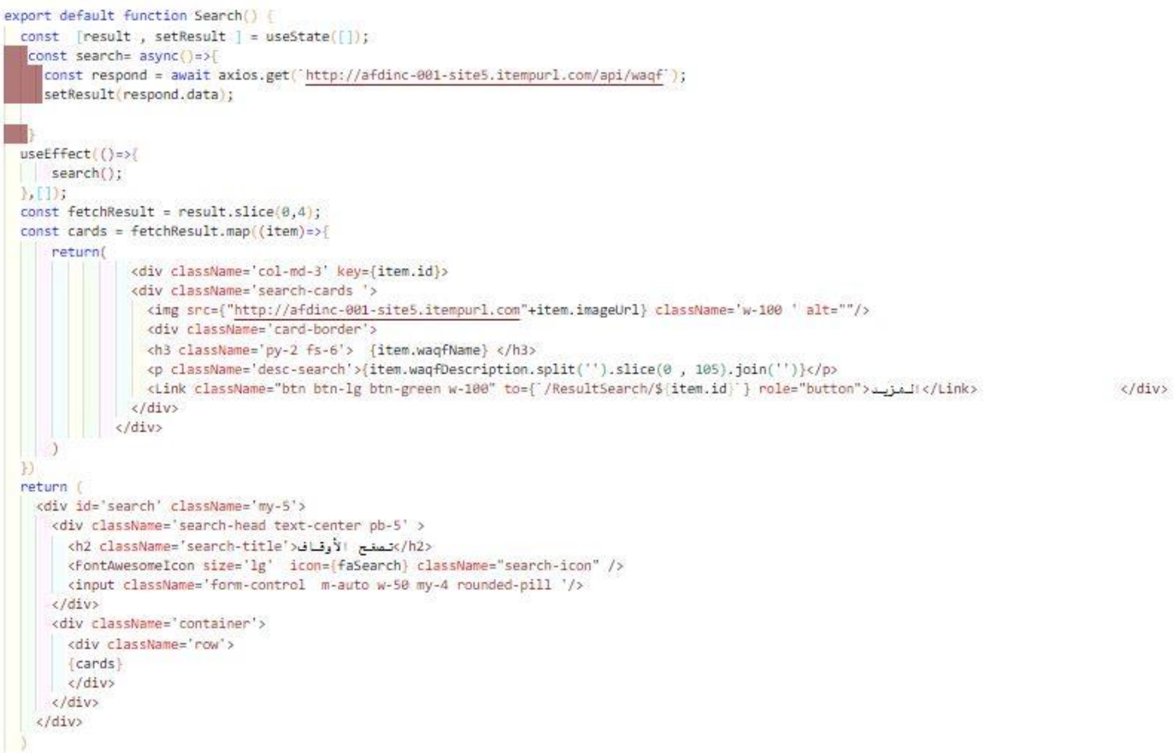


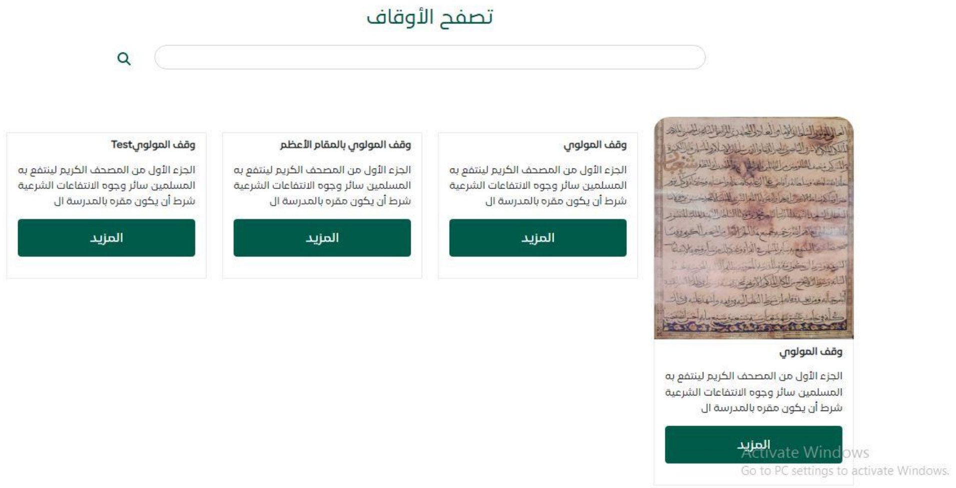
Figure 4-19:Showing Counter Section

The SW shall illustrate: waqfs search section which include input search to do advanced search of waqfs in DAWAM.



```
export default function Search() {
  const  [result , setResult ] = useState([]);
  const search= async()=>{
    const respond = await axios.get(`http://afdinc-001-site5.itempurl.com/api/waqf`);
    setResult(respond.data);

  }
  useEffect(()=>{
      search();
  },[]);
  const fetchResult = result.slice(0,4);
  const cards = fetchResult.map((item)=>{
      return(
                <div className='col-md-3' key={item.id}>
                <div className='search-cards '>
                  <img src={"http://afdinc-001-site5.itempurl.com"+item.imageUrl} className='w-100 ' alt=""/>
                  <div className='card-border'>
                  <h3 className='py-2 fs-6'>  {item.waqfName} </h3>
                  <p className='desc-search'>{item.waqfDescription.split('').slice(0 , 105).join('')}</p>
                  <Link className="btn btn-lg btn-green w-100" to={`/ResultSearch/${item.id}`} role="button">المزيد</Link>                  </div>
                </div>
              </div>
      )
  })
  return (
    <div id='search' className='my-5'>
      <div className='search-head text-center pb-5' >
        <h2 className='search-title'>تصفح الأوقاف</h2>
        <FontAwesomeIcon size='lg'  icon={faSearch} className="search-icon" />
        <input className='form-control  m-auto w-50 my-4 rounded-pill '/>
      </div>
      <div className='container'>
        <div className='row'>
        {cards}
        </div>
      </div>
    </div>
  )
```

Figure 4-21:showing waqfs search section code



تصفح الأوقاف

وقف المولوي

الجزء الأول من المصحف الكريم لينتفع به المسلمين سائر وجوه الانتفاعات الشرعية شرط أن يكون مقره بالمدرسة ال

المزيد

وقف المولوي

الجزء الأول من المصحف الكريم لينتفع به المسلمين سائر وجوه الانتفاعات الشرعية شرط أن يكون مقره بالمدرسة ال

المزيد

وقف المولوي بالمقام الأعظم

الجزء الأول من المصحف الكريم لينتفع به المسلمين سائر وجوه الانتفاعات الشرعية شرط أن يكون مقره بالمدرسة ال

المزيد

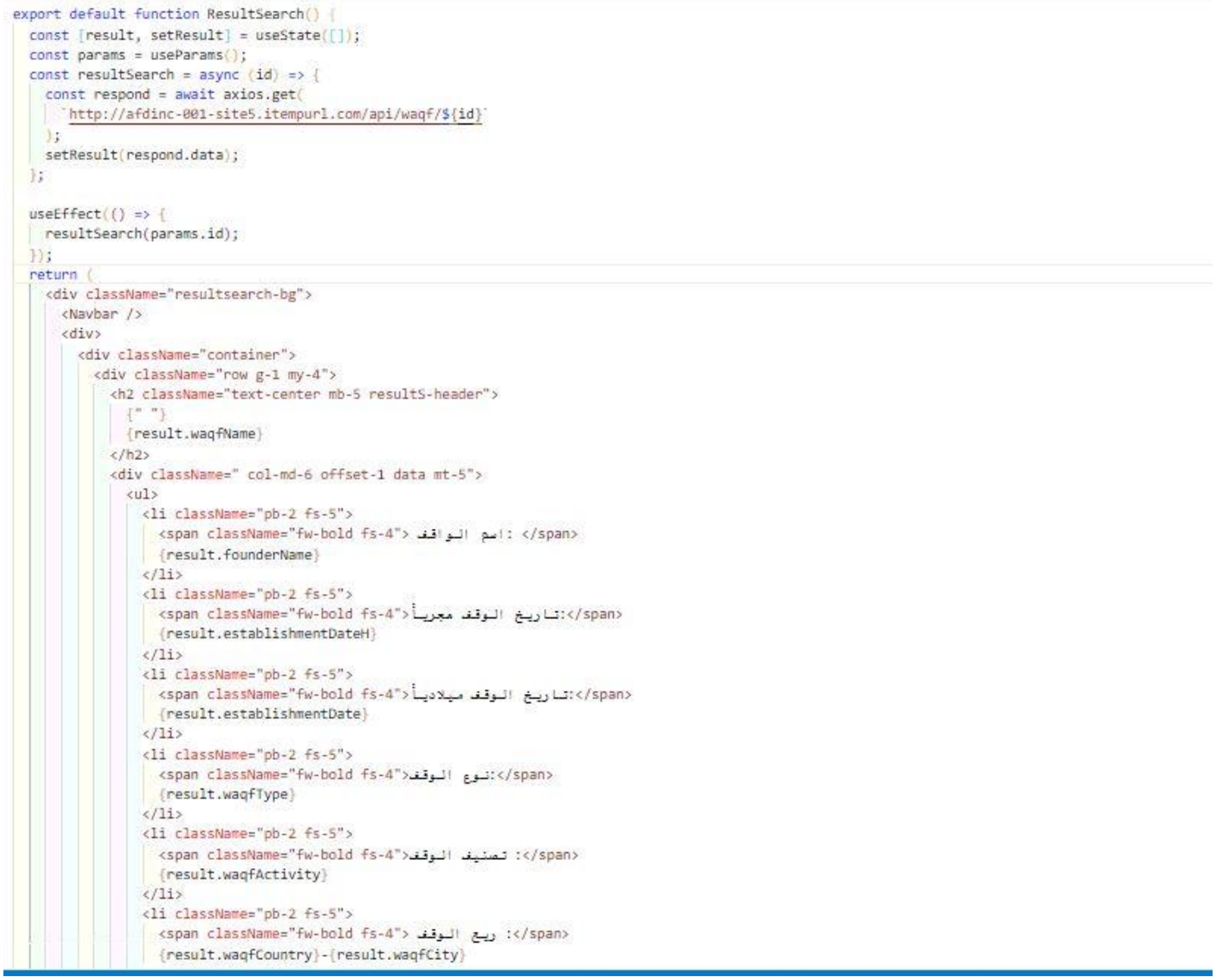
وقف المولويTest

الجزء الأول من المصحف الكريم لينتفع به المسلمين سائر وجوه الانتفاعات الشرعية شرط أن يكون مقره بالمدرسة ال

المزيد

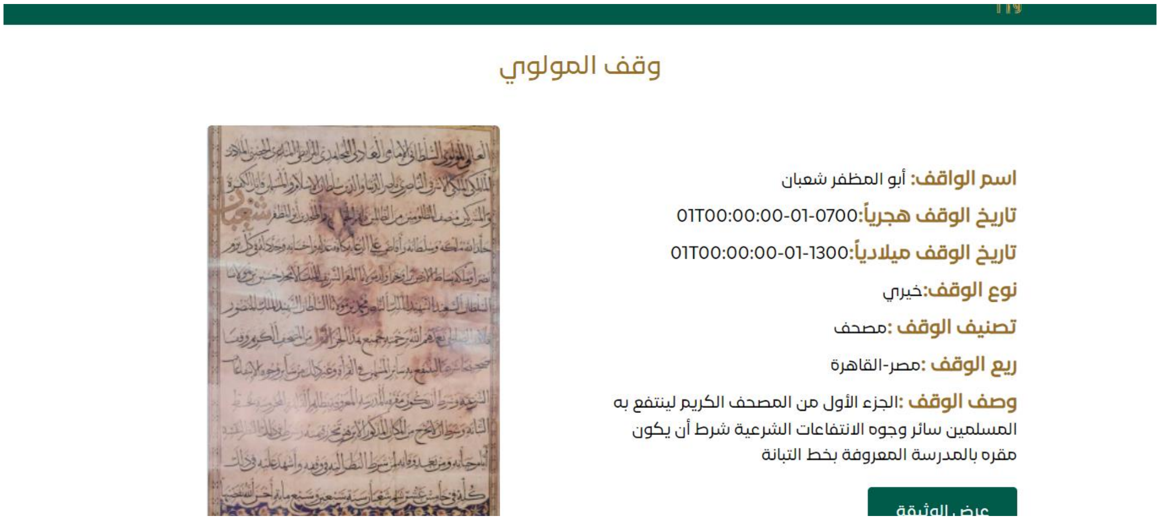
Figure 4-22:showing waqfs search section in home page

