Palestine polytechnic university

College of information technology and computer engineering

Department of Information Technology and Computer Science

Second Hand Store

(فيد و استفيد)

Team members:

Diana Mujahed Yafa Farajallah

Supervised by:

Mohammed Jabari

2019-2020

نبذة مختصرة

معظم الأشخاص يسيئون استخدام أغراضهم المستعملة إما برميها بعيدًا أو الاحتفاظ بها دون الاستفادة منها. يرغب معظمهم في التخلص من أغراضهم المستخدمة بطريقة مناسبة ، لكنهم لا يعرفون كيف ، لذلك يركز هذا المشروع على هذه المشكلة. أنشأنا موقعًا على شبكة الإنترنت يسمح للأشخاص بعرض أغراضهم المستعملة التي يريدون التخلص منها عن طريق التبرع بها أو بيعها أو تأجيرها. كما يزود المستخدمين بالموقع الدقيق للأغراض المستخدمة المعروضة ، حتى يتمكنوا من التواصل مع المالك ثم الذهاب واستلامها ، أو يمكنهم طلب تسليمها إليهم إذا قدم البائع هذا الخيار ، كما يمكن للجمعيات الخيرية الاشتراك في الموقع لجمع التبرعات بسهولة.

تكمن أهمية هذا الموقع في أنه سيكون نقطة تقاطع لكل من يرغب في شراء أو بيع أو تأجير الأغراض المستعملة ، لذلك سيكون الناس على دراية بمكان مناسب لعرض الأغراض المستخدمة التي لن يستخدموها مرة أخرى ، والمشتري الذي يريد شراء سلع مستعملة بسعر قليل يعرف مكان البحث الأمثل، ويستهدف الموقع جميع السكان الفلسطينيين والأفراد والمؤسسات الخيرية.

بعد إنشاء الموقع ، قمنا باختبار ما إذا كان يحقق المتطلبات الوظيفية و غير الوظيفية التي تم تحديدها في بداية المشروع و لقد تحققت جميعها.

Abstract

Recently, we have recognized that people misuse their used items by either throwing them away or keeping them without getting benefit of them. Most of them wish to get rid of their used items in a proper way, but they don’t know how, so this project focuses on this issue. We built a website that allows people to exhibit their used items that they want to get rid of by donating, selling or renting them . It also provides users with the exact location of the exhibited used items, so they can communicate with the owner and then go and pick them up, or they can ask to be delivered to them if the user provides this option, also Charities can subscribe to the site to collect donation easily.

The importance of this website is that it will be a cross point area for all who wants to buy, sell or rent used items, so people will be aware where to exhibit their used items that they will not use again, and buyer who wants low price and used things knows where to search, the website targets all Palestinian residents, individuals and charitable organizations.

After creating the site, we tested whether it fulfills the functional and non-functional requirements that were identified at the start of the project and they all met.

List of contents

List of figures

List of tables

Chapter 1: introduction.

1.1 Problem statement

As it is known, if we want to get rid of a used item in order to buy a newer one, the best act is to sell or donate it, but not to throw it in random places and mess the environment.

However, the bad disposal of used items without giving such items to people who might need them is a common issue.

In response to this problem, we have created a website that allows people to exhibit their used items in order to donate, sell or rent them with another kind of used items which exhibited in the website.

1.2 Objectives

1. Save the environment of the country through reducing the mess resulting from throwing the used items in public places.
2. Help charities to collect donations, where the website will be a primary source for them.
3. Giving the opportunity for poor people to get things with low prices.
4. Giving the opportunity to the buyer to bring what he buys by himself, this saves him the trouble of dealing with a third party for delivery.
5. Help people in taking benefits of their used things that they would like to get rid of.

1.3 Scope

The website targets all Palestinian residents, individuals and charities.

1.4 Context diagram

The context diagram shows the context that the website works with, it consists of inputs, processes and outputs, as well as how Interaction of people and other websites is done with the proposed website.

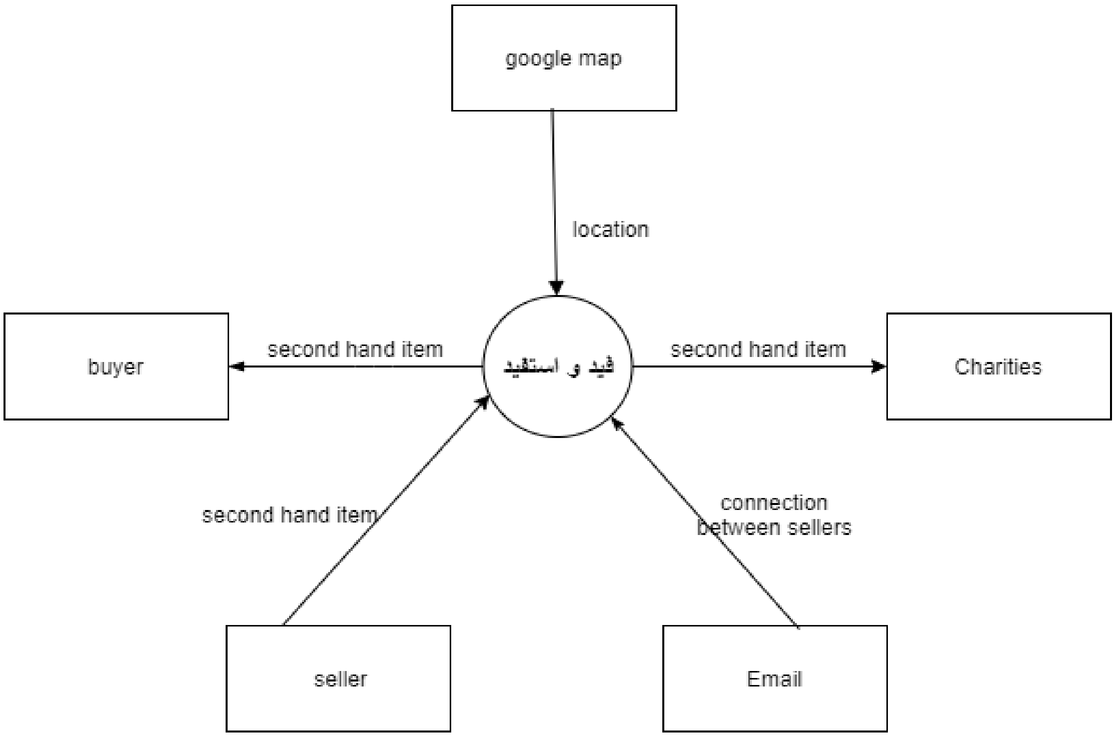


Figure 1: context diagram

1.5 The importance of the website

The Importance for seller:

1. Get rid of his used items in the right place.
2. Getting a financial return from selling his used items.
3. A convenient website for people who want to donate their items, but they don’t know people who in need.

The importance for buyer:

1. The site is an attractive selling site for people who need to buy cheap items.
2. Elimination of distrust barriers to purchase through websites resulting from a delivery service, where the seller attaches the exact geographical location with the item.

The importance for charities organization:

1. Considered as a rich source of donations for distribution to the poor and orphans.
2. A way to connect with people who want to donate their used items.

1.6 Methodology

The team followed one of software engineering methodologies used to achieve the expected outputs of the website, followed the waterfall style, this methodology is known as SDLC waterfall, which starts from the planning stage of the website and then requirements analysis followed by website design, website development, operation, and testing and its maintenance.

1.7 Role distribution

Both members participated in all phases of website development, the team consists of two members between programming, information gathering, analysis and other tasks in order to bring the project to a high level of efficiency.

1. The first role: gathering information and then planning for the website.

2. The second role: determining the requirements of the website based on the information collected.

3. Third Role: website requirements analysis.

4. Fourth role: website design.

5. Fifth role: programming and development for the website.

6. Sixth role: testing the website and ensure that it includes all requirements.

7. Seventh floor: documentating the website.

1.8 Limitations and risks

There are two limitations that have been taken into account during the process of building and operating the website:

1. Build the website within the specified period.

2. Availability of a server to host the website.

1.9 Timeline /Project Scheduling

Table 1: project tasks

Task number	Task name	Time required in weeks
1.	Website planning and information gathering	4
2.	website requirements determining	4
3.	Describe and analyze website requirements	4
4.	Website design	6
5.	Programming and website development	8
6.	Website testing	4
7.	Documenting	Along the working period

task	Time in weeks															
	First semester								🟨	Second semester						
	2	4	6	8	10	12	14	16	🟨	2	4	6	8	10	12	16
Website planning and information gathering	🟩	🟩							🟨							
website requirements determining			🟩	🟩					🟨							
Describe and analyze website requirements					🟩	🟩			🟨							
Website design							🟩	🟩	🟨	🟩						
Programming and website development									🟨		🟩	🟩	🟩	🟩		
Website testing									🟨			🟩	🟩	🟩	🟩	🟩
Documenting	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟨	🟩	🟩	🟩	🟩	🟩	🟩	🟩

Figure 2: project timeline

🟩 Estimated time to finish the task	🟨 Holyday between semesters

1.10 Website Development Resources

During the process of building the website we have a set of supplies necessary to complete the website construction, namely:

Software Sources: Includes all software components required for website development.

Table 2:software sources

Programming component	Quantity
Windows 10	2
Microsoft Office 2016	2
Visual Studio Code	2
Adobe XD	2
postman	2
Xamp server	2

Physical resources: Includes the physical components needed to develop and build the website.

Table 3: Physical Sources

Physical components	Specifications	count
Computer device	Core i3 laptop, 6 GB RAM	2

Human Resources:

This includes the team working on building the website.

1. Database Developer: Build website databases and create relationships between tables.
2. Website Designer: Fully designed website interfaces.
3. Website Programmer: Work on programming and building the application

Chapter 2: Requirements Specification.

2.1 Introduction

In this chapter, users of the website, functional and non-functional requirements, use case diagram, class diagram and detailed description of website requirements will be presented.

2.2 Alternative solutions

1. Website.

1.1) Features of this website:

- Easy to use by users.
- Access to the site from any type of devices (computers and smart devices).
- Easy to update at any time.
- Can be opened from a browser running on any operating system.
- Ability to make the website responsive when it is opened from mobile

1.2) Disadvantages of this website:

- Permanent need to connect to the Internet.

2. Mobile Application.

2.1) Features of this aplication:

- Ease of use.
- Available always to the user through the mobile phone.
- Easy to update.

2.2) Disadvantages of this aplication:

- Permanent need to connect to the Internet.
- People avoid to download apps due to space limitations on their phones
- Diversity of phone operating systems.

2.3 Proposed system.

After looking at the solutions that can be selected, a website has been chosen to be used by the provider and the consumer, because it is the best solution in terms of reaching the largest group of the community, and anyone can open the website and use it from his phone.

2.4 Website users.

1. Visitor: can be any person who visits the site and views all items in it.
2. User: can sell, buy, donate or rent used items.
3. Website administrator: responsible for reviewing reports/complaints about exhibited used items, then decide to delete them or not.

2.5 Website requirements.

Functional requirements

For visitor:

1. Ability to view/browse all exhibited items using filters.
2. Ability to create user account.
3. Ability to contact any user.

For user:

1. Ability to log in to his account.
2. Ability to view/browse all exhibited items using filters.
3. Ability to contact any user.

4. Ability to rate any user account.
5. Ability to do CRUD (Create,Read,Update,Delete) operations on his items.
6. Ability to edit his account information.
7. Ability to report/complain any inappropriate item.
8. Ability to log out from his account.

For website administrator:

1. Reviewing reports/complaints about any exhibited item from website users.
2. Ability to control any account or item.

Nonfunctional requirements.

Non-functional requirements are the basis for the success of the functional requirements because they are based on their support and quality, either directly or indirectly. Non-functional requirements include the following:

A) Ease of dealing with the website.