Cards include of more button for switching to show more details about waqf



```
export default function ResultSearch() {
  const [result, setResult] = useState([]);
  const params = useParams();
  const resultSearch = async (id) => {
    const respond = await axios.get(
      `http://afdinc-001-site5.itempurl.com/api/waqf/${id}`
    );
    setResult(respond.data);
  };

  useEffect(() => {
    resultSearch(params.id);
  });
  return (
    <div className="resultsearch-bg">
      <Navbar />
      <div>
        <div className="container">
          <div className="row g-1 my-4">
            <h2 className="text-center mb-5 resultS-header">
              {" "}
              {result.waqfName}
            </h2>
            <div className=" col-md-6 offset-1 data mt-5">
              <ul>
                <li className="pb-2 fs-5">
                  <span className="fw-bold fs-4"> اسم الواقف : </span>
                  {result.founderName}
                </li>
                <li className="pb-2 fs-5">
                  <span className="fw-bold fs-4">تاريخ الوقف هجرياً:</span>
                  {result.establishmentDateH}
                </li>
                <li className="pb-2 fs-5">
                  <span className="fw-bold fs-4">تاريخ الوقف ميلادياً:</span>
                  {result.establishmentDate}
                </li>
                <li className="pb-2 fs-5">
                  <span className="fw-bold fs-4">نوع الوقف:</span>
                  {result.waqfType}
                </li>
                <li className="pb-2 fs-5">
                  <span className="fw-bold fs-4">تصنيف الوقف :</span>
                  {result.waqfActivity}
                </li>
                <li className="pb-2 fs-5">
                  <span className="fw-bold fs-4"> ريع الوقف :</span>
                  {result.waqfCountry}-{result.waqfCity}
```

Figure 4-24:showing more details about waqf code



وقف المولوي

اسم الواقف: أبو المظفر شعبان

تاريخ الوقف هجرياً:0700-01-01T00:00:00

تاريخ الوقف ميلادياً:1300-01-01T00:00:00

نوع الوقف:خيري

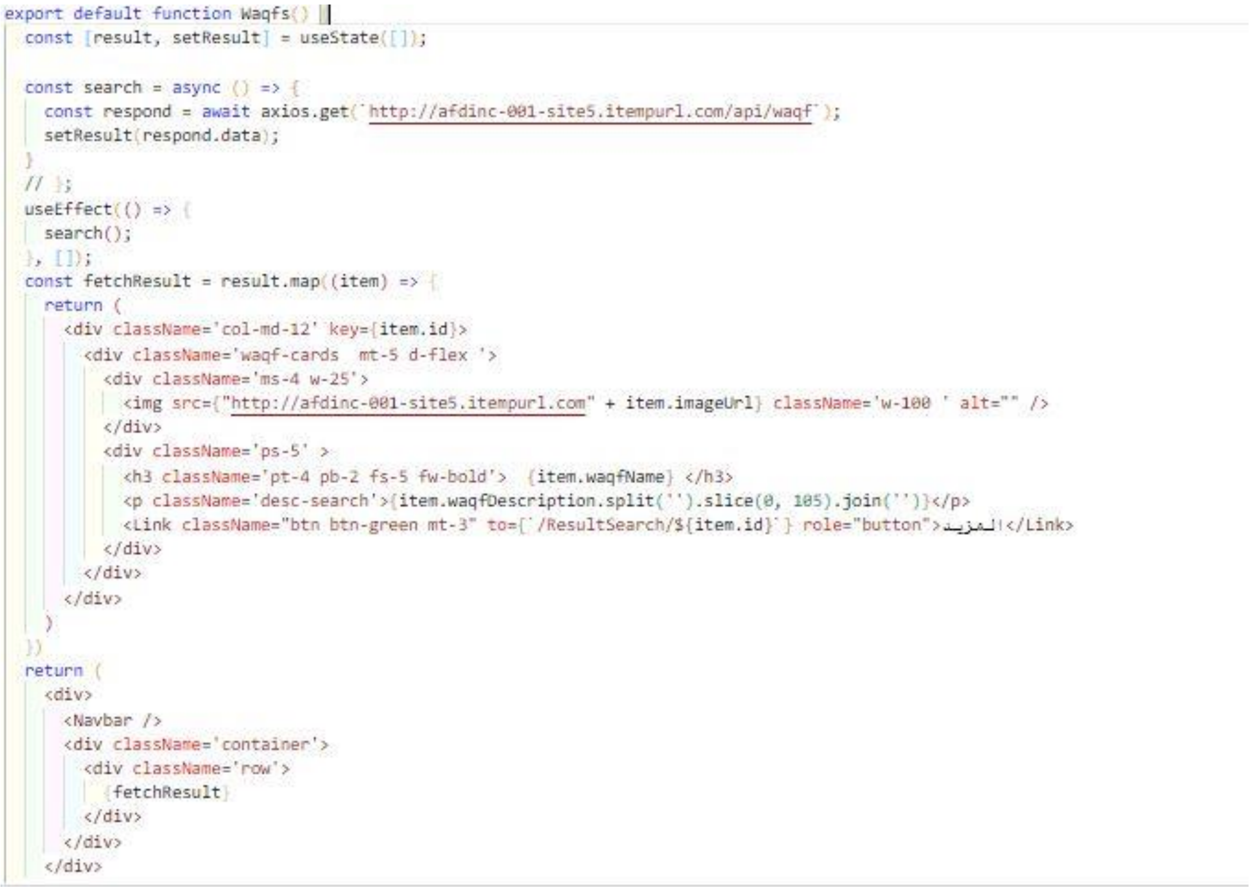
تصنيف الوقف :مصحف

ريع الوقف :مصر-القاهرة

وصف الوقف :الجزء الأول من المصحف الكريم لينتفع به المسلمين سائر وجوه الانتفاعات الشرعية شرط أن يكون مقره بالمدرسة المعروفة بخط التبانة

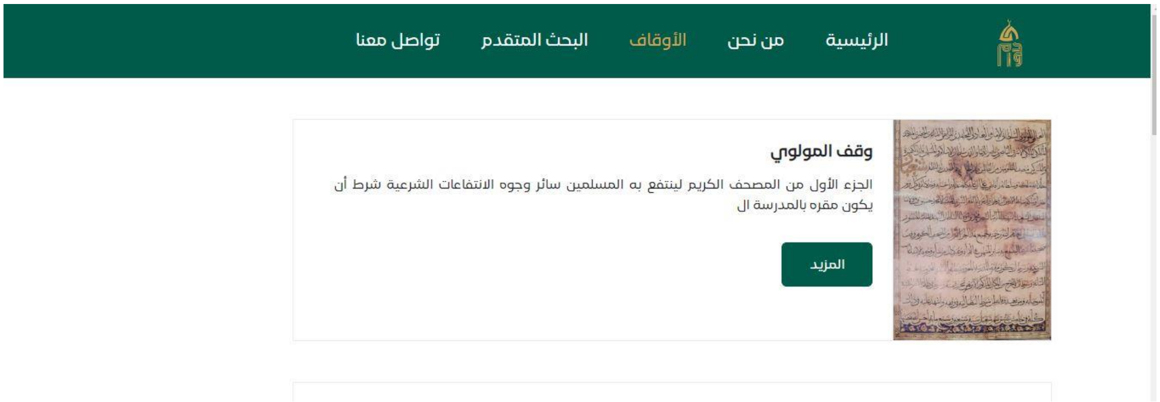
Figure 4-23:showing more details about waqf

This page include all waqfs in DAWAM platform



```
export default function Waqfs() {
  const [result, setResult] = useState([]);

  const search = async () => {
    const respond = await axios.get(`http://afdinc-001-site5.itempurl.com/api/waqf`);
    setResult(respond.data);
  }
  // };
  useEffect(() => {
    search();
  }, []);
  const fetchResult = result.map((item) => {
    return (
      <div className='col-md-12' key={item.id}>
        <div className='waqf-cards  mt-5 d-flex '>
          <div className='ms-4 w-25'>
            <img src={"http://afdinc-001-site5.itempurl.com" + item.imageUrl} className='w-100 ' alt="" />
          </div>
          <div className='ps-5' >
            <h3 className='pt-4 pb-2 fs-5 fw-bold'>  {item.waqfName} </h3>
            <p className='desc-search'>{item.waqfDescription.split('').slice(0, 105).join('')}</p>
            <Link className="btn btn-green mt-3" to={`/ResultSearch/${item.id}`} role="button">المزيد!</Link>
          </div>
        </div>
      </div>
    )
  })
  return (
    <div>
      <Navbar />
      <div className='container'>
        <div className='row'>
          {fetchResult}
        </div>
      </div>
    </div>
```

Figure 4-26:showing waqfs page code

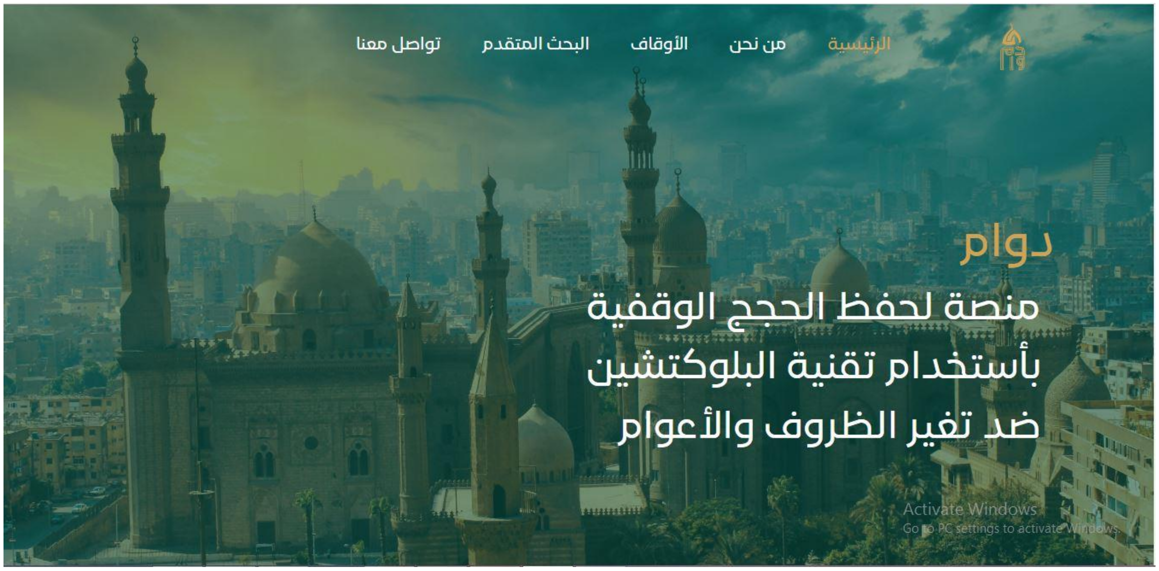


وقف المولوي

الجزء الأول من المصحف الكريم لينتفع به المسلمين سائر وجوه الانتفاعات الشرعية شرط أن
يكون مقره بالمدرسة ال