1. The website design interface should be easy to handle and use, to access all website features easily, as the user will be able to fully handle the website within 15 minute of using it.
2. The interface should be simple, with comfortable colors for the user
3. The website should be responsive when it is opened from mobile browser.
4. The website should be provided with filters that facilitate the search for used items.

B) Security.

The website have been abled to maintain trust between users by following these methods during the establishment of the website:

1- Validates user input.
2- Prevents SQL injection attacks and XSS attacks, this is achieved by angular framework.
3- Every user can provide his personal information to prove his identity to other users.

2.6 use case diagram

User’s scenarios:

Visitor:

As a visitor, I can enter the second hand store website using valid URL, and view the home page where I can view the exhibited items, or search about any item I'm looking for, there are many filters that help me to personalize my search, if I want to contact any owner in order to buy or Inquire about his item, I can enter his profile and contact him via his contact information.

And if I want to upload any used item to offer it in the second hand store, rate any user or report inappropriate item, I must register in the website to have account with full functionality.

User:

As a user, I have two options to use the website; I can enter the second hand store website using valid URL as a visitor without login with my account.

But when I log in with my account, I have more functionality in the website, like add any item to offer it at the website, rate any user or report inappropriate item, edit my profile or items, add any item to my favorites, view my favorites, and at the end, I can logout from my account.

Admin:

As admin, I must log in with my admin account, and review the reports sent by users, then I can do the appropriate action, since I can view all users accounts and items.

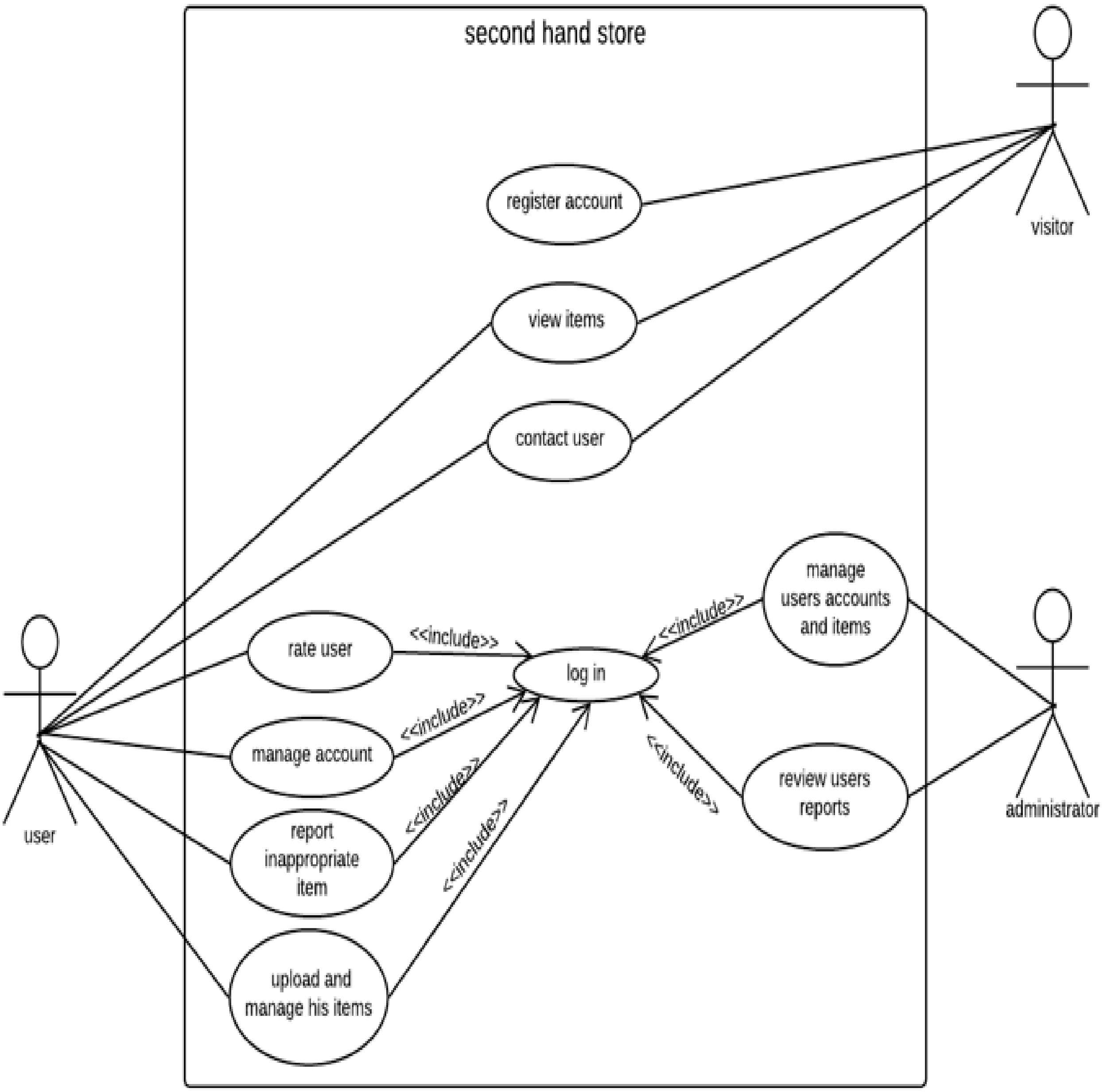


Figure 3: use case diagram

2.7 functional requirements specifications

Functional requirements for visitor:

1. Ability to view/browse all exhibited items using filters

requirement name	view/browse all exhibited items using filters
Main user	Visitor, User
Target	view item that user concerned with
Conditions	- in case of search : Enter item name or select category or any other filters
Procedures	- Login to the main screen of the site. - Browse the exhibits presented by default. If you want to search a specific item: - Click on the item search bar. - Enter the name of the item to be viewed. - Click on the search icon. - Shows all offers related to the searched name. - Or select a category and view all items that fall under that category.
Exceptions	- Item name does not exist. - There is no data yet in the selected filter.
Exceptions solution	- Enter correct and existed item name in the search bar.

2. Ability to create user account.

requirement name	Create user account
Main user	Visitor
Target	Have an account in the website to sell, donate, rent or buy used items
Conditions	-
Procedures	- Login to the main screen of the website. - Click New or Create Account.

	- Enter new user personal informaion. - Click on the Create button.
Exceptions	- Username and email already exists. - The data type is incorrect.
Exceptions solution	- Enter name and email not already used. - Re-enter data correctly.

3. Ability to contact any user.

requirement name	Contact another user
Main user	User, Visitor
Target	To have connection between users.
Conditions	-valid contact information
Procedures	- View item which user interested in. - enter the items detail page. - contact the item owner with the available contact information
Exceptions	Invalid contact information.
Exceptions solution	Search about another item

Functional requirements for user:

4. Ability to log in to his account.

requirement name	Login
Main user	Admin ,user
Target	Enable the user to access permission available to him in the website.
Conditions	To be registered in the website.
Procedures	- open the main screen of the website. - Write your username and password.

	- Click on the login button. - Login to the website and explore the home page.
Exceptions	- The username or password is incorrect. - Inability to create a session between the server and the user's browser
Exceptions solution	- Re-enter the correct data. - Modify browser settings.

5. Ability to rate any user account.

requirement name	Rate another user account.
Main user	User.
Target	Enrich the user's account with people's opinions about his dealings and offers.
Conditions	Has logged in
Procedures	- Click on the user search bar. - enter user name you want to rate. - click search button. - enter the user account. - write comment about him or rate him by stars in people feedback section.
Exceptions	
Exceptions solution	

6. Ability to do CRUD operations on his items.

requirement name	Do CRUD operations on user items.
Main user	User
Target	Create, Read, Update and Delete his items.
Conditions	To be the owner of the item.
Procedures	- Login to his account. - press my items button. - Click add or edit or delete second hand item button related to specific item. - enter the required data in each case.
Exceptions	- The data type is incorrect in case of update or create.
Exceptions solution	- re-enter correct data.

7. Ability to edit his account information.

Requirement name	edit his account information
Main user	User
Target	edit account information when necessary
Conditions	Has logged in
Procedures	-the user goes to the account's profile -he clicks on edit information button -edits information then save
Exceptions	- The data type is incorrect in case of update.
Exceptions solution	- re-enter correct data.

8. Ability to report/complain any inappropriate item.

requirement name	Ability to report/complain any inappropriate item
Main user	User
Target	To give users the ability to report any inappropriate item on the site.
Conditions	
Procedures	-The user clicks on the options icon of the item -then he choose report item option -he can write messages with the report
Exceptions	
Exceptions solution	

9. Ability to log out from his account.

requirement name	Log out
Main user	User, admin
Target	To log out from user account
Conditions	Be logged in
Procedures	-user clicks on the log out button on the naval
Exceptions	The session was not successfully deleted from the browser.
Exceptions solution	Try to log out again

Functional requirements for website administrator:

1. Reviewing reports/complaints about any exhibited item from

website users.

requirement name	reviewing reports/complaints about any exhibited item from website users
Main user	admin
Target	To receive the reports about items, then decide if the items should stay or not.
Conditions	Logged in as admin
Procedures	-admin log in with his account -he clicks on the reports section -then he views the reports due to reporting date.
Exceptions	
Exceptions solution	

2. Ability to control any account or item.

requirement name	Control any account or item.
Main user	Admin
Target	To give the administrator a full control over any account so he can: retrieve and delete any account or item.
Conditions	
Procedures	-admin log in with his account -he clicks on the users section or items section -Administrator searches about the account or item -he clicks on it -he view or delete the account or item.
Exceptions	The account was not successfully deleted.
Exceptions solution	Try to re-update or re-delete the account.

2.8 class diagram

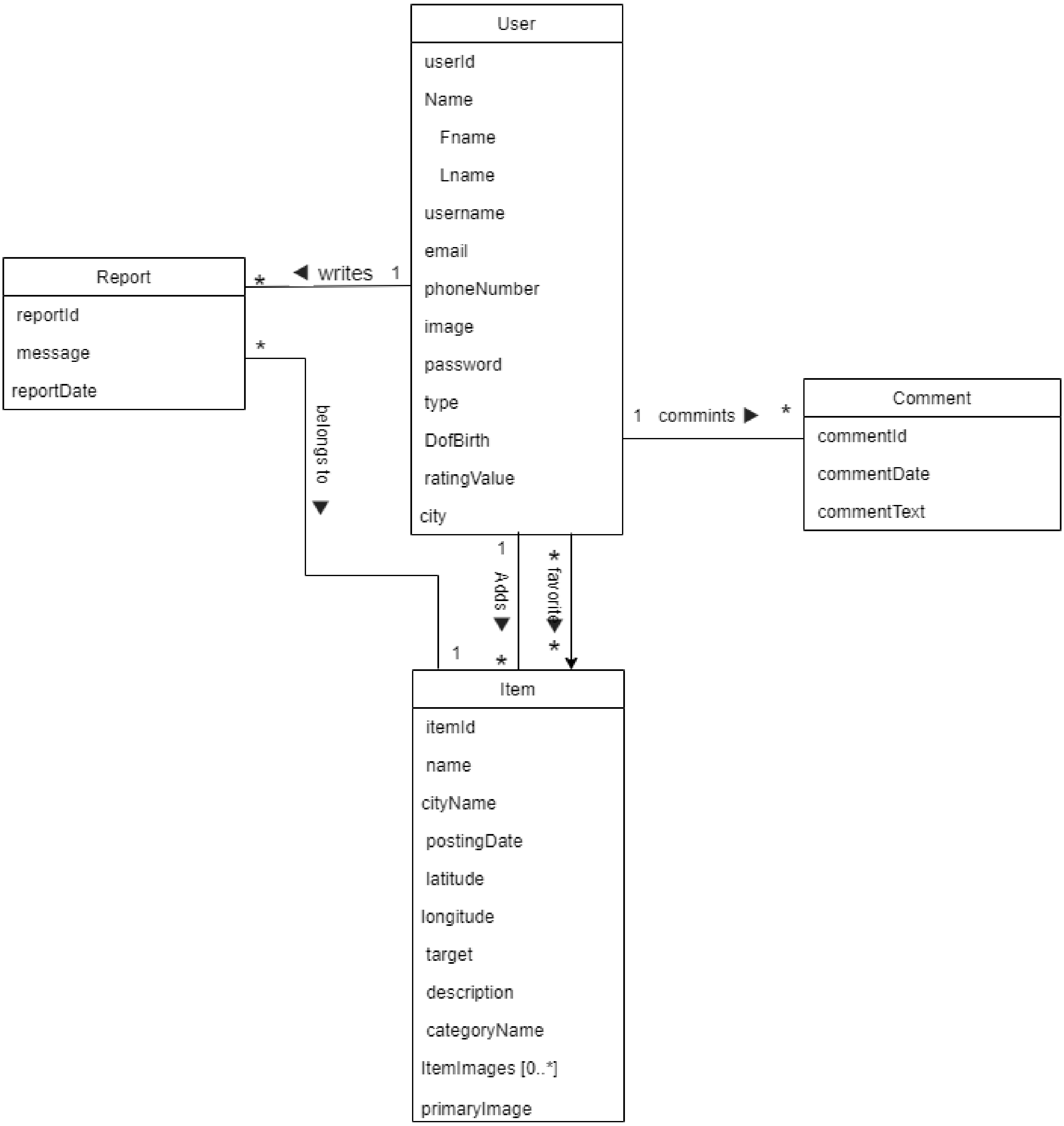


Figure 4: class diagram

Chapter 3: Software Design.