Figure 4-25:showing waqfs page

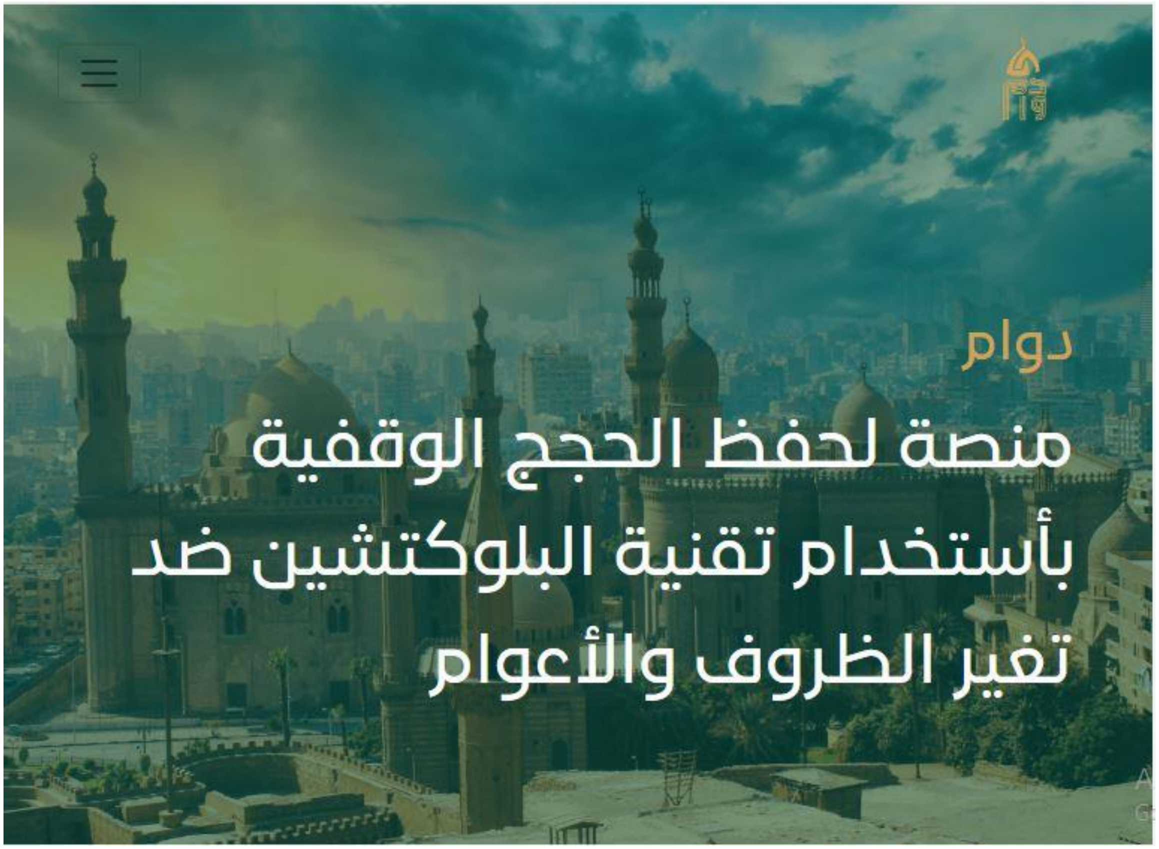
The website is responsive on all screens with different sizes.



دوام

منصة لحفظ الحجج الوقفية
بأستخدام تقنية البلوكتشين
ضد تغير الظروف والأعوام

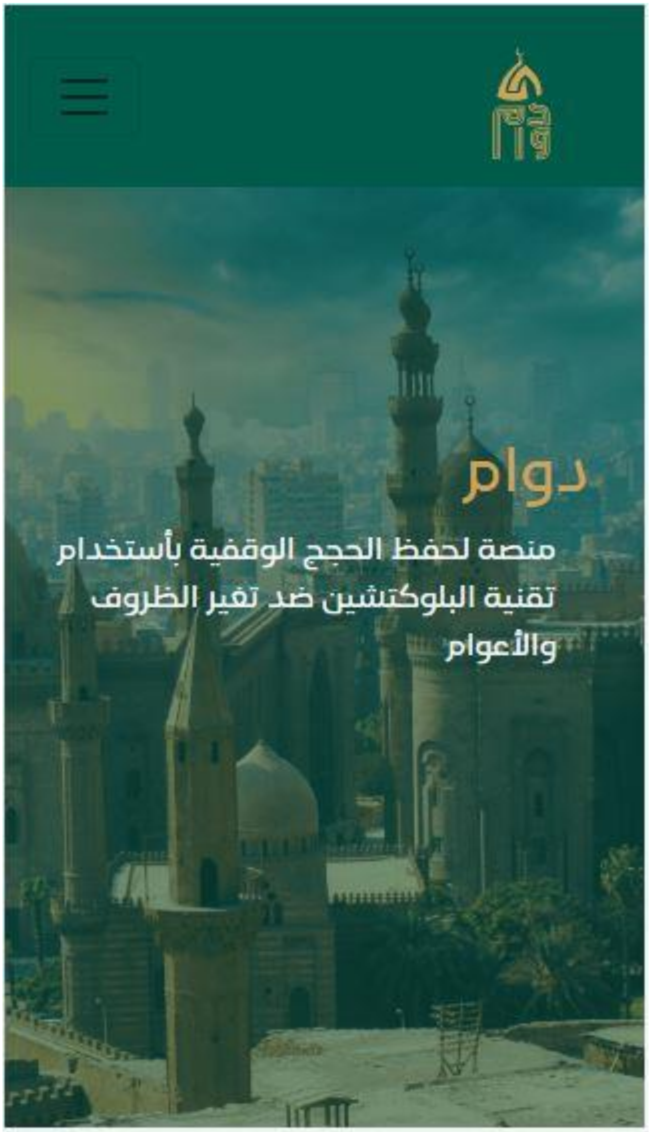
Figure 4-27:showing website in Desktops



دوام

منصة لحفظ الحجج الوقفية
بأستخدام تقنية البلوكتشين ضد
تغير الظروف والأعوام

Figure 4-28:showing website in tablet



دوام

منصة لحفظ الحجج الوقفية بأستخدام
تقنية البلوكتشين ضد تغير الظروف
والأعوام

Figure 4-29:showing website in phone

4.2.2 Dashboard Website

For Supervisors and Admins

Add Waqf

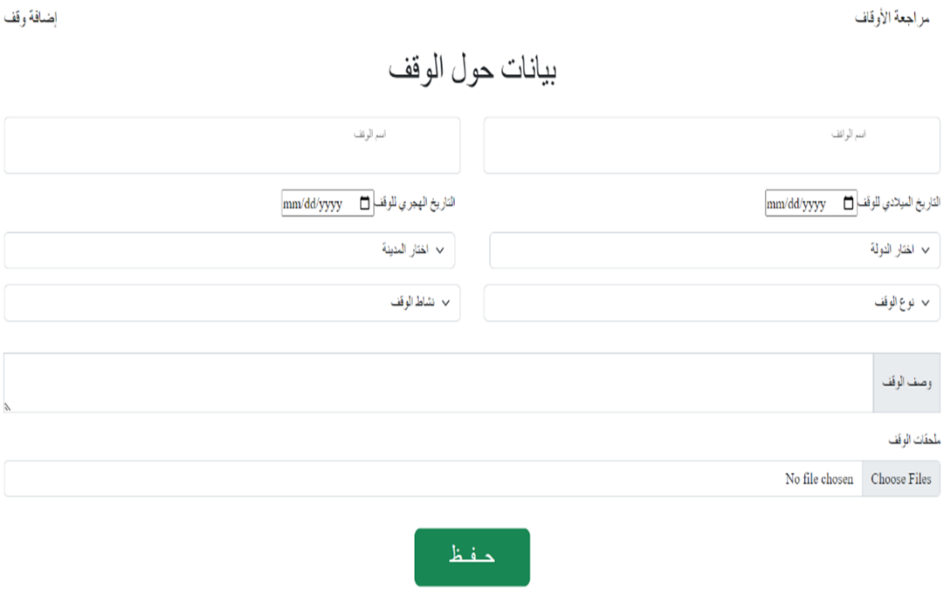


Figure 4-30:Add Waqf

4.3 configuration code for the website (Back-End):

4.3.1 Database mapping

Database was created with code first model with this sample diagram:

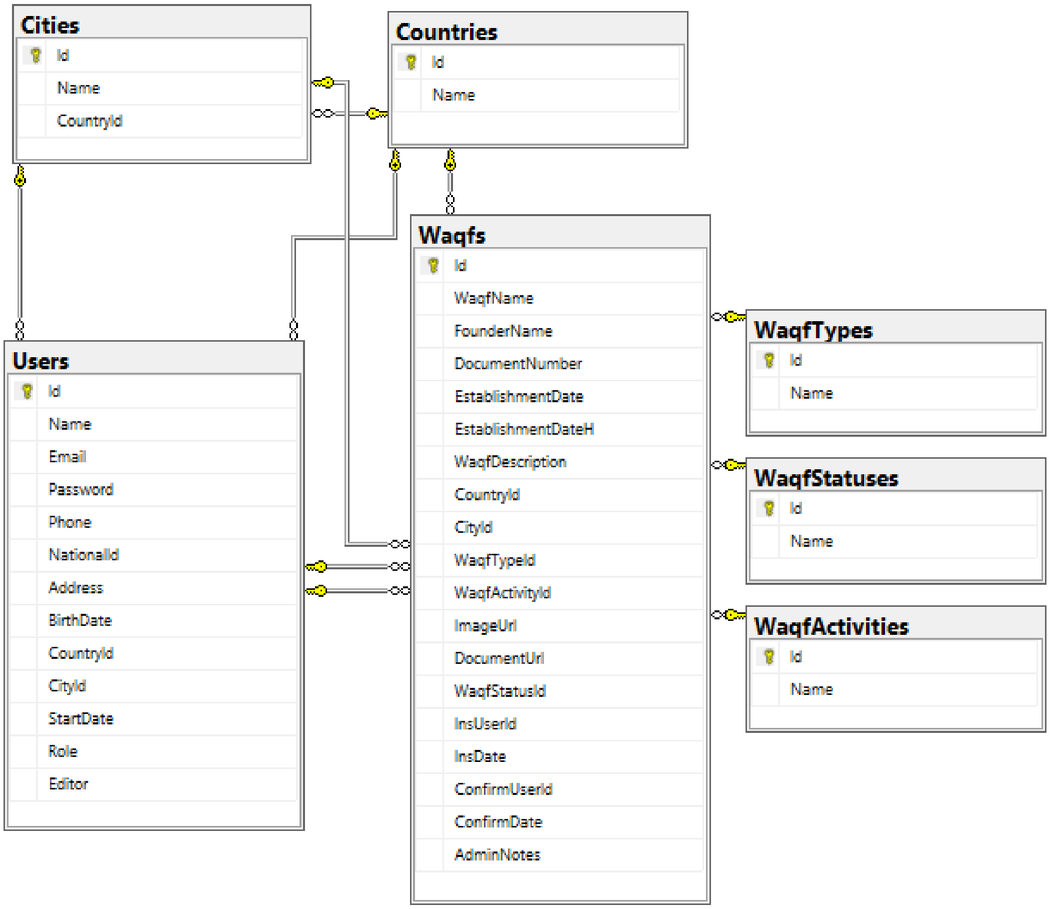
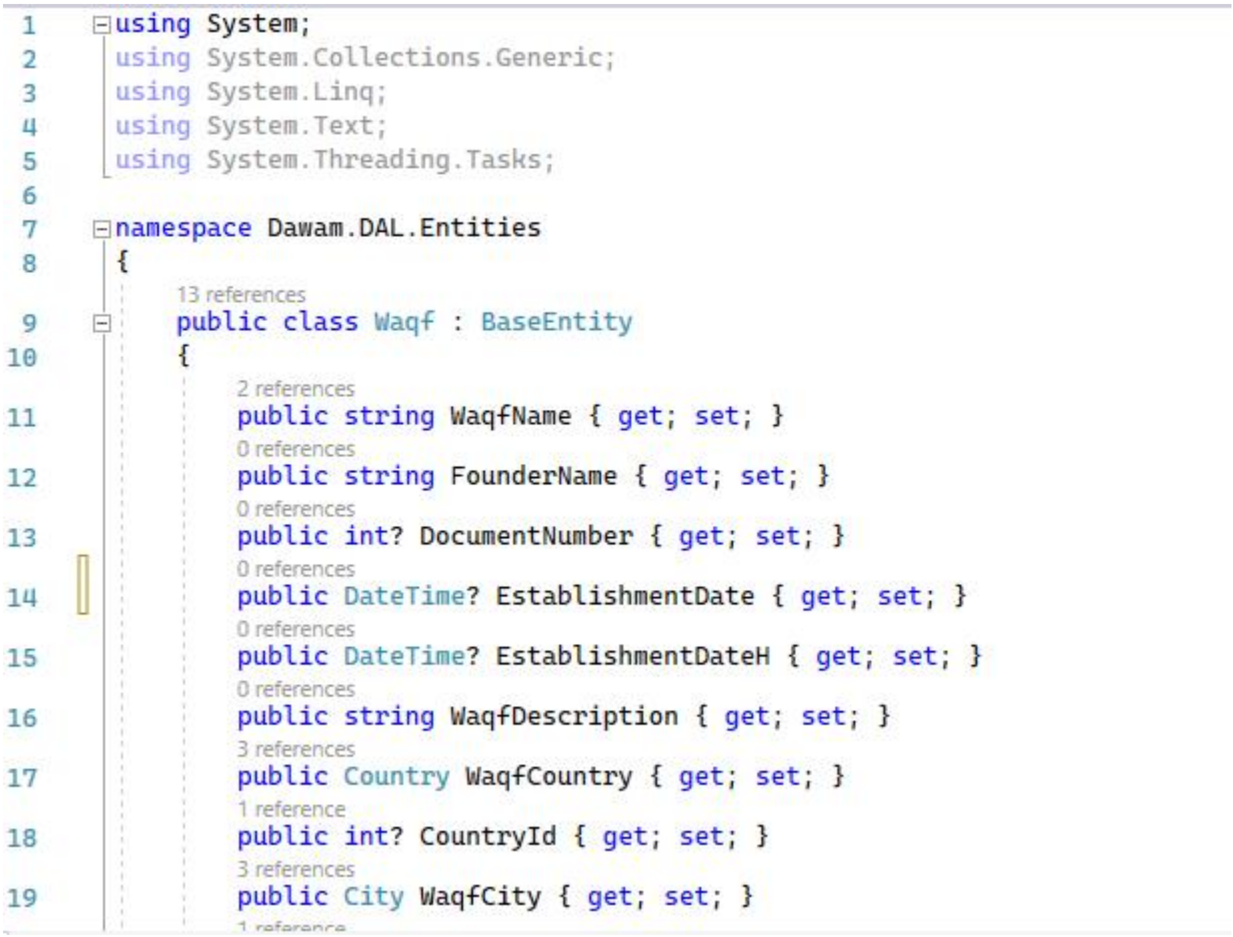
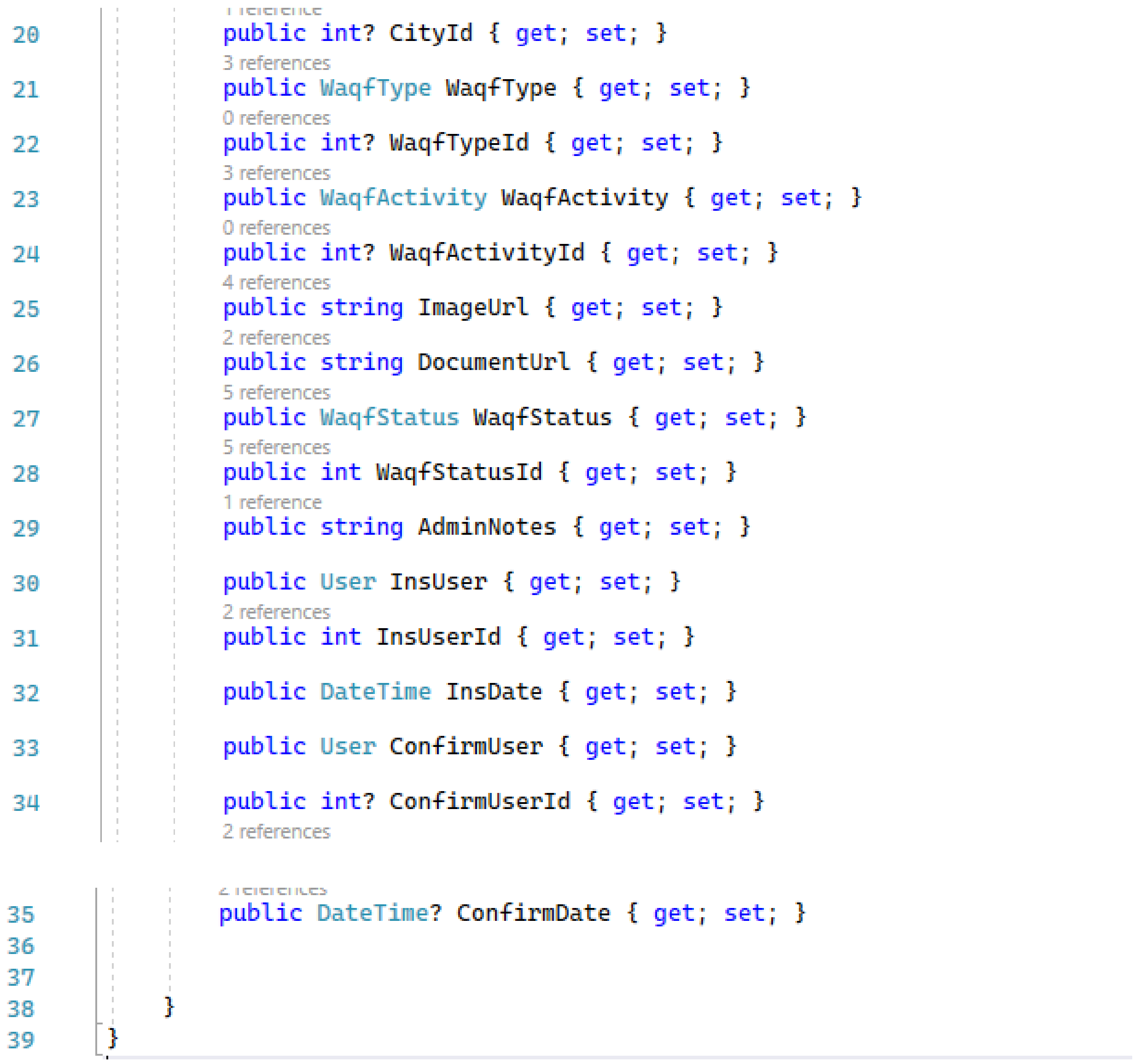


Figure 4-31:Database Sample Diagram

Sample of database entities Waqf entity:



```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Dawam.DAL.Entities
{
    13 references
    public class Waqf : BaseEntity
    {
        2 references
        public string WaqfName { get; set; }
        0 references
        public string FounderName { get; set; }
        0 references
        public int? DocumentNumber { get; set; }
        0 references
        public DateTime? EstablishmentDate { get; set; }
        0 references
        public DateTime? EstablishmentDateH { get; set; }
        0 references
        public string WaqfDescription { get; set; }
        3 references
        public Country WaqfCountry { get; set; }
        1 reference
        public int? CountryId { get; set; }
        3 references
        public City WaqfCity { get; set; }
        1 reference
```



```
        1 reference
        public int? CityId { get; set; }
        3 references
        public WaqfType WaqfType { get; set; }
        0 references
        public int? WaqfTypeId { get; set; }
        3 references
        public WaqfActivity WaqfActivity { get; set; }
        0 references
        public int? WaqfActivityId { get; set; }
        4 references
        public string ImageUrl { get; set; }
        2 references
        public string DocumentUrl { get; set; }
        5 references
        public WaqfStatus WaqfStatus { get; set; }
        5 references
        public int WaqfStatusId { get; set; }
        1 reference
        public string AdminNotes { get; set; }

        public User InsUser { get; set; }
        2 references
        public int InsUserId { get; set; }

        public DateTime InsDate { get; set; }

        public User ConfirmUser { get; set; }

        public int? ConfirmUserId { get; set; }
        2 references

        2 references
        public DateTime? ConfirmDate { get; set; }


    }
}
```

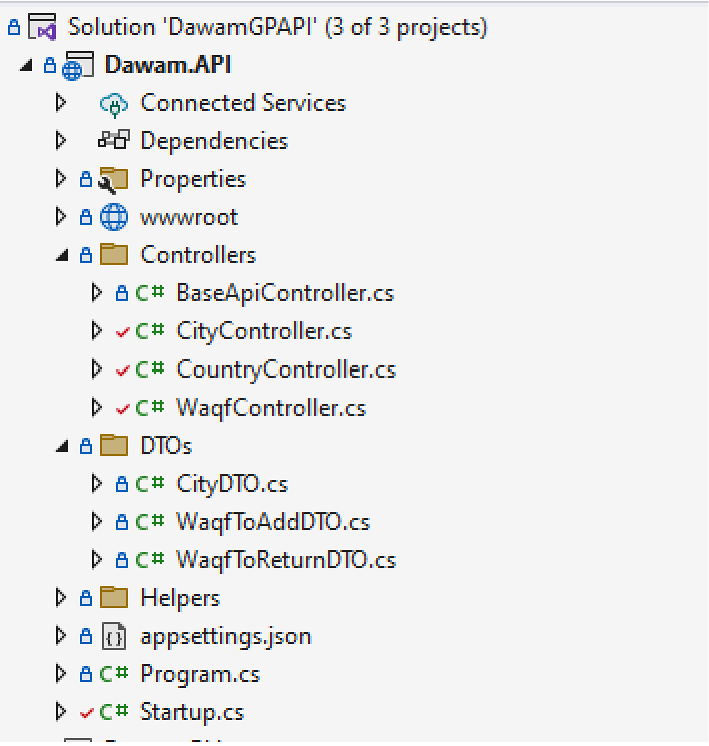
Figure 4-32:Waqf Entity

4.3.2 API Architecture:

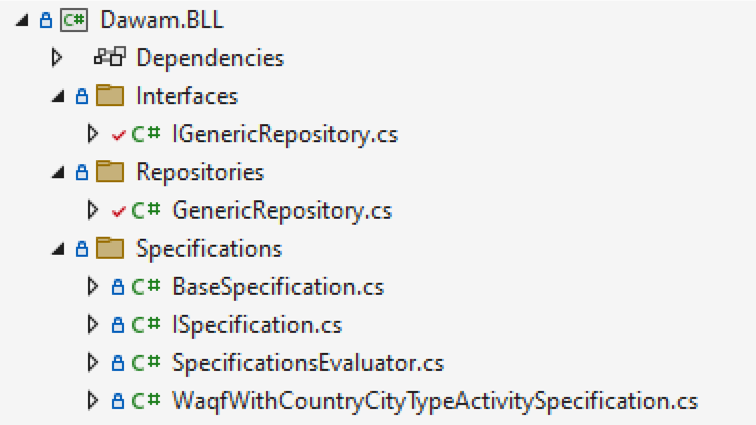
API is created using three-layer project architecture for more security and maintainability the three layers are Data Access layer (DAL), Business Logic Layer (BLL) and Implementation API layer