3.1 Introduction.

In this chapter, the block diagram, the website architecture, the website database description, the database table's description, as well as the graphical interfaces will be explained.

3.2 block diagram.

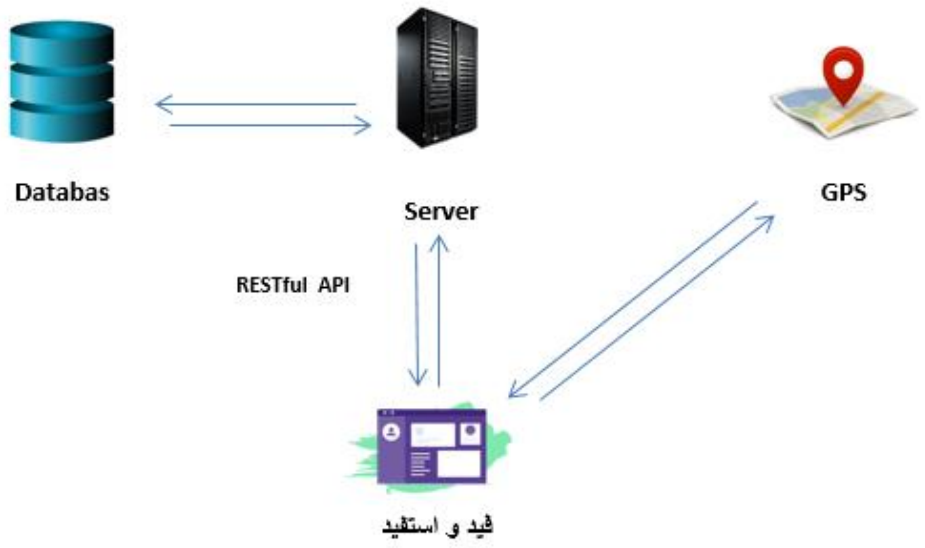


Figure 5:block diagram

3.3 design decision.

A) Architectural pattern: architectural pattern is a description of good design practice, which has been tried and tested in different environments, like MVC pattern, layered pattern, we chose MVC pattern as our architectural design, because this pattern allows the data to change independently of its representation and vice versa and supports presentation of the same data in different ways with changes made in one representation shown in all of them.

B) Backend: we built the backend as RESTful APIs, since we have a website today, but we may add a mobile app tomorrow. We may need to change databases, change technology stacks, and deal with total rewrites of our user interface as our website grows.

This is why decoupling the front-end from the back-end with a RESTful APIs is so valuable. It makes these major architectural shifts possible, since data between the front-end and back-end will be in JSON format and the view will be rendered at the client-side.

3.4 Website Architecture.

The architecture of MVC (Model, View, Controller) has been adopted. Since back-end is the Model (RESTfull APIs), and front-end is the view and controller, as shown in the following figure:

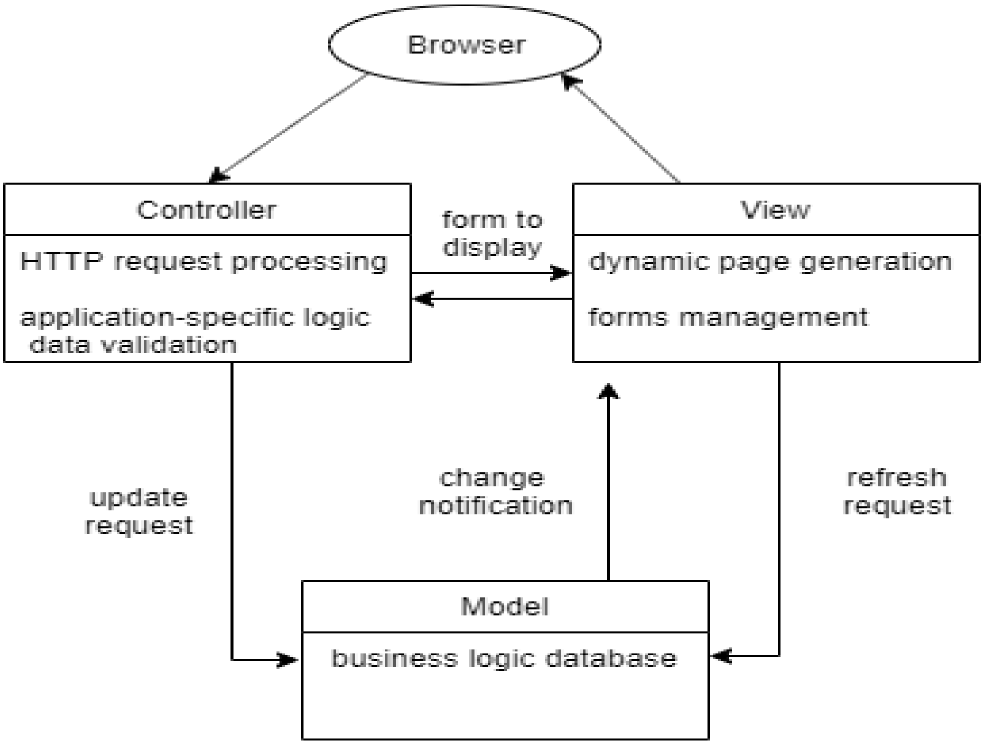


Figure 6:MVC architecture

1. Model: It works to manage data and operations related to the website databases, in second hand store website, this is the normalized database:

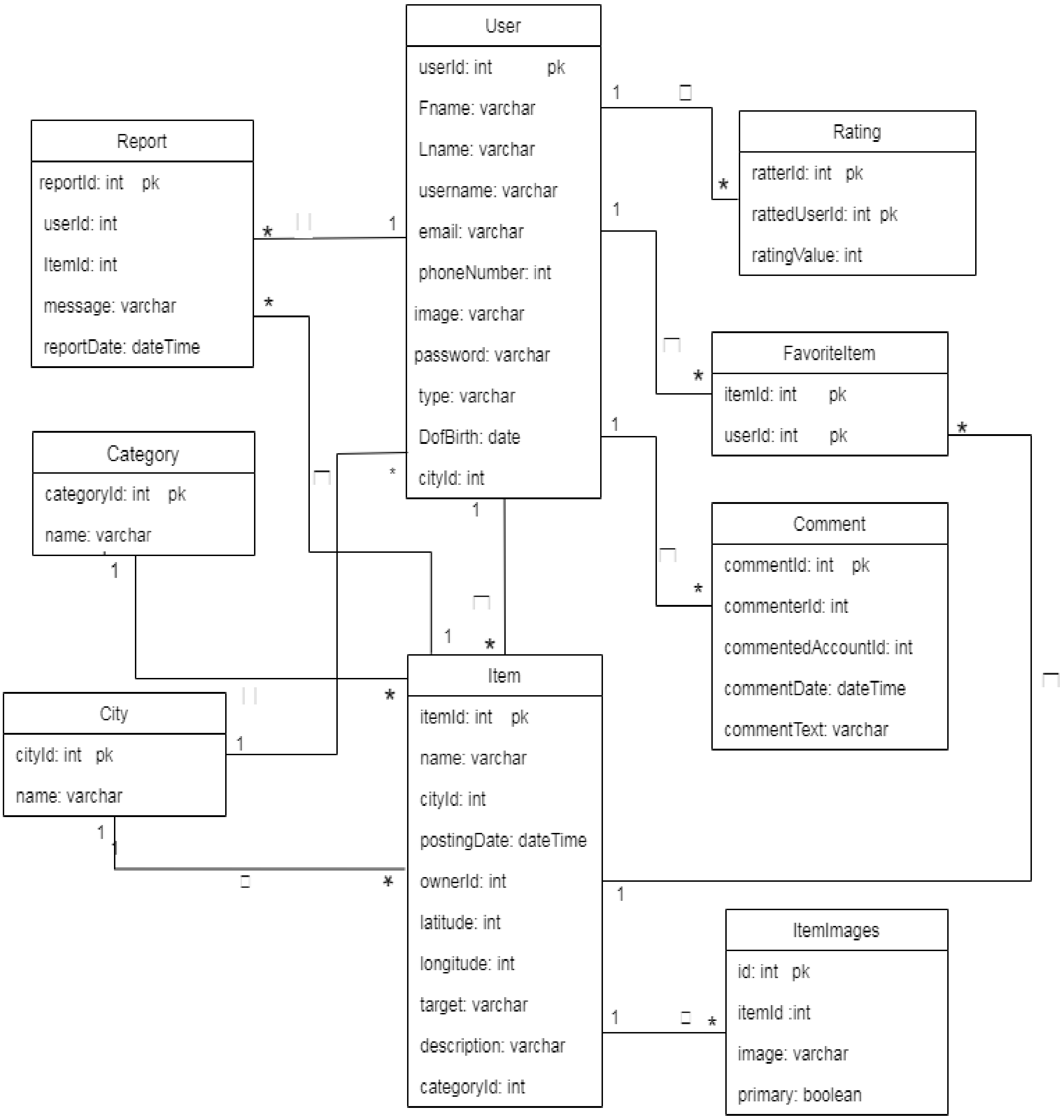


Figure 7 : database mapping

Table 4: database tables

The name of the table in the database	Table description
User	Store data related to users and user type
Item	Store data related to second hand items that users add
Rating	Store the rating value that users rate
FavoriteItem	Store favorite items for the users
Comment	Store users comments on another users profile
Report	Store user complaints/reports about inappropriate items
Category	Store the categories that items belong to
City	Store the cities that items placed in
ItemImages	Store item’s images

Database tables description:

Table 5: user

Field name	**Data type**	**Size**	**Null**	**Unique**	**Description**
userId	int	10	No	Yes	PK
Fname	varchar	40	No	No	
Lname	varchar	40	No	No	
usrname	varchar	40	No	Yes	
Email	varchar	100	Yes	Yes	
PhoneNumber	int	10	Yes	Yes	
image	varchar	100	Yes	No	
Password	varchar	16	No	No	
type	bit	1	No	No	
DOfBirth	Date	-	Yes	No	
city	varchar	40	Yes	No	