The business logic layer use Generic and specification Design Patterns

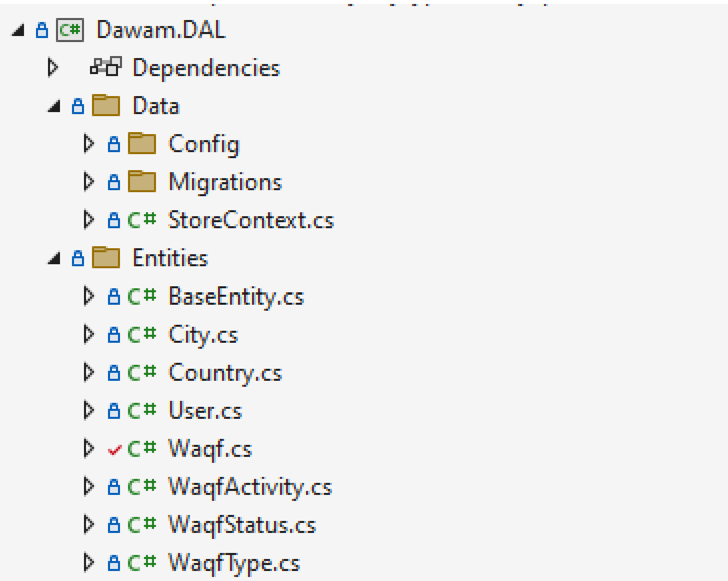
Project Structure is shown as follows:



```
Solution 'DawamGPAPI' (3 of 3 projects)
  Dawam.API
    Connected Services
    Dependencies
    Properties
    wwwroot
    Controllers
      BaseApiController.cs
      CityController.cs
      CountryController.cs
      WaqfController.cs
    DTOs
      CityDTO.cs
      WaqfToAddDTO.cs
      WaqfToReturnDTO.cs
    Helpers
    appsettings.json
    Program.cs
    Startup.cs
```



```
  Dawam.BLL
    Dependencies
    Interfaces
      IGenericRepository.cs
    Repositories
      GenericRepository.cs
    Specifications
      BaseSpecification.cs
      ISpecification.cs
      SpecificationsEvaluator.cs
      WaqfWithCountryCityTypeActivitySpecification.cs
```

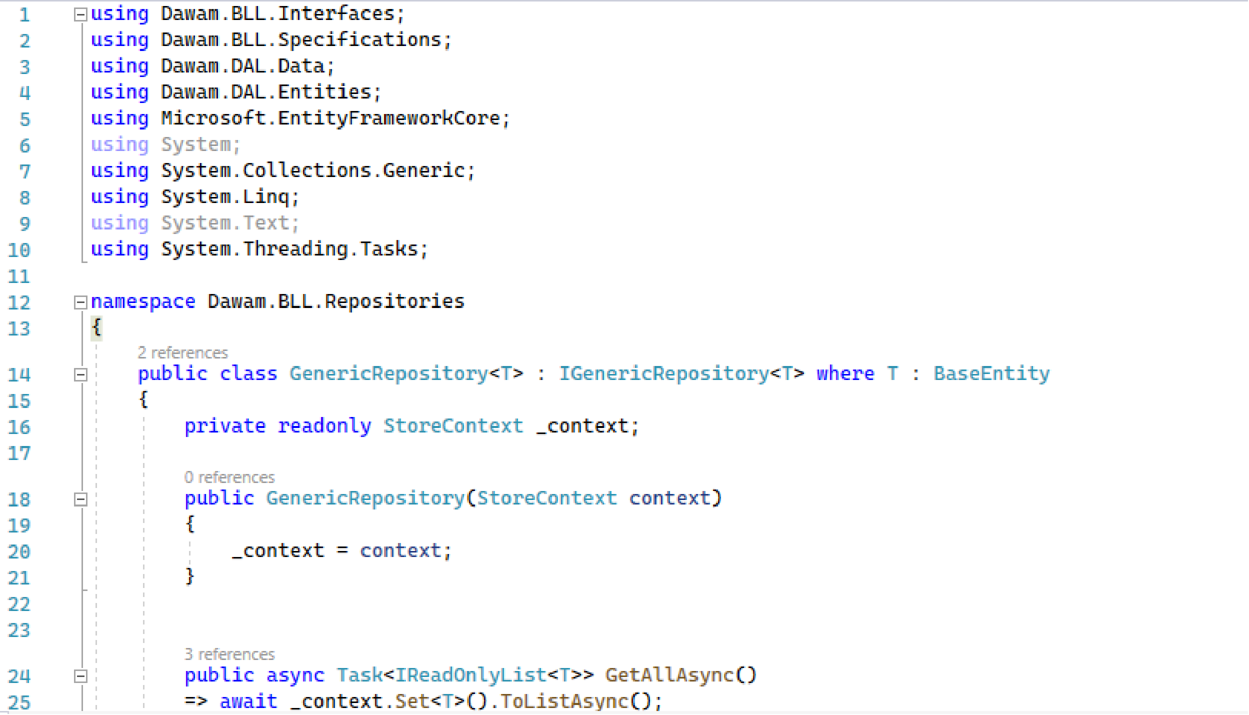


```
Dawam.DAL
    Dependencies
    Data
        Config
        Migrations
        StoreContext.cs
    Entities
        BaseEntity.cs
        City.cs
        Country.cs
        User.cs
        Waqf.cs
        WaqfActivity.cs
        WaqfStatus.cs
        WaqfType.cs
```

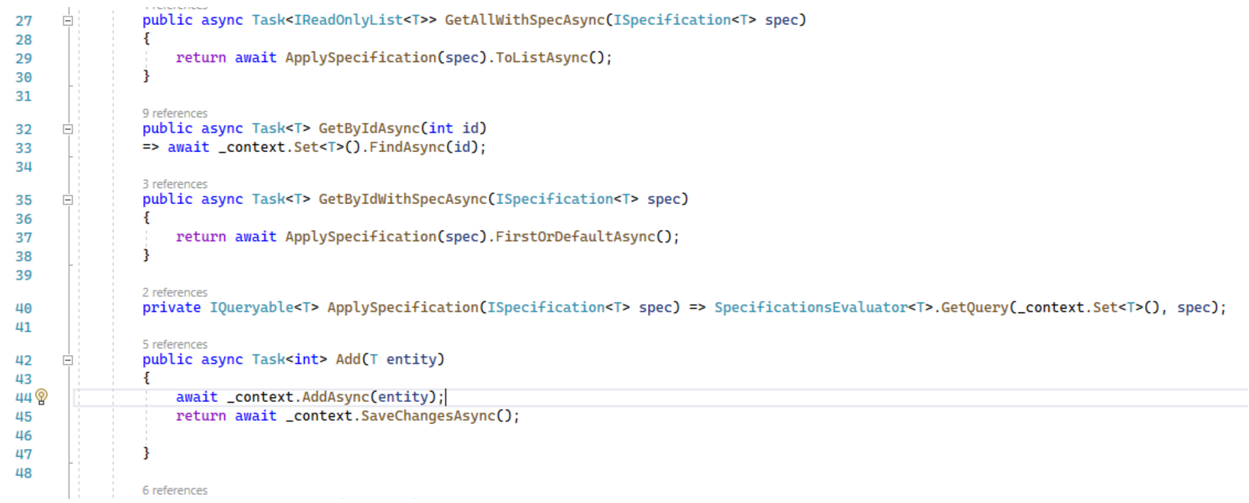
Figure 4-33:Back_end Project Architecture

4.3.3 Implementation sample:

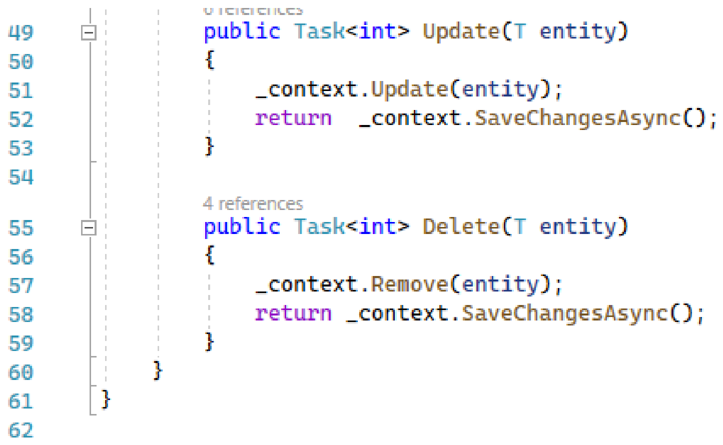
Generic Repository is as shown:



```
using Dawam.BLL.Interfaces;
using Dawam.BLL.Specifications;
using Dawam.DAL.Data;
using Dawam.DAL.Entities;
using Microsoft.EntityFrameworkCore;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Dawam.BLL.Repositories
{
    2 references
    public class GenericRepository<T> : IGenericRepository<T> where T : BaseEntity
    {
        private readonly StoreContext _context;

        0 references
        public GenericRepository(StoreContext context)
        {
            _context = context;
        }


        3 references
        public async Task<IReadOnlyList<T>> GetAllAsync()
        => await _context.Set<T>().ToListAsync();
```



```
        public async Task<IReadOnlyList<T>> GetAllWithSpecAsync(ISpecification<T> spec)
        {
            return await ApplySpecification(spec).ToListAsync();
        }

        9 references
        public async Task<T> GetByIdAsync(int id)
        => await _context.Set<T>().FindAsync(id);

        3 references
        public async Task<T> GetByIdWithSpecAsync(ISpecification<T> spec)
        {
            return await ApplySpecification(spec).FirstOrDefaultAsync();
        }

        2 references
        private IQueryable<T> ApplySpecification(ISpecification<T> spec) => SpecificationsEvaluator<T>.GetQuery(_context.Set<T>(), spec);

        5 references
        public async Task<int> Add(T entity)
        {
            await _context.AddAsync(entity);
            return await _context.SaveChangesAsync();

        }

        6 references
```