Table 6: item

Field name	Data type	Size	Null	Unique	Description
itemId	int	10	No	Yes	PK
ownerId	int	10	No	No	FK
name	varchar	40	No	No	
cityId	int	10	No	No	FK
postingDate	dateTime	-	No	No	
price	int	10	Yes	No	
categoryId	int	10	No	No	FK
lattitude	Varchar	-	Yes	No	
longitude	varchar	-	Yes	No	
target	varchar	10	No	No	
description	varchar	500	Yes	No	

Table 7: favoriteItem

Field name	Data type	Size	Null	Unique	Description
itemId	int	10	No	No	PK
userId	int	10	No	No	PK

Table 8: rating

Field name	Data type	Size	Null	Unique	Description
raterId	int	10	No	No	PK
ratedUserId	int	10	No	No	PK
RatingValue	int	1	No	No	

Table 9: category

Field name	Data type	Size	Null	Unique	Description
categoryId	int	10	No	Yes	PK
name	varchar	40	No	Yes	

Table 10: city

Field name	Data type	Size	Null	Unique	Description
cityId	int	10	No	Yes	PK
name	varchar	20	No	Yes	

Table 11: report

Field name	Data type	Size	Null	Unique	Description
reportId	int	10	No	Yes	PK
reporterId	int	10	No	No	FK
itemId	int	10	No	No	FK
message	varchar	255	Yes	No	
reportDate	dateTime	-	No	No	

Table 12: itemImages

Field name	Data type	Size	Null	Unique	Description
id	int	10	No	Yes	PK
itemId	int	10	No	No	FK
image	varchar	100	No	Yes	
major	boolean	1	No	No	

Table 13: comment

Field name	Data type	Size	Null	Unique	Description
commentId	int	10	No	Yes	PK
commenterId	int	10	No	Yes	FK
commentedAccountId	int	10	No	No	FK
commentDate	dateTime	-	No	Yes	
text	varchar	500	No	No	

RESTful APIs:

Here is the preliminarylist of RESTful APIs the website need:

1. GET /items => returns all items information as JSON array.
2. GET /items/:id => returns specific item information using the id
3. POST /items => Create new item
4. DELETE /item/:id => DELETE an item
5. PUT /item/:id => update an item information
6. GET / items/:id/itemsImages => returns specific item images using the id
7. GET /categories => returns all categories information as JSON array
8. GET / cities => returns all cities information as JSON array
9. GET /users => return all users
10. POST /users => Create new user
11. GET /users/:id => return specified user
12. PUT /users/:id =>edit specific user account
13. DELETE /users/:id => Delete Specified user
14. POST /users/login => login user
15. POST /users/logout => login user
16. POST /:favid => add fav item to specified user fav items
17. GET /fav => returns all fav items for specified user
18. DELETE /:favid => Delete specified fav item for specified user

19. POST /:accountid/rating=> add rate for specified user
20. GET /:accountid /rating => return rating of specified user
21. POST /:accountid/comment => add comment on specified user profile
22. GET /:accountid /comments => return all comments for specified user
23. POST /report => add report about specific item
24. GET /reports => return all reports

2. Controller:

Manages user interaction and passes these interactions to the View and the Model, which is HTTP request work as the moderator between the operations performed by the user and the website.

3. View.

Defines and manages how the data is presented to the user.

Here are some interfaces for Second Hand Store:

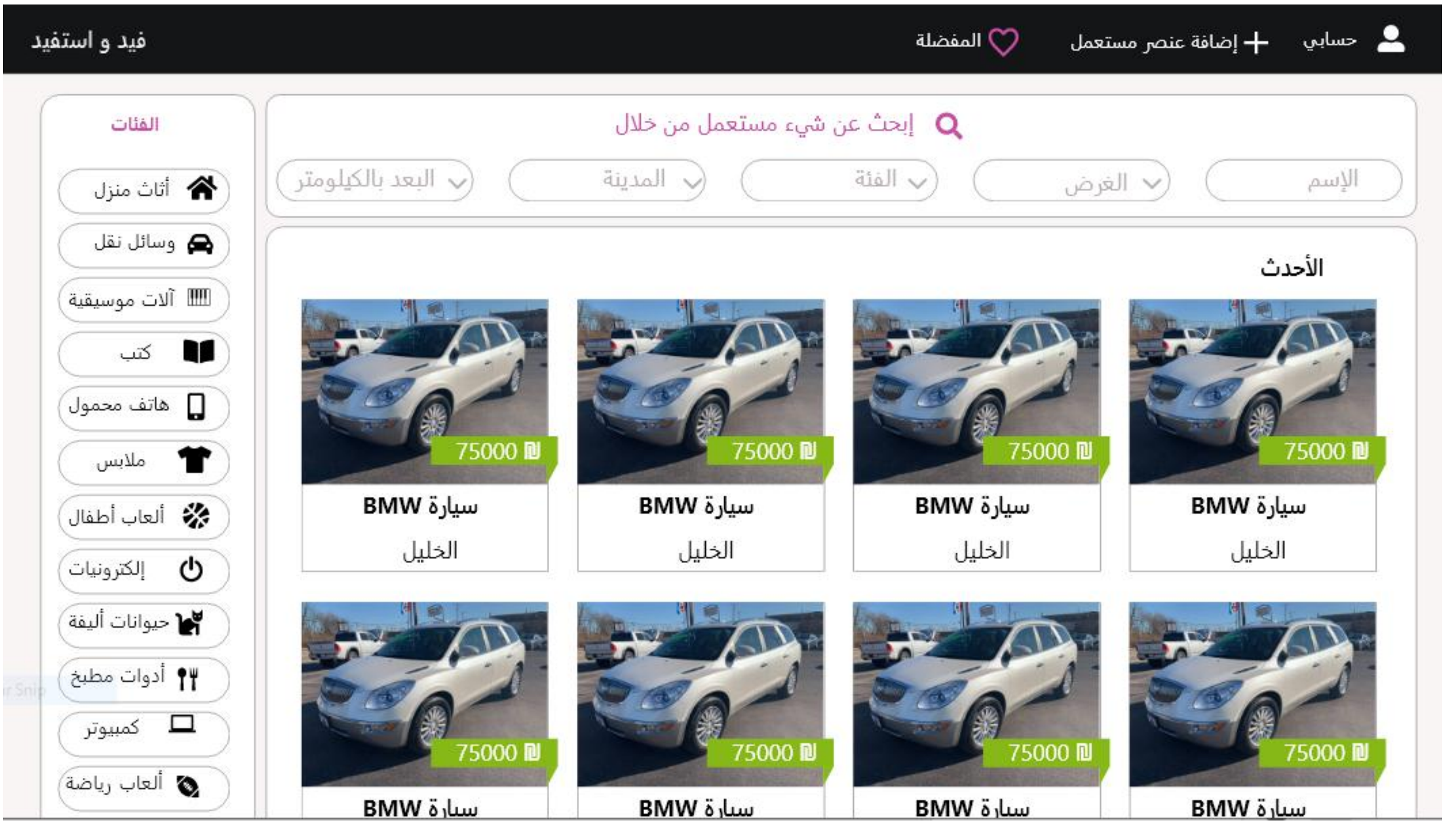
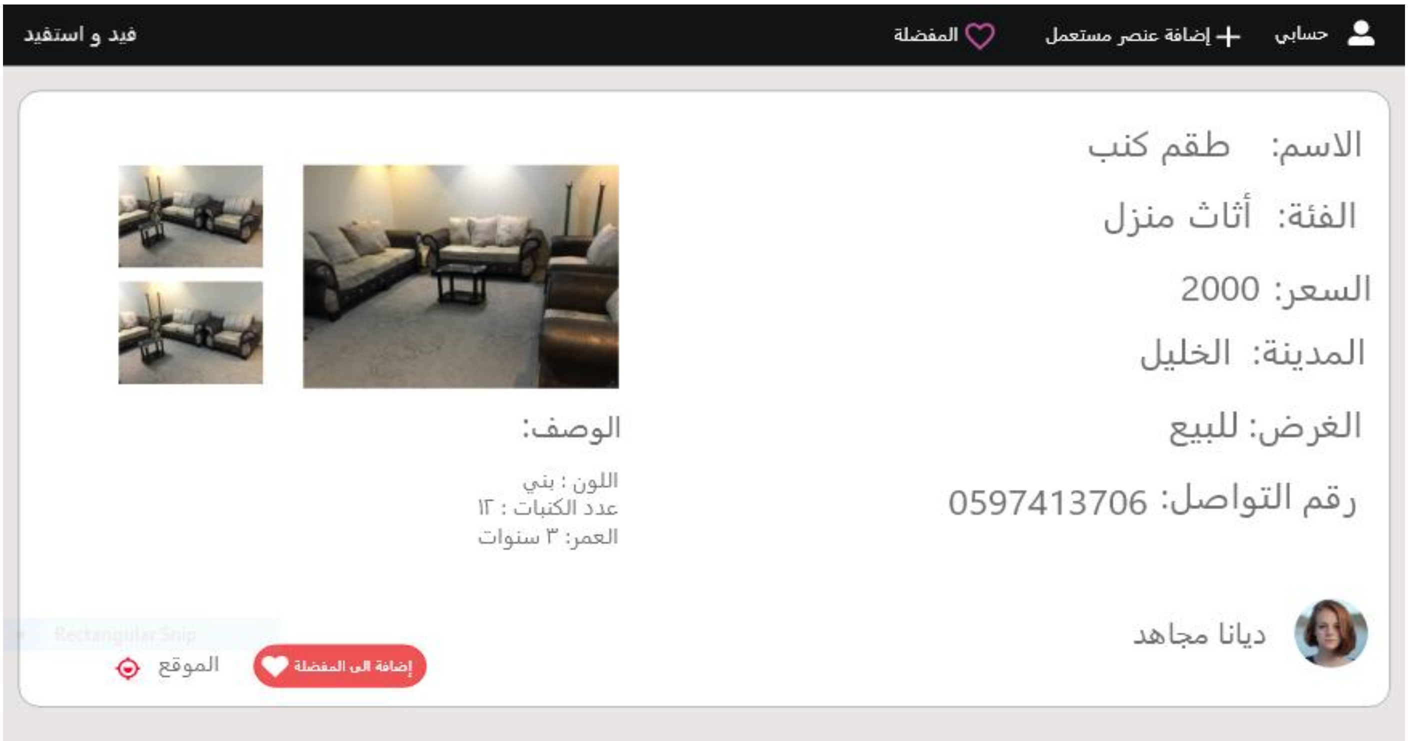


Figure 8: home page



الاسم: طقم كنب

الفئة: أثاث منزل

السعر: 2000

المدينة: الخليل

الغرض: للبيع

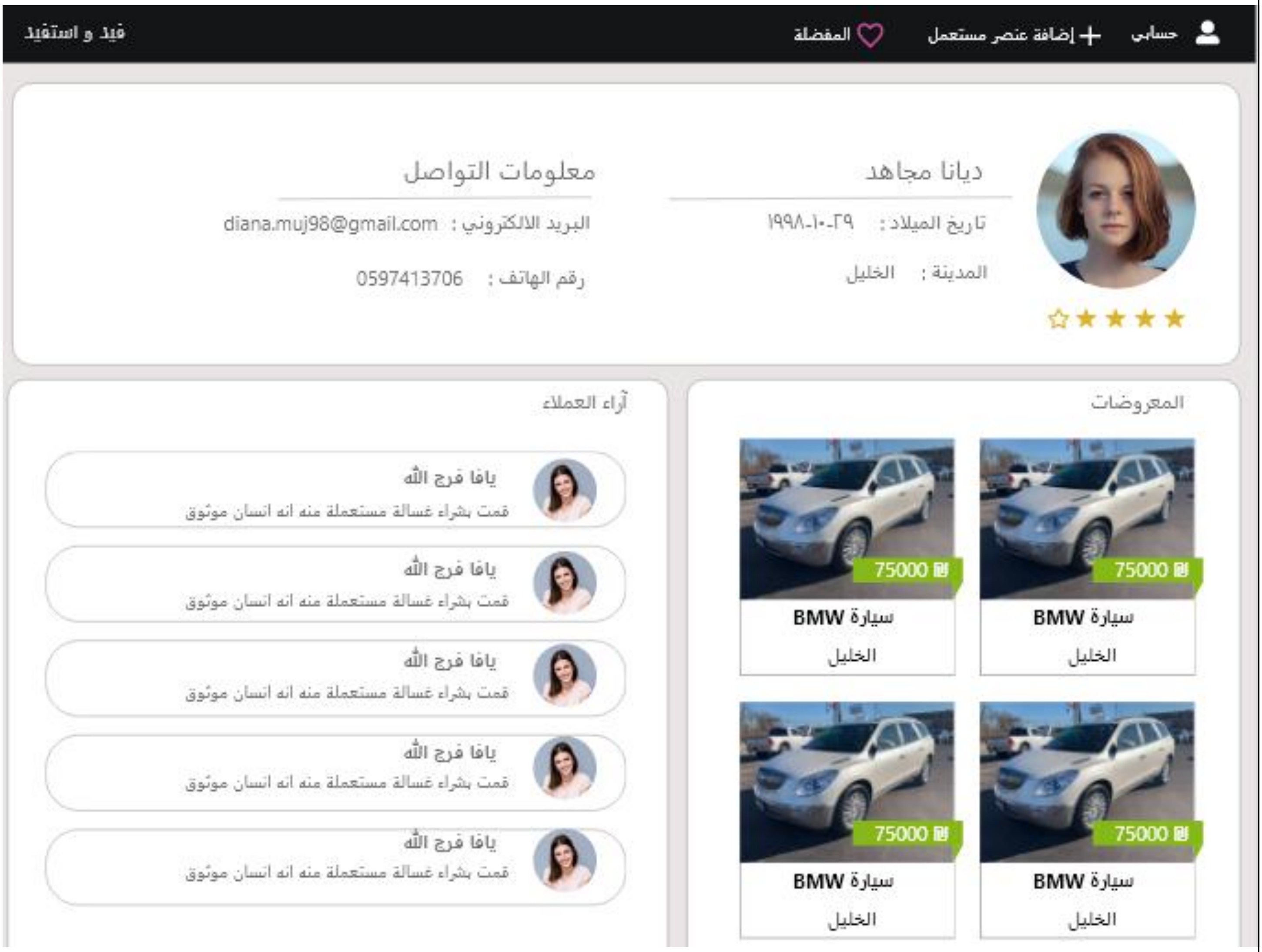
رقم التواصل: 0597413706

الوصف:

اللون : بني
عدد الكنبات : ١٢
العمر: ٣ سنوات

ديانا مجاهد

Figure 9: item details page



ديانا مجاهد

تاريخ الميلاد : ١٩٩٨-١٠-٢٩

المدينة : الخليل

معلومات التواصل

البريد الالكتروني : diana.muj98@gmail.com

رقم الهاتف : 0597413706

المعروضات

75000 ₪ سيارة BMW الخليل

75000 ₪ سيارة BMW الخليل

75000 ₪ سيارة BMW الخليل

75000 ₪ سيارة BMW الخليل

آراء العملاء

يافا فرج الله
قمت بشراء غسالة مستعملة منه انه انسان موثوق

يافا فرج الله
قمت بشراء غسالة مستعملة منه انه انسان موثوق

يافا فرج الله
قمت بشراء غسالة مستعملة منه انه انسان موثوق

يافا فرج الله
قمت بشراء غسالة مستعملة منه انه انسان موثوق

يافا فرج الله
قمت بشراء غسالة مستعملة منه انه انسان موثوق

Figure 10: user details page

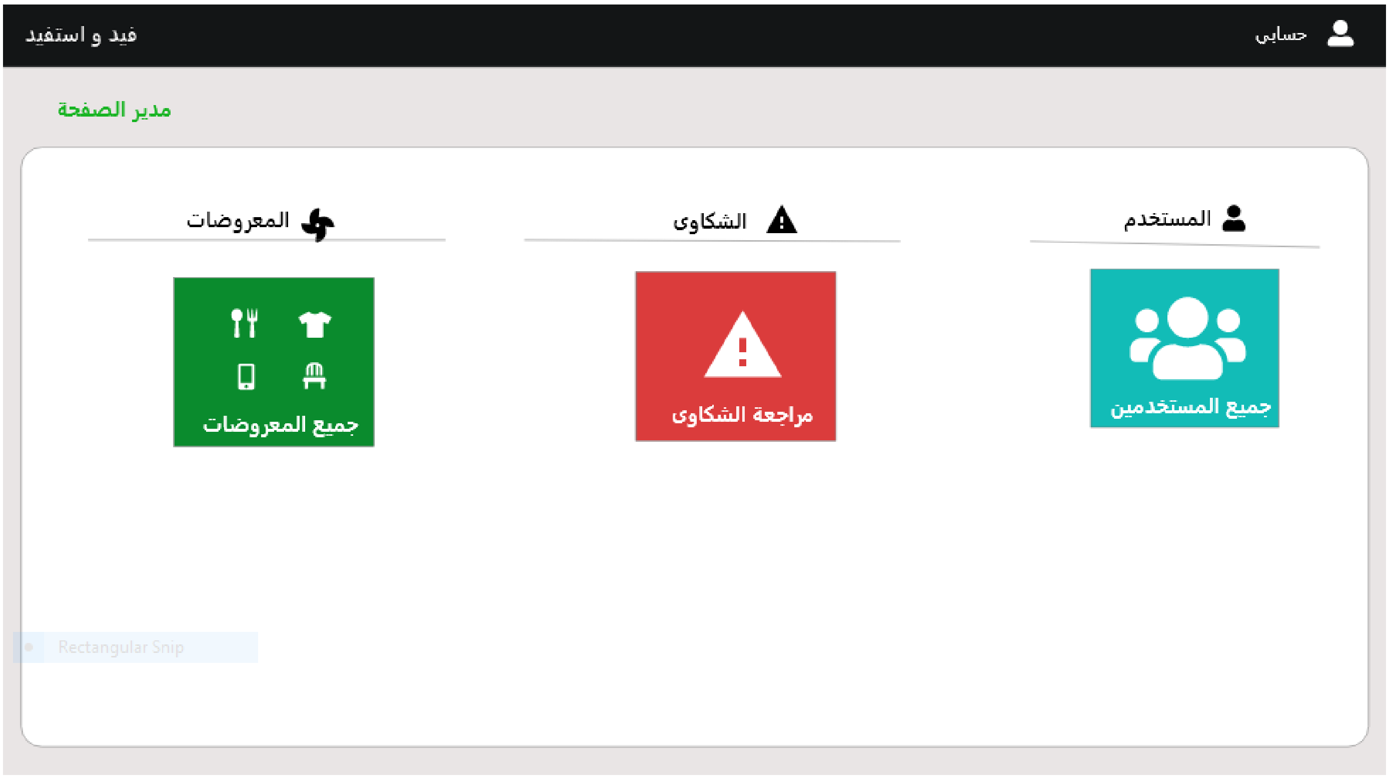


Figure 11: admin home page

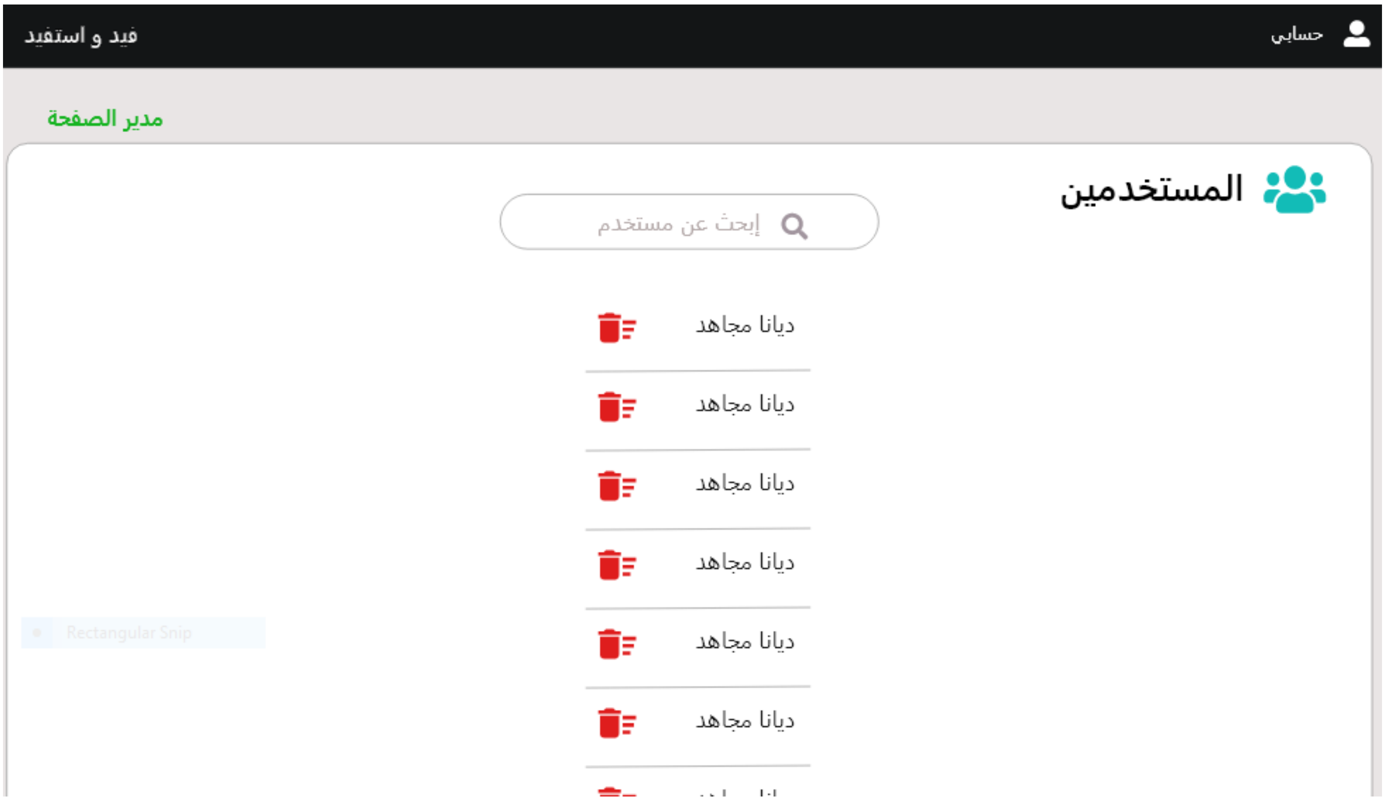
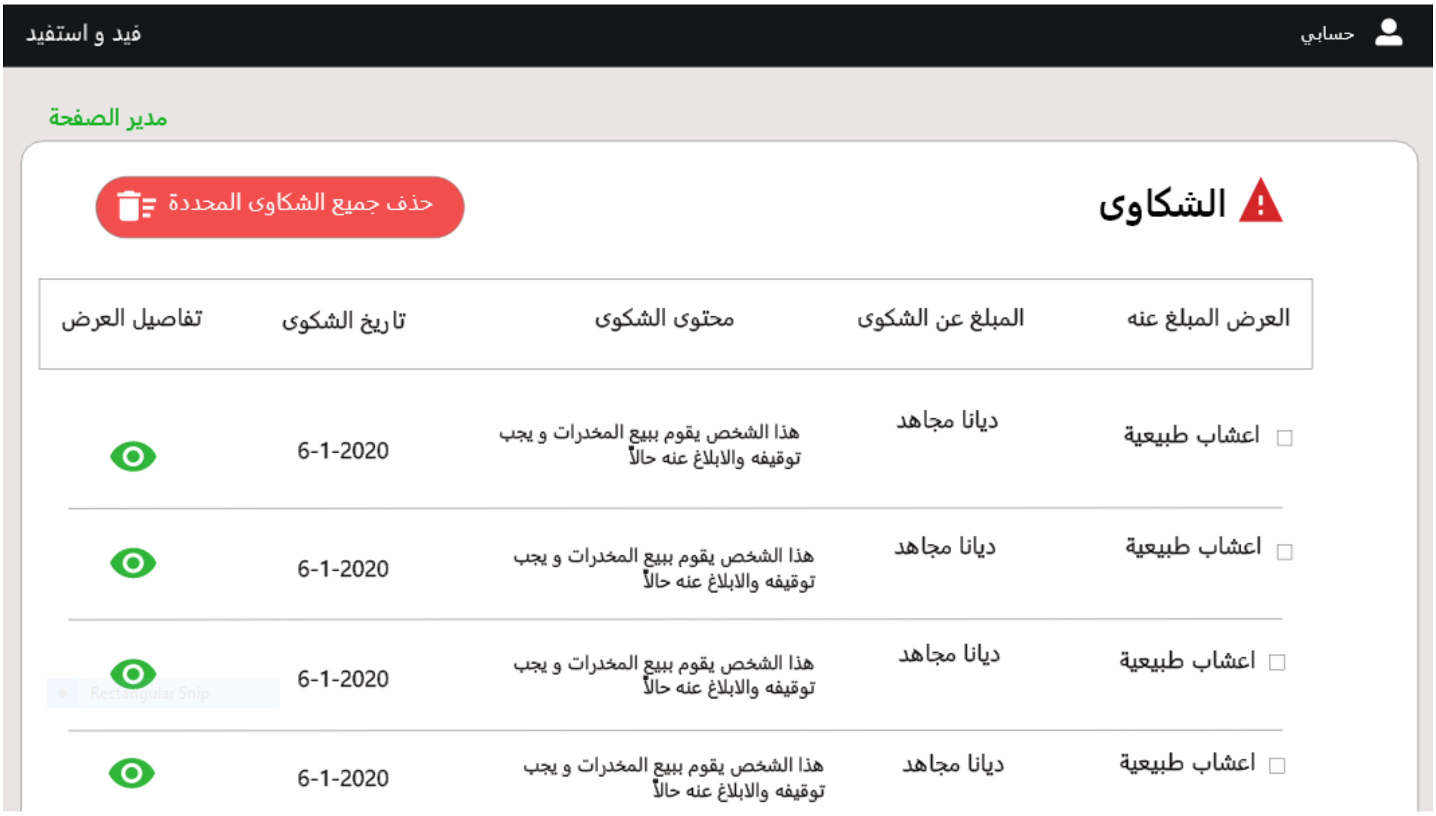