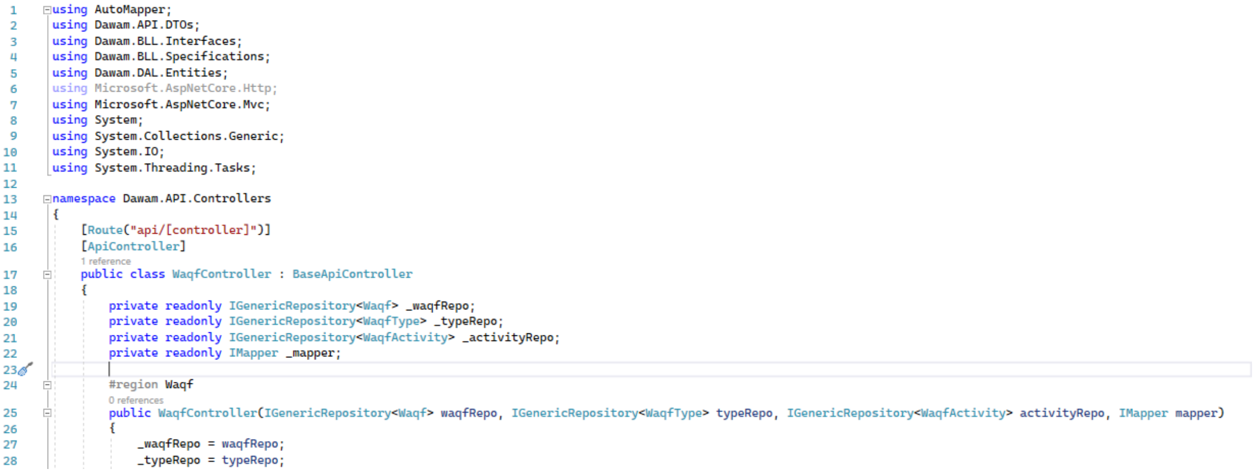


```
            0 references
            public Task<int> Update(T entity)
            {
                _context.Update(entity);
                return  _context.SaveChangesAsync();
            }

            4 references
            public Task<int> Delete(T entity)
            {
                _context.Remove(entity);
                return _context.SaveChangesAsync();
            }
        }
    }
```

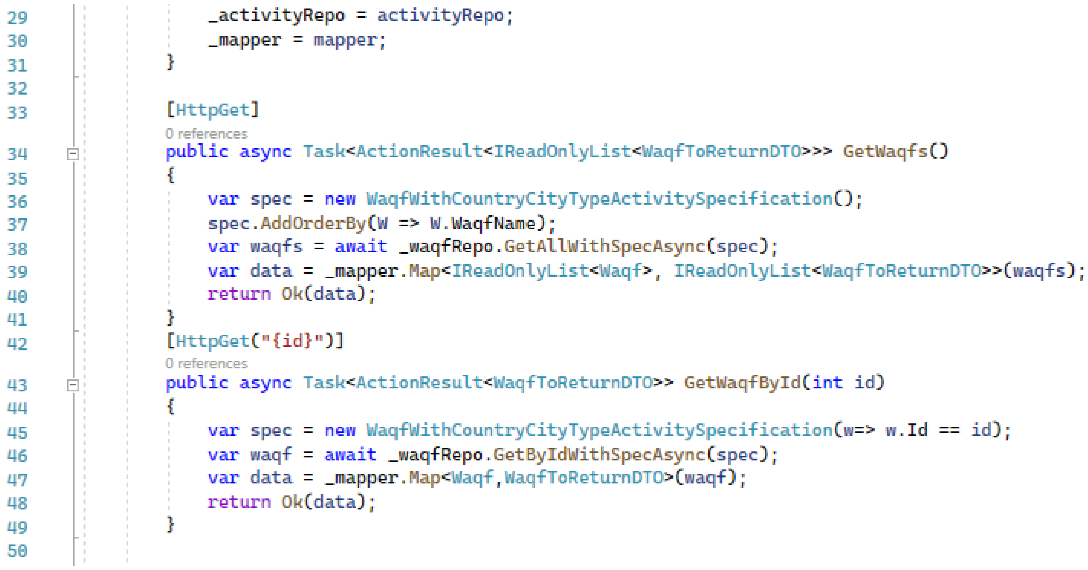
Figure 4-34:Generic Repository code

4.3.4 API Controller model:

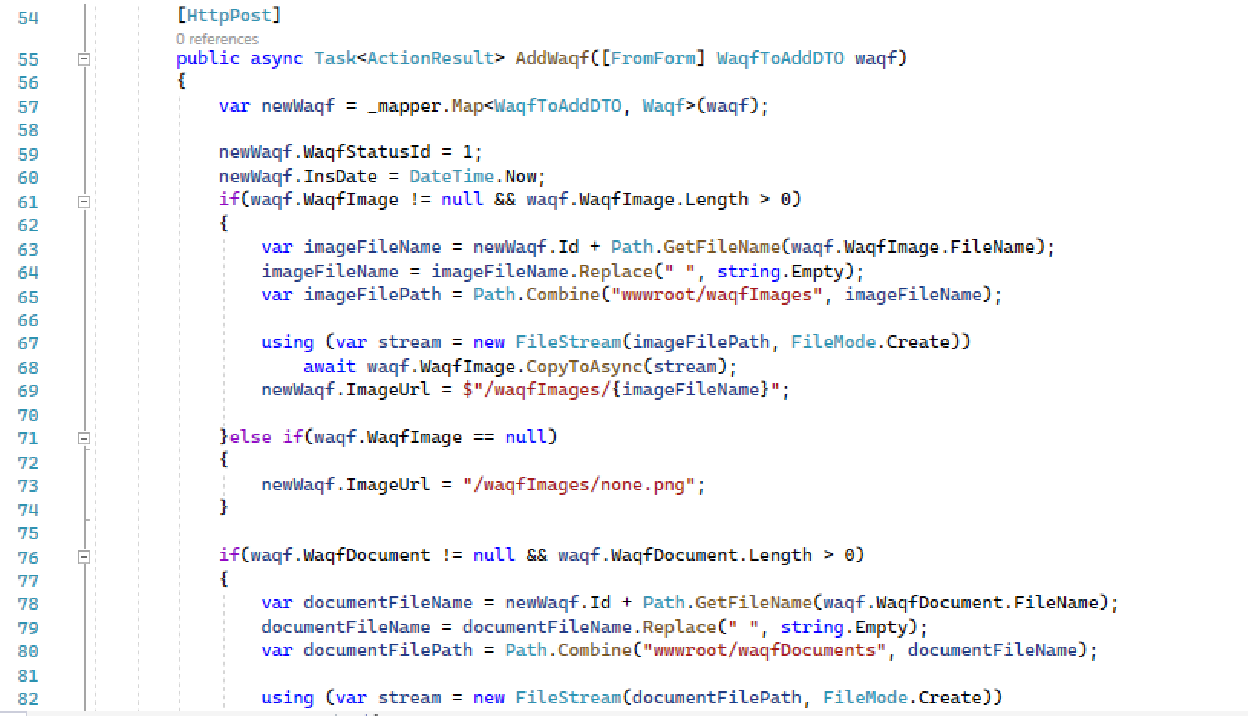
And as a sample waqf controller is as shown:



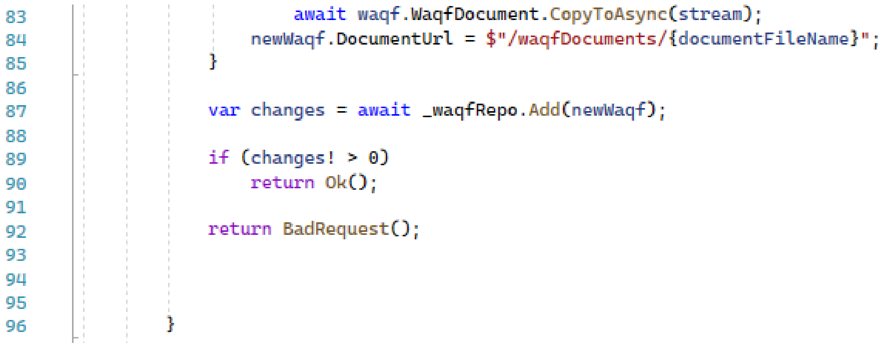
```
using AutoMapper;
using Dawam.API.DTOs;
using Dawam.BLL.Interfaces;
using Dawam.BLL.Specifications;
using Dawam.DAL.Entities;
using Microsoft.AspNetCore.Http;
using Microsoft.AspNetCore.Mvc;
using System;
using System.Collections.Generic;
using System.IO;
using System.Threading.Tasks;

namespace Dawam.API.Controllers
{
    [Route("api/[controller]")]
    [ApiController]
    1 reference
    public class WaqfController : BaseApiController
    {
        private readonly IGenericRepository<Waqf> _waqfRepo;
        private readonly IGenericRepository<WaqfType> _typeRepo;
        private readonly IGenericRepository<WaqfActivity> _activityRepo;
        private readonly IMapper _mapper;

        #region Waqf
        0 references
        public WaqfController(IGenericRepository<Waqf> waqfRepo, IGenericRepository<WaqfType> typeRepo, IGenericRepository<WaqfActivity> activityRepo, IMapper mapper)
        {
            _waqfRepo = waqfRepo;
            _typeRepo = typeRepo;
```



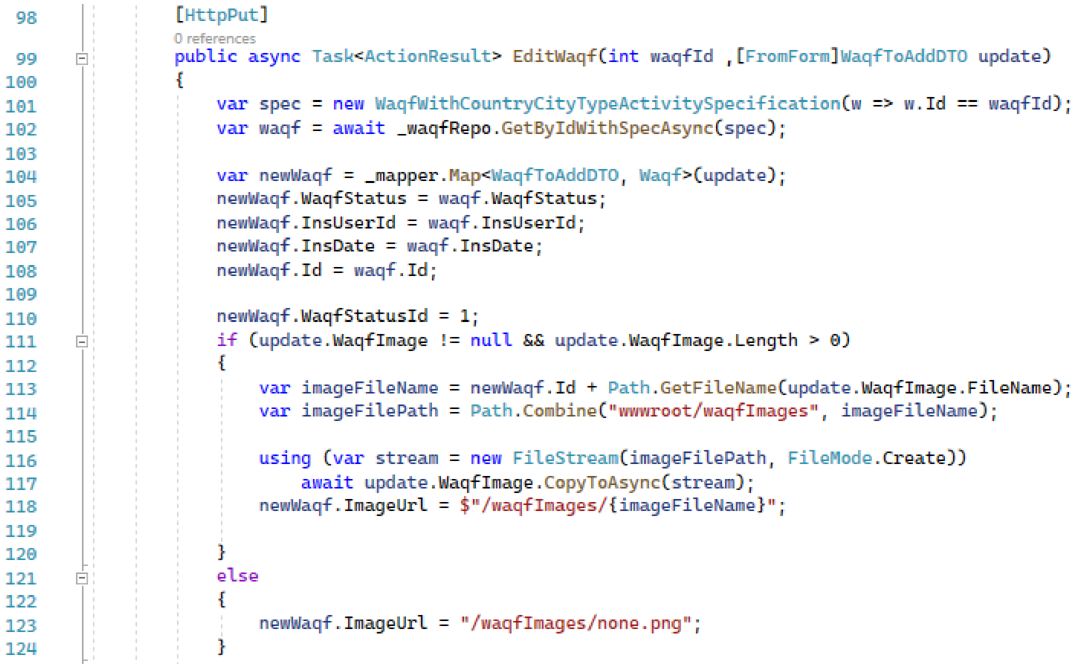
```
            _activityRepo = activityRepo;
            _mapper = mapper;
        }

        [HttpGet]
        0 references
        public async Task<ActionResult<IReadOnlyList<WaqfToReturnDTO>>> GetWaqfs()
        {
            var spec = new WaqfWithCountryCityTypeActivitySpecification();
            spec.AddOrderBy(W => W.WaqfName);
            var waqfs = await _waqfRepo.GetAllWithSpecAsync(spec);
            var data = _mapper.Map<IReadOnlyList<Waqf>, IReadOnlyList<WaqfToReturnDTO>>(waqfs);
            return Ok(data);
        }
        [HttpGet("{id}")]
        0 references
        public async Task<ActionResult<WaqfToReturnDTO>> GetWaqfById(int id)
        {
            var spec = new WaqfWithCountryCityTypeActivitySpecification(w=> w.Id == id);
            var waqf = await _waqfRepo.GetByIdWithSpecAsync(spec);
            var data = _mapper.Map<Waqf,WaqfToReturnDTO>(waqf);
            return Ok(data);
        }