Figure 12: all users page



الشكاوى

العرض المبلغ عنه	المبلغ عن الشكوى	محتوى الشكوى	تاريخ الشكوى	تفاصيل العرض
□ اعشاب طبيعية	ديانا مجاهد	هذا الشخص يقوم ببيع المخدرات و يجب توقيفه والابلاغ عنه حالاً	6-1-2020	👁
□ اعشاب طبيعية	ديانا مجاهد	هذا الشخص يقوم ببيع المخدرات و يجب توقيفه والابلاغ عنه حالاً	6-1-2020	👁
□ اعشاب طبيعية	ديانا مجاهد	هذا الشخص يقوم ببيع المخدرات و يجب توقيفه والابلاغ عنه حالاً	6-1-2020	👁
□ اعشاب طبيعية	ديانا مجاهد	هذا الشخص يقوم ببيع المخدرات و يجب توقيفه والابلاغ عنه حالاً	6-1-2020	👁

Figure 13: reports page

Chapter ٤: implementation.

4.1 introduction.

In this chapter, the technologies that used in building the website will be discussed.

4.2 technology used and why .

- For back-end:

1. Nodejs : is an open-source, cross-platform, JavaScript runtime environment that executes JavaScript code outside of a browser. Node.js lets developers use JavaScript to write command line tools and for server-side scripting—running scripts server-side to produce dynamic web page content before the page is sent to the user's web browser.

2. Express : Express.js, or simply Express, is a web application framework for Node.js, released as free and open-source software under the MIT License. It is designed for building web applications and RESTFull APIs. It has been called the de facto standard server framework for Node.js.

3. Typescript : TypeScript is an open-source programming language developed and maintained by Microsoft. It is designed for development of large applications and transcompiles to JavaScript. As TypeScript is a superset of JavaScript, existing JavaScript programs are also valid TypeScript programs. TypeScript may be used to develop JavaScript applications for both client-side and server-side execution (as with Node.js).

4. MySql : is an open-source relational database management system (RDBMS).

5. Postman: its APIs test tools.

6. Visual studio code : it is a source-code editor developed by Microsoft for Windows Linux and macOS. It includes support for debugging, embedded Git control and GitHub, syntax highlighting, intelligent code completion, snippets, and code refactoring.

- For front-end

Angular framework : A JavaScript based open-source Framework to build client-side applications using HTML, CSS, JavaScript/TypeScript.

Great for building Single Page Application (SPA).

Why SPA ?

- Single-page application (or SPA) are applications that are accessed via a web browser like other websites but offer more dynamic interactions.
- It resembles native mobile and desktop apps.
- The most notable difference between a regular website and SPA is the reduced amount of page refreshes.

- SPAs have a heavier usage of AJAX -
- The process of rendering pages happens mostly on the client-side

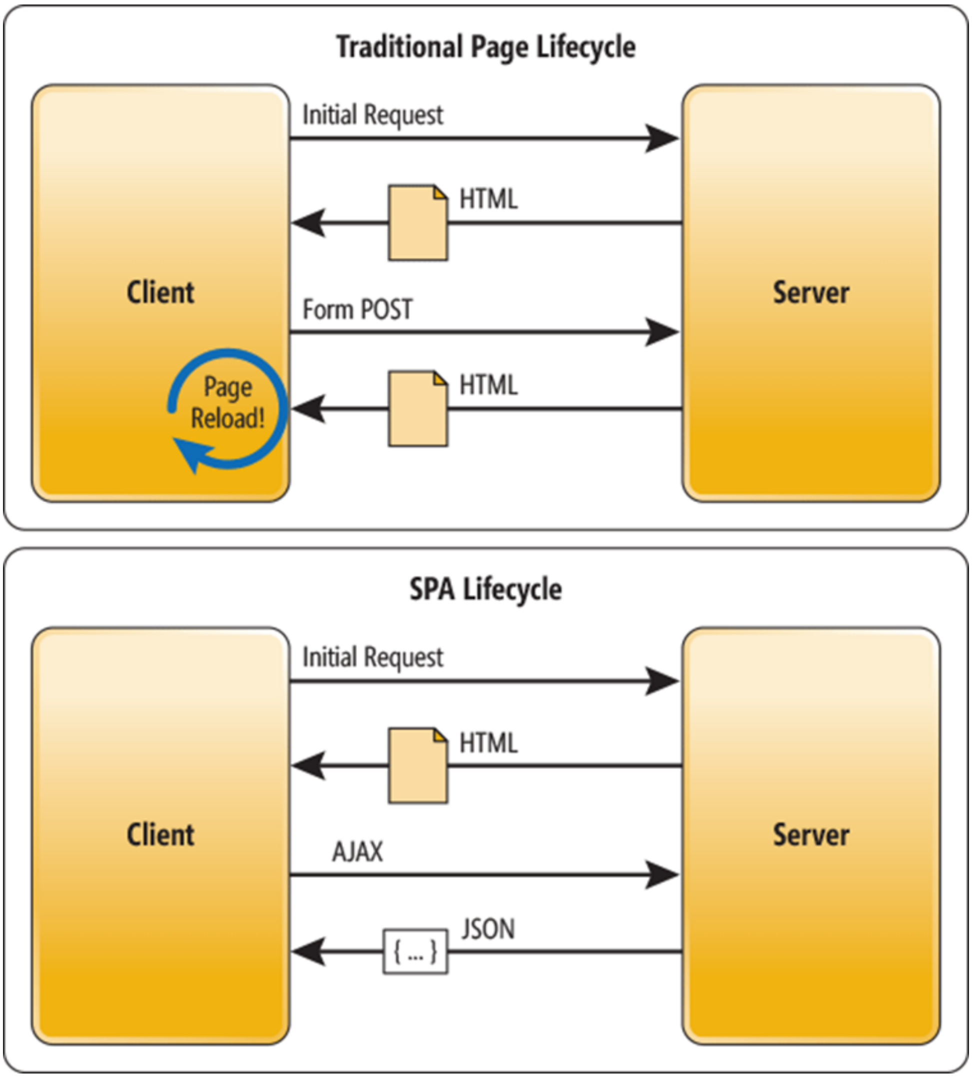


Figure 14: SPA

4.3 system diagram.

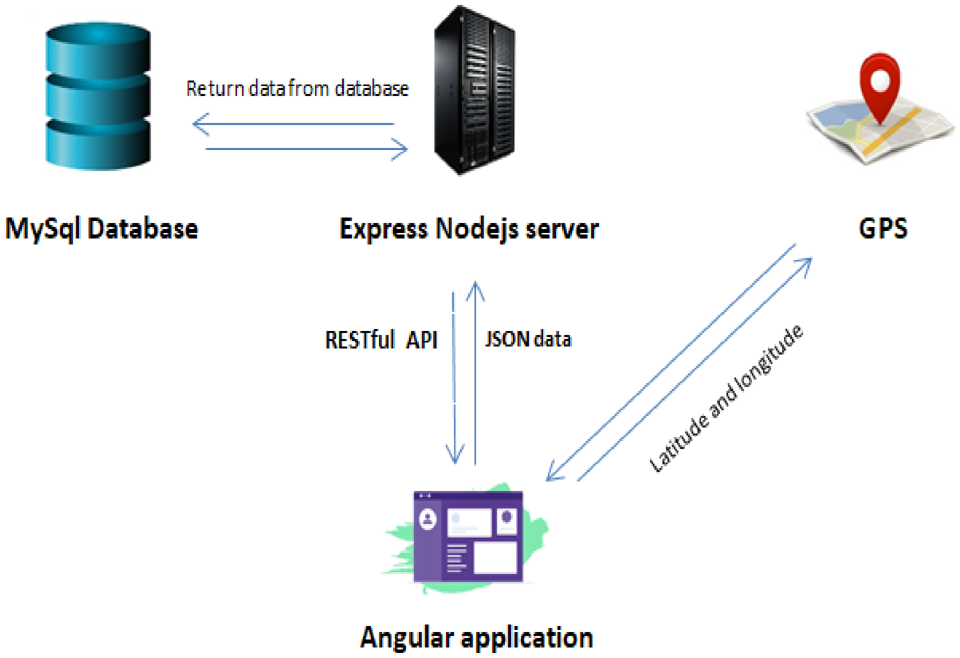


Figure ١٥: system diagram

Chapter 5 : Testing.

5.1 introduction.

In this chapter, unit testing for backend APIs and functional requirements testing will be discussed.

When building software, we often forget the importance of testing. Testing not only ensures your application or system is working as expected, but it also helps manage new changes in specification or implementation.

There are different types of testing for different test approaches:

- Functional testing
- Unit testing

5.2 Functional requirements testing .

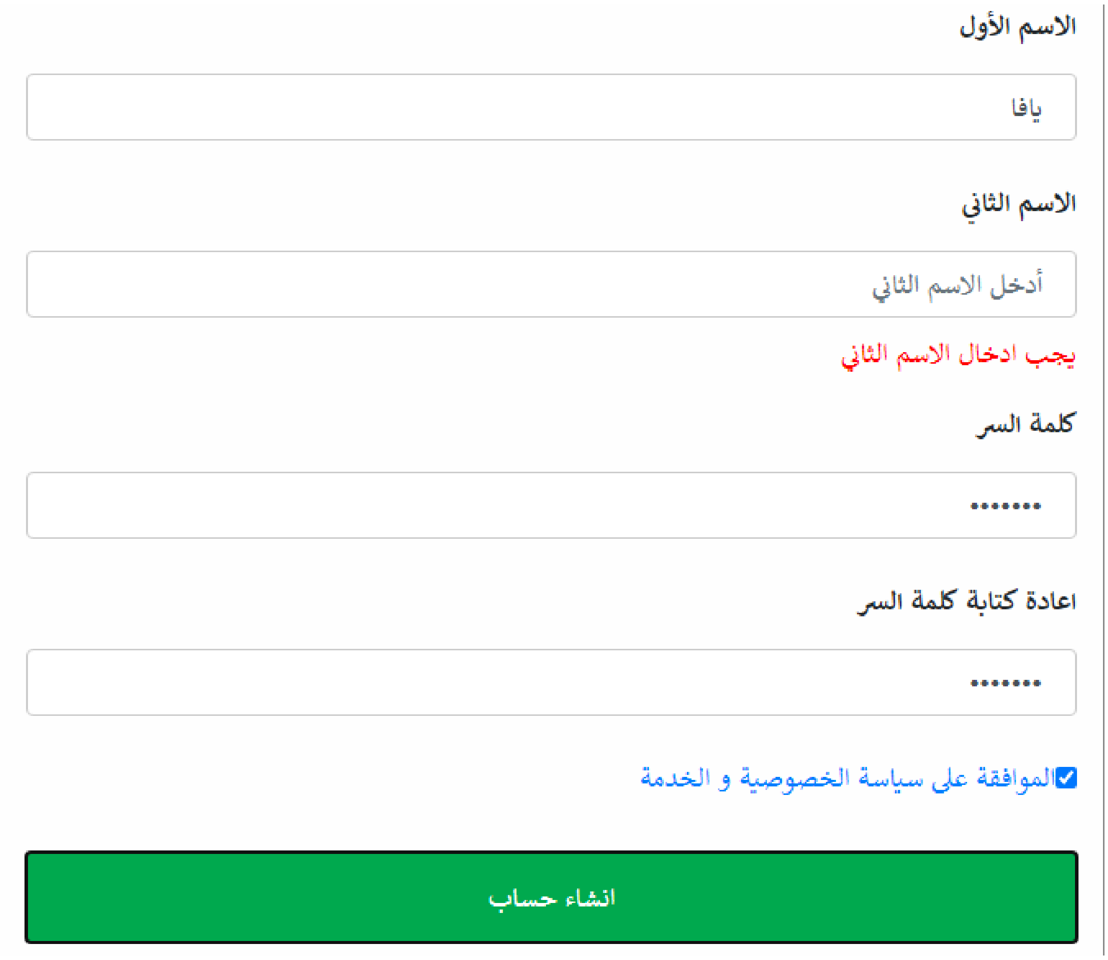
Functional testing is nessessary to know whether the system is working as it should be ,since there’s a mechanism in place to check if the system works well, this mechanism is to check the functional requirements of the project if it’s applied or not.

Table١٤ : functional testing

Test case	Senario	Input data	Expected Output	Actual Output	Test result
Create user account	1.user enter all required fields correctly	اسم المستخدم : يافا الاسم الاول :يافا الاسم الثاني :فرج الله كلمة السر :٢٦٣١٩٩٨ اعادة كتابة كلمة السر :٢٦٣١٩٩٨	New user account created in database	New user account created in database	pass
	2. user enter an incomplete information	اسم المستخدم : يافا الاسم الاول :يافا الاسم الثاني : كلمة السر :٢٦٣١٩٩٨ اعادة كتابة كلمة السر :٢٦٣١٩٩٨	Red warning messages appeared in the blank fields, see Figure 16	Red warning messages appeared in the blank fields	Pass

	3. user enter two passwords don't match each other	اسم المستخدم : يافا الاسم الاول :يافا الاسم الثاني :فرج الله كلمة السر ٢٦٣١٩٩٨: اعادة كتابة كلمة السر :٨٥٢٦٣١٩	Red warning message "كلمة السر غير متطابقة" will appear,	Red warning message "كلمة السر غير متطابقة" will appear	Pass
View/browse all exhibited items using filters	User want to search for specific item using city filter	اسم السلعة: مروحة المدينة : الخليل	جميع المراوح المعروضة في مدينة الخليل سوف تظهر في الصفحة الرئيسية	جميع المراوح المعروضة في مدينة الخليل سوف تظهر في الصفحة الرئيسية	Pass
Log in	1. User enter his username and password correctly	اسم المستخدم : يافا كلمة السر: ٢٦٣١٩٩٨	Token provided from database and stored in local storage and user logged in his account successfully	Token provided from database and stored in local storage and user logged in his account successfully	pass
	2. User enter his username with wrong password	اسم المستخدم : يافا كلمة السر: ٨7٢٦٣١٩	Red warning message "كلمة السر غير صحيحة " appeared	Red warning message "كلمة السر غير صحيحة " appeared	Pass
rate any user account	User navigate to another user account to rate him	User picks 4 stars	the new value 4 stored in database and rating for rated user is recalculated	the new value 4 stored in database and rating for rated user is recalculated	Pass
report/complain any inappropriate item	User navigate to item details page and click on report button	User enter report text in report form	Report information stored in database and confirm massage appeared, see figuare 17	Report information stored in database and confirm massage appeared	Pass
log out	User clicks on logout button	Click on logout button	Token removed from local storage and	Token removed from local storage and	Pass

			user loged out from his account	user loged out from his account	
Admin can reviewing reports/complaints about any exhibited item from website users	Admin clicks on reports icon	Click on reports icon	All reports displayed from database in a table, see figuare 18	All reports displayed from database in a table	Pass
Admin can control any account	Admin clicks on all users icon	Click on all users icon	All users stored in database displayed with ability to delete any user, see figuare 19	All users stored in database displayed with ability to delete any user	Pass
User can edit his account information	1.User choose to change his password	New password entered	Password updated successfully in database	Password updated successfully in database	Pass
	2.User choose to change his profile image	New image selected	The new image uploaded successfully in database and changed directly in the profile	The new image uploaded successfully in database and changed directly in the profile	
	3.User choose to edit his personal information	User change some fields in edit profile form	The edited information stored in database and appeared in the profile info section	The edited information stored in database and appeared in the profile info section	Pass
User can do CRUD operations on his items	1. User wants to add new item to the website	User filled add item form with the needed data	Item added successfully in the database and viewed with other items	Item added successfully in the database and viewed with other items	Pass
	2. User wants to delete his item from the website	User clicks on delete button of specific item	Item and all related things to it deleted from the database	Item and all related things to it deleted from the database	Pass



الاسم الأول

يافا

الاسم الثاني

أدخل الاسم الثاني

يجب ادخال الاسم الثاني

كلمة السر

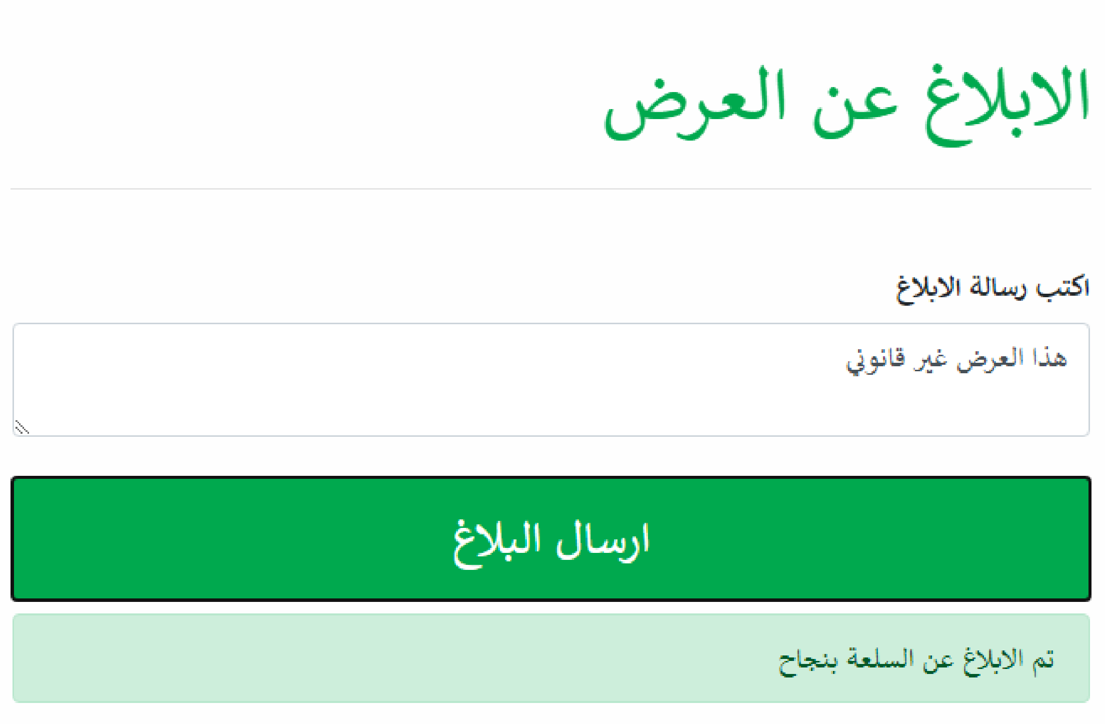
•••••••

اعادة كتابة كلمة السر

•••••••

☑الموافقة على سياسة الخصوصية و الخدمة

Figure١٦



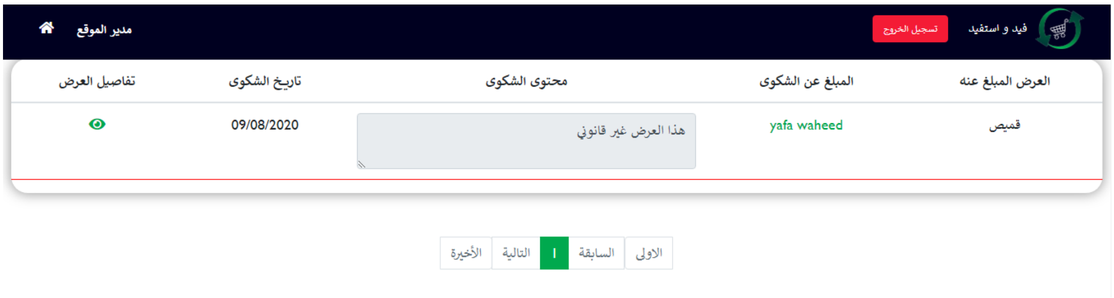
الابلاغ عن العرض

اكتب رسالة الابلاغ

هذا العرض غير قانوني

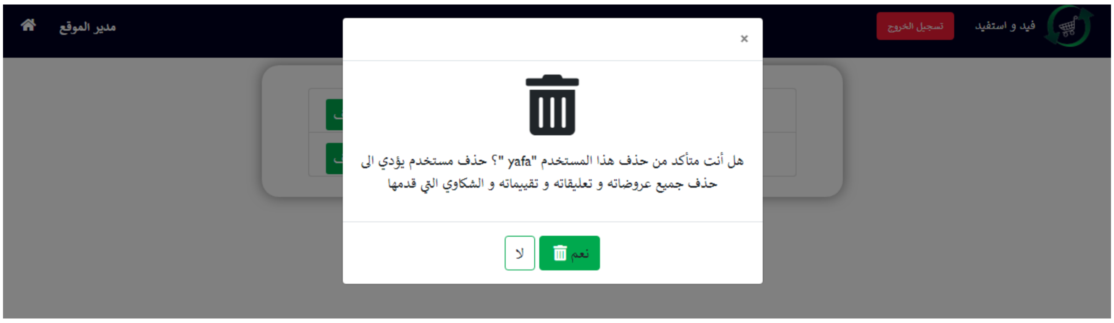
تم الابلاغ عن السلعة بنجاح

Figure 17



العرض المبلغ عنه	المبلغ عن الشكوى	محتوى الشكوى	تاريخ الشكوى	تفاصيل العرض
قميص	yafa waheed	هذا العرض غير قانوني	09/08/2020	👁

Figure 18



هل أنت متأكد من حذف هذا المستخدم "yafa" ؟ حذف مستخدم يؤدي الى
حذف جميع عروضاته و تعليقاته و تقييماته و الشكاوي التي قدمها

Figure 19

5.3 Unit testing for APIs .

The most popular and important testing types is unit testing. Unit testing is basically testing if a unit or component of the system is working as expected. You either just call the component, if no input is required, or give it an input and determine the output.