```



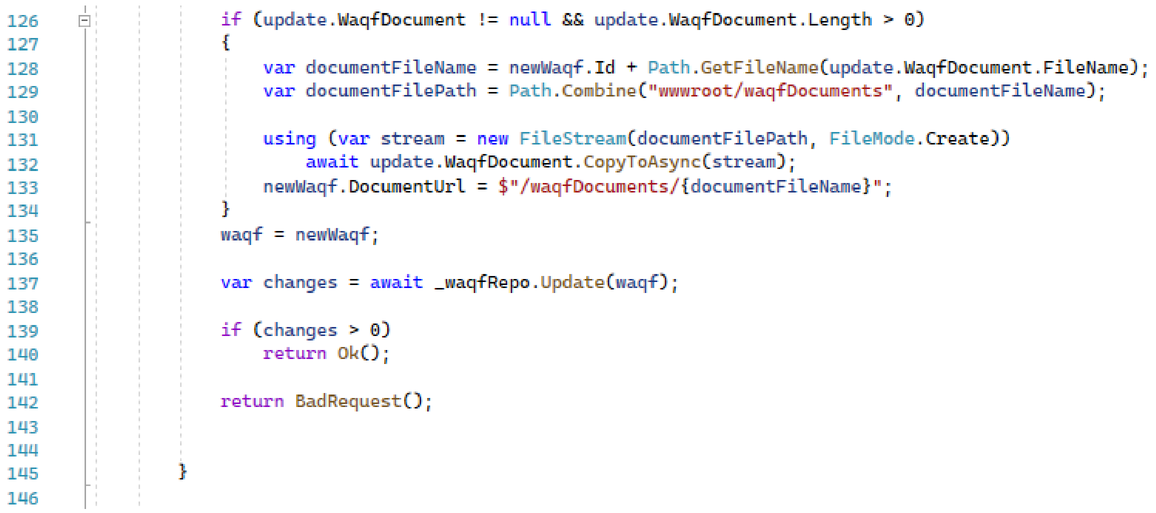
```
[HttpPost]
0 references
public async Task<ActionResult> AddWaqf([FromForm] WaqfToAddDTO waqf)
{
    var newWaqf = _mapper.Map<WaqfToAddDTO, Waqf>(waqf);

    newWaqf.WaqfStatusId = 1;
    newWaqf.InsDate = DateTime.Now;
    if(waqf.WaqfImage != null && waqf.WaqfImage.Length > 0)
    {
        var imageFileName = newWaqf.Id + Path.GetFileName(waqf.WaqfImage.FileName);
        imageFileName = imageFileName.Replace(" ", string.Empty);
        var imageFilePath = Path.Combine("wwwroot/waqfImages", imageFileName);

        using (var stream = new FileStream(imageFilePath, FileMode.Create))
            await waqf.WaqfImage.CopyToAsync(stream);
        newWaqf.ImageUrl = $"/waqfImages/{imageFileName}";

    }else if(waqf.WaqfImage == null)
    {
        newWaqf.ImageUrl = "/waqfImages/none.png";
    }

    if(waqf.WaqfDocument != null && waqf.WaqfDocument.Length > 0)
    {
        var documentFileName = newWaqf.Id + Path.GetFileName(waqf.WaqfDocument.FileName);
        documentFileName = documentFileName.Replace(" ", string.Empty);
        var documentFilePath = Path.Combine("wwwroot/waqfDocuments", documentFileName);

        using (var stream = new FileStream(documentFilePath, FileMode.Create))
```



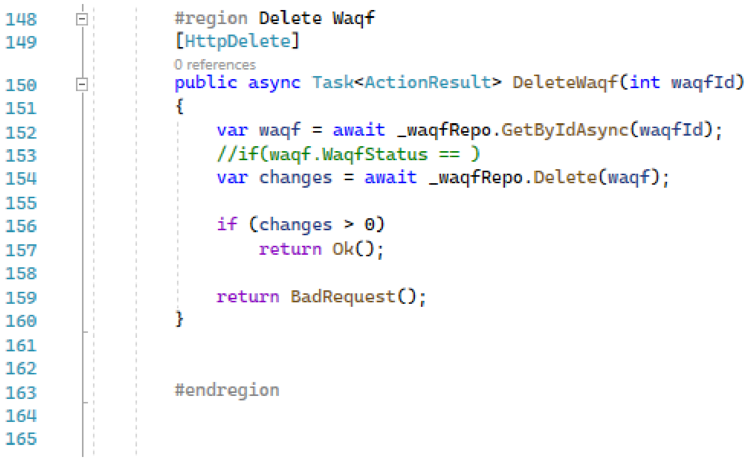
```
            await waqf.WaqfDocument.CopyToAsync(stream);
        newWaqf.DocumentUrl = $"/waqfDocuments/{documentFileName}";
    }

    var changes = await _waqfRepo.Add(newWaqf);

    if (changes! > 0)
        return Ok();

    return BadRequest();



}
```



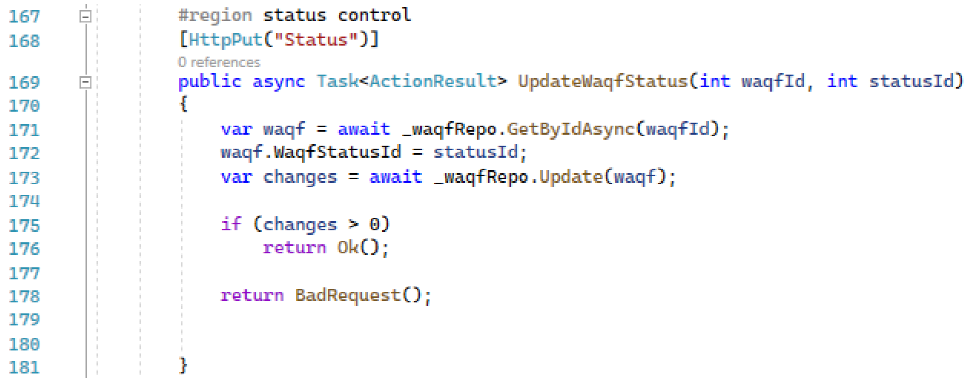
```
[HttpPut]
public async Task<ActionResult> EditWaqf(int waqfId ,[FromForm]WaqfToAddDTO update)
{
    var spec = new WaqfWithCountryCityTypeActivitySpecification(w => w.Id == waqfId);
    var waqf = await _waqfRepo.GetByIdWithSpecAsync(spec);

    var newWaqf = _mapper.Map<WaqfToAddDTO, Waqf>(update);
    newWaqf.WaqfStatus = waqf.WaqfStatus;
    newWaqf.InsUserId = waqf.InsUserId;
    newWaqf.InsDate = waqf.InsDate;
    newWaqf.Id = waqf.Id;

    newWaqf.WaqfStatusId = 1;
    if (update.WaqfImage != null && update.WaqfImage.Length > 0)
    {
        var imageFileName = newWaqf.Id + Path.GetFileName(update.WaqfImage.FileName);
        var imageFilePath = Path.Combine("wwwroot/waqfImages", imageFileName);

        using (var stream = new FileStream(imageFilePath, FileMode.Create))
            await update.WaqfImage.CopyToAsync(stream);
        newWaqf.ImageUrl = $"/waqfImages/{imageFileName}";

    }
    else
    {
        newWaqf.ImageUrl = "/waqfImages/none.png";
    }
```



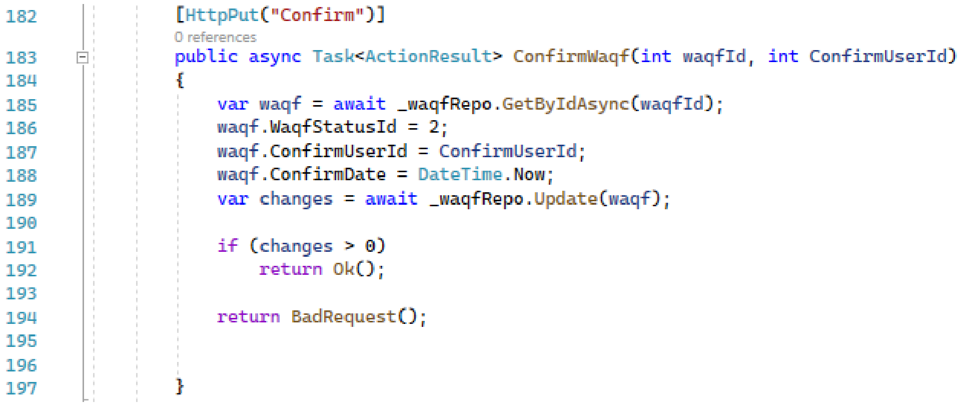
```
    if (update.WaqfDocument != null && update.WaqfDocument.Length > 0)
    {
        var documentFileName = newWaqf.Id + Path.GetFileName(update.WaqfDocument.FileName);
        var documentFilePath = Path.Combine("wwwroot/waqfDocuments", documentFileName);

        using (var stream = new FileStream(documentFilePath, FileMode.Create))
            await update.WaqfDocument.CopyToAsync(stream);
        newWaqf.DocumentUrl = $"/waqfDocuments/{documentFileName}";
    }
    waqf = newWaqf;

    var changes = await _waqfRepo.Update(waqf);

    if (changes > 0)
        return Ok();

    return BadRequest();


}
```



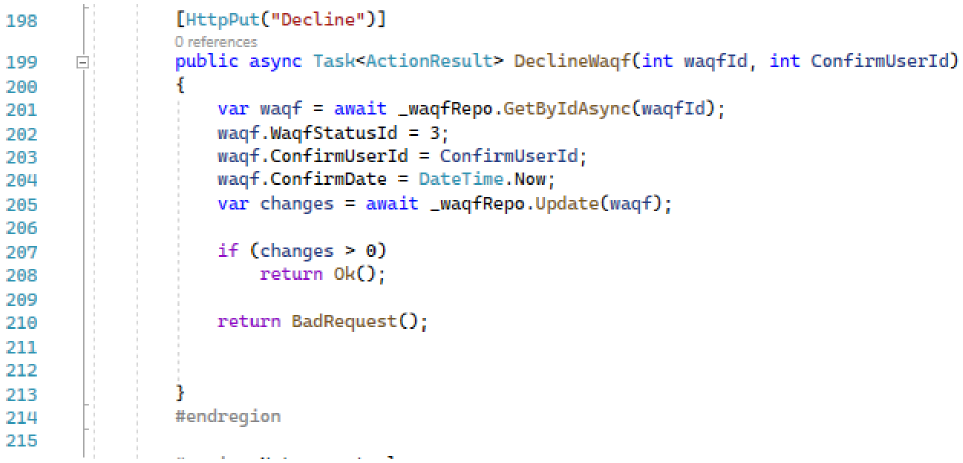
```
#region Delete Waqf
[HttpDelete]
public async Task<ActionResult> DeleteWaqf(int waqfId)
{
    var waqf = await _waqfRepo.GetByIdAsync(waqfId);
    //if(waqf.WaqfStatus == )
    var changes = await _waqfRepo.Delete(waqf);

    if (changes > 0)
        return Ok();

    return BadRequest();
}


#endregion
```