In the context of REST API, a unit is a single endpoint request, and writing a unit test for this particular API depends on what you want to test in its response base on the request sent.

We test our APIs using postman testing tool.

1. GET user/items

- Expected output: json object contains first 8 items on the website with json pager object.
- Actual output :

```
{
    "pager": {
        "totalItems": 4,
        "currentPage": 1,
        "pageSize": 8,
        "totalPages": 1,
        "startPage": 1,
        "endPage": 1,
        "startIndex": 0,
        "endIndex": 3,
        "pages": [
            1
        ]
    },
    "pageOfItems": [
        {
            "itemid": 49,
            "ownerid": 30,
            "name": "سيارة",
            "cityid": 2,
            "postingdate": "2020-08-08T13:36:17.000Z",
            "price": 2000,
            "categoryid": 6,
            "lattitude": "31.534841399999998",
            "longitude": "35.077861299999995",
            "target": "للبيع",
            "description": "",
            "cdescription": ""
        },
        {
            "itemid": 48,
            "ownerid": 30,
            "name": "سجادة",
            "cityid": 1,
            "postingdate": "2020-08-08T13:35:30.000Z",
            "price": 100,
            "categoryid": 4,
            "lattitude": "31.534841399999998",
            "longitude": "35.077861299999995",
            "target": "للتبرع",
            "description": "",
            "cdescription": ""
        },
        {
            "itemid": 47,
            "ownerid": 30,
            "name": "سنسال",
            "cityid": 1,
            "postingdate": "2020-08-08T13:22:49.000Z",
            "price": 0,
            "categoryid": 5,
            "lattitude": "31.534841399999998",
            "longitude": "35.077861299999995",
            "target": "للتبرع",
            "description": "",
            "cdescription": ""
        }
    ]
}
```

- Passed/failed : **Passed**

2. POST user/createuser

- Body Input:

```
{
     "user":{

     "fname":"صفاء",
     "lname":"مجاهد",
     "username":"صفاء",
     "password":"صفاء"

     }
}
```

- Expected output:

```
{
    "registered": true
}
```

- Actual output:

```
{
    "registered": true
}
```

- Passed/failed : **Passed**

3. GET   user/anyitem/:itemid

- Input: user/anyitem/49
- Expected output: details about the itemid 49
- Actual output:

```
[
    {
        "itemid": 49,
        "ownerid": 30,
        "name": "سيارة",
        "cityid": 2,
        "postingdate": "2020-08-08T13:36:17.000Z",
        "price": 2000,
        "categoryid": 6,
        "lattitude": "31.534841399999998",
        "longitude": "35.077861299999995",
        "target": "للبيع",
        "description": "",
        "cdescription": ""
    }
]
```

- Passed/failed : **Passed**

4. POST user/ login

- Body Input:

```
{
     "user":{
     "username":"صفاء",
     "password":"صفاء"
      }
}
```

- Expected output: json object contains user information and token.
- Actual output:

```
{
     "user":{
     "username":"صفاء",
     "password":"صفاء"
      }
}
```

- Passed/failed : **Passed**

5. POST user/search

- Body Input:

```
{
    "searchfor":{

        "name":"سجادة",
        "target":"للتبرع",
        "cityid":1,
        "categoryid":4,
        "killometers":2
    },
     "status": {
        "name":true,
        "target":true,
        "cityid":true,
        "categoryid":true,
        "killometers":true
      }
      ,
      "cords": {
        "latitude":31.530174337269706,
        "longitude":35.07542615102615

      }
}
```

- Expected output: json object contains the array of items that matched the search with json pager object.

- Actual output:

```
{
    "pager": {
        "totalItems": 1,
        "currentPage": 1,
        "pageSize": 8,
        "totalPages": 1,
        "startPage": 1,
        "endPage": 1,
        "startIndex": 0,
        "endIndex": 0,
        "pages": [
            1
        ]
    },
    "pageOfItems": [
        {
            "itemid": 48,
            "ownerid": 30,
            "name": "سجادة",
            "cityid": 1,
            "postingdate": "2020-08-08T13:35:30.000Z",
            "price": 100,
            "categoryid": 4,
            "lattitude": "31.534841399999998",
            "longitude": "35.077861299999995",
            "target": "للتبرع",
            "description": "",
            "cdescription": ""
        }
    ]
}
```

- Passed/failed : **Passed**

6. GET user/anyuser/:userid

- Input: GET /user/anyuser/30

- Expected output: Json data containing information about that userid

- Actual output:

```
[
        {
            "fname": "ديانا",
            "lname": "مجاهد",
            "phonenumber": "0597413706",
            "email": "diana.muj98@gmail.com"
        }
    ]
```

- Passed/failed : **Passed**

7. POST user/rating

- Body Input:

```
{
    "rateduserid":31,
    "ratingvalue":1
}
```

- Expected output:

```
{
   "updated": "success"
}
```

- Actual output:

```
{
   "updated": "success"
}
```

- Passed/failed : **Passed**

8. DELETE user/items/:itemid

- Input: DELETE user/items/46

- Header input :
  Authorization :
  eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VyaWQiOjMwLCJyb2xlIjoidXNlciIsImlhdCI6MTU5MTc3ODQzNn0.ihnGkDW5DUSbW1aA_1MmgS8AIg5pJKXk3FZd6meEt_4

- Expected output
  Json object contains the result of the delete

- Actual output:

```
{
    "fieldCount": 0,
    "affectedRows": 1,
    "insertId": 0,
    "serverStatus": 2,
    "warningCount": 0,
    "message": "",
    "protocol41": true,
    "changedRows": 0
}
```

- Passed/failed : **Passed**

9. PUT user/edituser

- Body Input

```
{
   "user": {
        "fname":"ديانا",
        "lname":"جهاد",
        "email":"dianamujahed@gmail.com",
        "phonenumber":"0597413706",
        "image":"user.png",
        "dateofbirth":null,
        "city":5,
        "username":"ديانا"
     }
}
```

- Header input :

  Authorization :

  eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VyaWQiOjMwLCJyb2xlIjoidXNlciIsImlhdCI6MTU5MTc3ODQzNn0.ihnGkDW5DUSbW1aA_1MmgS8AIg5pJKXk3FZd6meEt_4

- Expected output
- Json object contains the result of the update

- Actual output:

```
{
    "fieldCount": 0,
    "affectedRows": 1,
    "insertId": 0,
    "serverStatus": 2,
    "warningCount": 1,
    "message": "(Rows matched: 1  Changed: 1  Warnings: 1",
    "protocol41": true,
    "changedRows": 1
}
```

- Passed/failed : **Passed**

10. POST user/report

- Body Input

```
{
    "report": {
        "itemid":46,
        "massage":"سلعة غير مناسبة للعرض "

      }
}
```

- Header input :

  Authorization :
  eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VyaWQiOjMwLCJyb2xlIjoidXNlciIsImlhdCI6MTU5MTc3ODQzNn0.ihnGkDW5DUSbW1aA_1MmgS8AIg5pJKXk3FZd6meEt_4

- Expected output

```
{
    "created": "success"
   }
```

- Actual output:

```
{
    "created": "success"
}
```

- Passed/failed : **Passed**

11. GET admin/reports

- Header input :

  Authorization :
  eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VyaWQiOjMzLCJyb2xlIjoiYWRtaW4iLCJpYXQiOjE1OTY5MTg1ODh9.gvpkJmIossPEkCZ7pP4fhCuqYecAfSfWVr8ExyNPaFU

- Expected output

  Json object contais array of reports and pager object.

- Actual output:

```
{
    "pager": {
        "totalItems": 1,
        "currentPage": 1,
        "pageSize": 10,
        "totalPages": 1,
        "startPage": 1,
        "endPage": 1,
        "startIndex": 0,
        "endIndex": 0,
        "pages": [
            1
        ]
    },
    "pageOfItems": [
        {
            "reportid": 7,
            "reporterid": 30,
            "itemid": 46,
            "massage": "سلعة غير مناسبة للعرض",
            "reportdate": "2020-08-08T20:09:56.000Z"
        }
    ]
}
```

- Passed/failed : **Passed**

12. GET admin/users

- Header input :
  Authorization :
  eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VyaWQiOjMzLCJyb2xlIjoiYWRtaW4i
  LCJpYXQiOjE1OTY5MTg1ODh9.gvpkJmlossPEkCZ7pP4fhCuqYecAfSfWVr8ExyNPa
  FU

- Expected output
  Json object contais array of users and pager object.
- Actual output:

```
{
    "pager": {
        "totalItems": 4,
        "currentPage": 1,
        "pageSize": 10,
        "totalPages": 1,
        "startPage": 1,
        "endPage": 1,
        "startIndex": 0,
        "endIndex": 3,
        "pages": [
            1
        ]
    },
    "pageOfUsers": [
        {
            "userid": 30,
            "fname": "ديانا",
            "lname": "جهاد",
            "username": "ديانا",
            "email": "dianamujahed@gmail.com",
            "phonenumber": "0597413706",
            "image": "user.png",
            "dateofbirth": "0000-00-00",
            "role": "user",
            "city": "5"
        },
        {
            "userid": 31,
            "fname": "يافا",
            "lname": "فرج الله",
            "username": "يافا",
            "email": "yafa@gmail.com",
            "phonenumber": "599882058",
            "image": "user.png",
            "dateofbirth": null,
            "role": "user",
            "city": "1"
        },
        {
            "userid": 37,
            "fname": "بلقيس",
            "lname": "مجاهد",
            "username": "بلقيس",
            "email": "",
            "phonenumber": "",
            "image": "159546686485210006191 6_1905038969625993_2561
053535059836928_o.jpg",
            "dateofbirth": "0000-00-00",
            "role": "user",
            "city": "null"
        },
        {
            "userid": 39,
            "fname": "صفاء",
            "lname": "مجاهد",
            "username": "صفاء",
            "email": null,
            "phonenumber": null,
            "image": "user.png",
            "dateofbirth": null,
            "role": "user",
            "city": null
        }
    ]
}
```

- Passed/failed : **Passed**

Chapter 6: Conclusion and Future Work.

In the end, we created a website that allows people to display their used items that they want to get rid of by donating, selling or renting them, and it allows users to add the location of the displayed item to the map so that the address is clear to any buyer.

In the future, the site can be developed so that the seller can choose the charity to which he wishes to donate the price of his used item, and the notification feature will be added to inform users of the availability of a used item that he searched for in advance and did not find it available

.

In the near future, we will develop a hybrid application that works on Android phones and IOS phones, to facilitate its use and increase its popularity.

References:

[١] https://nodejs.org/en/about/

[٢] https://developer.mozilla.org/en-US/docs/Learn/Serverside/Express_Nodejs/Introduction

[٣] https://angular.io/docs