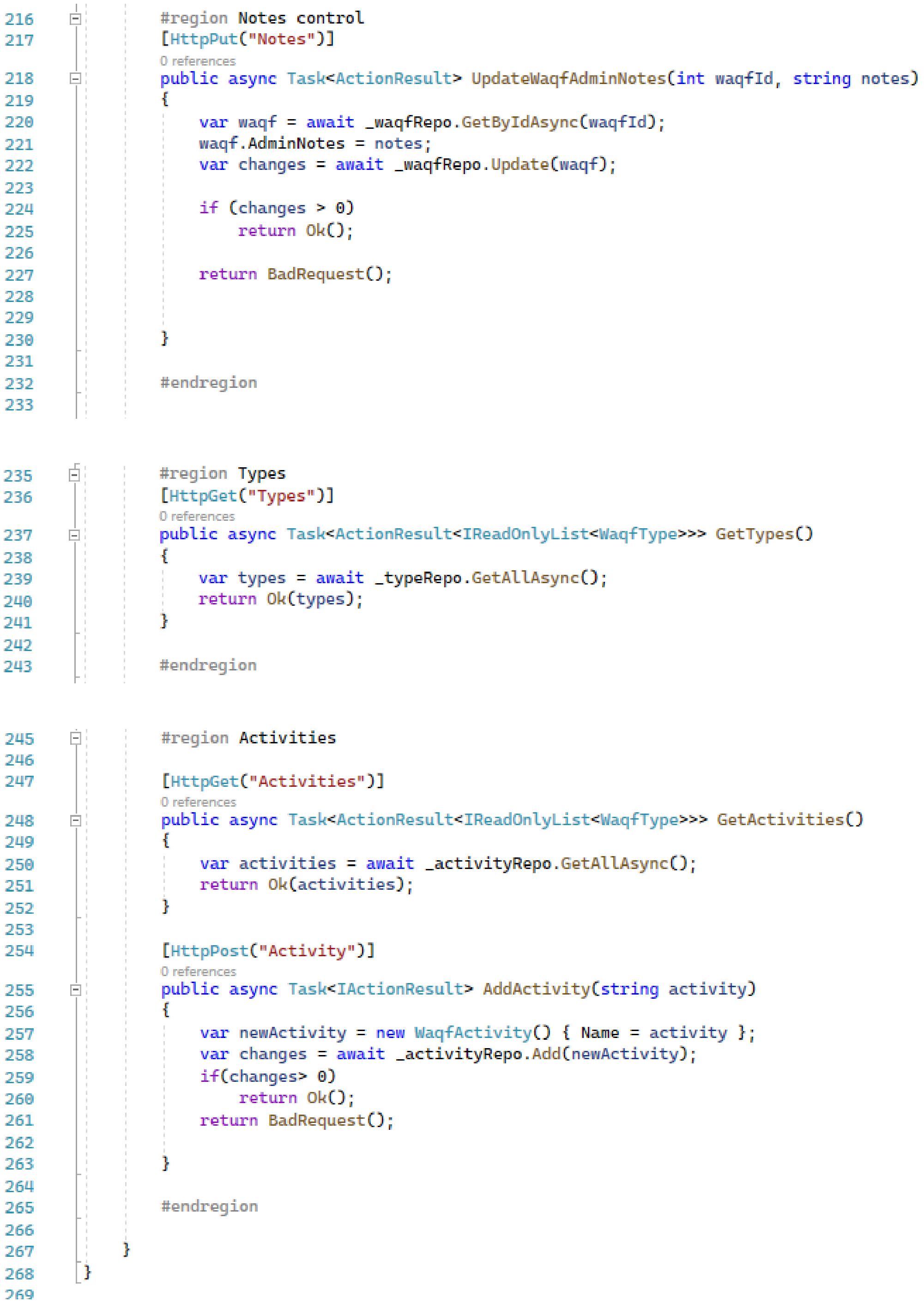
```
#region status control
[HttpPut("Status")]
0 references
public async Task<ActionResult> UpdateWaqfStatus(int waqfId, int statusId)
{
    var waqf = await _waqfRepo.GetByIdAsync(waqfId);
    waqf.WaqfStatusId = statusId;
    var changes = await _waqfRepo.Update(waqf);

    if (changes > 0)
        return Ok();

    return BadRequest();


}
```



```
[HttpPut("Confirm")]
0 references
public async Task<ActionResult> ConfirmWaqf(int waqfId, int ConfirmUserId)
{
    var waqf = await _waqfRepo.GetByIdAsync(waqfId);
    waqf.WaqfStatusId = 2;
    waqf.ConfirmUserId = ConfirmUserId;
    waqf.ConfirmDate = DateTime.Now;
    var changes = await _waqfRepo.Update(waqf);

    if (changes > 0)
        return Ok();

    return BadRequest();


}
```



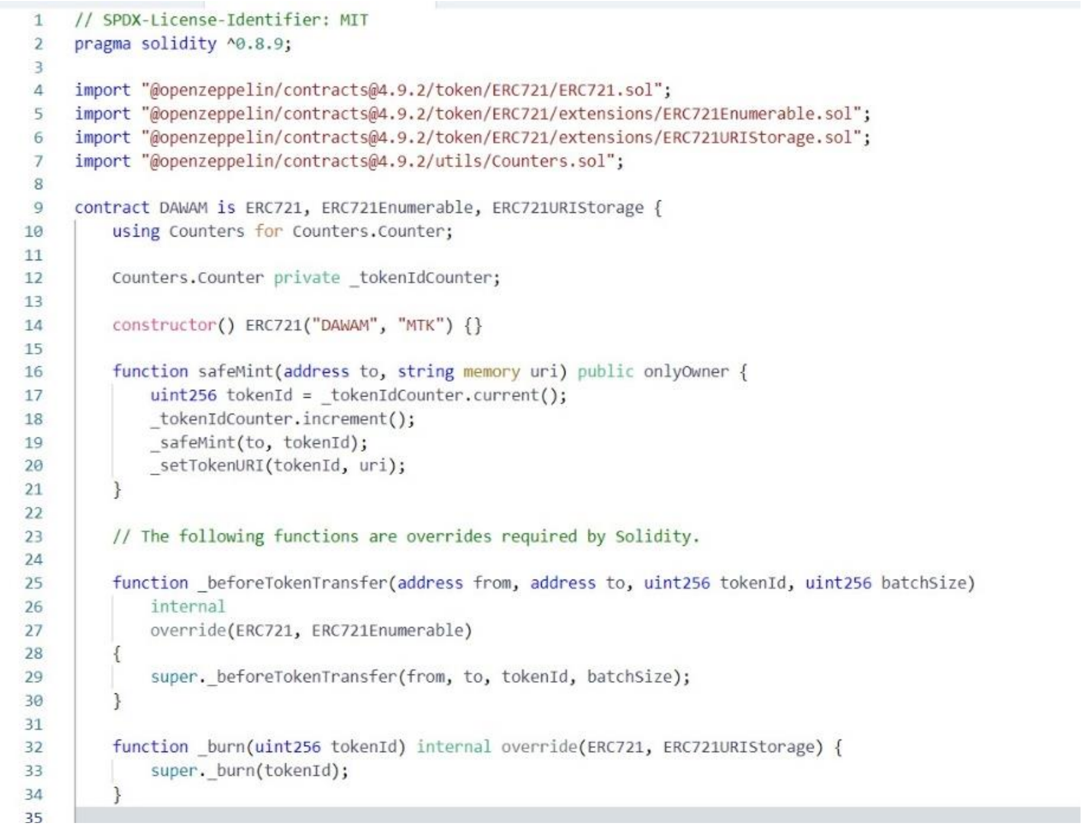
```
[HttpPut("Decline")]
0 references
public async Task<ActionResult> DeclineWaqf(int waqfId, int ConfirmUserId)
{
    var waqf = await _waqfRepo.GetByIdAsync(waqfId);
    waqf.WaqfStatusId = 3;
    waqf.ConfirmUserId = ConfirmUserId;
    waqf.ConfirmDate = DateTime.Now;
    var changes = await _waqfRepo.Update(waqf);

    if (changes > 0)
        return Ok();

    return BadRequest();


}
#endregion
```



```
        #region Notes control
        [HttpPut("Notes")]
        0 references
        public async Task<ActionResult> UpdateWaqfAdminNotes(int waqfId, string notes)
        {
            var waqf = await _waqfRepo.GetByIdAsync(waqfId);
            waqf.AdminNotes = notes;
            var changes = await _waqfRepo.Update(waqf);

            if (changes > 0)
                return Ok();

            return BadRequest();


        }

        #endregion

        #region Types
        [HttpGet("Types")]
        0 references
        public async Task<ActionResult<IReadOnlyList<WaqfType>>> GetTypes()
        {
            var types = await _typeRepo.GetAllAsync();
            return Ok(types);
        }

        #endregion

        #region Activities

        [HttpGet("Activities")]
        0 references
        public async Task<ActionResult<IReadOnlyList<WaqfType>>> GetActivities()
        {
            var activities = await _activityRepo.GetAllAsync();
            return Ok(activities);
        }

        [HttpPost("Activity")]
        0 references
        public async Task<IActionResult> AddActivity(string activity)
        {
            var newActivity = new WaqfActivity() { Name = activity };
            var changes = await _activityRepo.Add(newActivity);
            if(changes> 0)
                return Ok();
            return BadRequest();

        }

        #endregion

    }
}
```

Figure 4-35:waqf controller code

4.3.5 ERC721 NFT creating smart contract sample:



```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.9;

import "@openzeppelin/contracts@4.9.2/token/ERC721/ERC721.sol";
import "@openzeppelin/contracts@4.9.2/token/ERC721/extensions/ERC721Enumerable.sol";
import "@openzeppelin/contracts@4.9.2/token/ERC721/extensions/ERC721URIStorage.sol";
import "@openzeppelin/contracts@4.9.2/utils/Counters.sol";

contract DAWAM is ERC721, ERC721Enumerable, ERC721URIStorage {
    using Counters for Counters.Counter;

    Counters.Counter private _tokenIdCounter;

    constructor() ERC721("DAWAM", "MTK") {}

    function safeMint(address to, string memory uri) public onlyOwner {
        uint256 tokenId = _tokenIdCounter.current();
        _tokenIdCounter.increment();
        _safeMint(to, tokenId);
        _setTokenURI(tokenId, uri);
    }

    // The following functions are overrides required by Solidity.

    function _beforeTokenTransfer(address from, address to, uint256 tokenId, uint256 batchSize)
        internal
        override(ERC721, ERC721Enumerable)
    {
        super._beforeTokenTransfer(from, to, tokenId, batchSize);
    }

    function _burn(uint256 tokenId) internal override(ERC721, ERC721URIStorage) {
        super._burn(tokenId);
    }

```

Figure 4-36:ERC721 contract code

CHAPTER 5: Conclusion and Future Improvements

5.1 Conclusion

The system for saving Waqf documents using blockchain technology has the potential to revolutionize the way in which these important historical records are preserved and accessed. By leveraging the security and transparency features of blockchain technology, the system can provide a secure, reliable, and user-friendly environment for storing and managing Waqf documents.

Through the research and development of this project, we have identified the key challenges faced in preserving Waqf documents and how blockchain technology can help overcome these challenges. We have also defined the functional and non-functional requirements for a system that can effectively preserve Waqf documents using blockchain technology.

The system developed as part of this graduation project book provides a proof-of-concept for the use of blockchain technology in preserving Waqf documents. It demonstrates the potential for this technology to provide a secure and efficient way to manage and access these important historical records.

5.2 Future Work:

One of our project advantages is that we use future technology that could be expanded and implemented in different fields.

We can improve and expand our project in two ways, first: we can use it to secure other documents such as documents for land registry, real states, property contracts and university documents and more.

The second way is to add more smart contracts to our waqf project to support investing in waqf project With the trusted autonomous smart contracts for crowdfunding.

5.3 References

Al-Salahat, S. M., 2021. Endowment and Blockchain Technology: Invest and Finance from the Sharia Perspective.

Sasse, H., 2019. Use of Blockchain Applicationsin Developing Waqf: A Platform by Finterra Companies as aModalStudy. *Journal of Islam in Asia,* pp. 20-29.