DEVELOPMENT OF A WEB APPLICATION “DIGITAL MARKET”

GRADUATE QUALIFICATION WORK
SUSU–02.04.02.2018.308-641.GQW

Supervisor
Cand. Sci., Assoc. Prof.

__________ P.S. Kostenetskiy

Author,
the student of the group CE-219

__________ Z.D.S. Albadran

Normative control

__________ O.N. Ivanova

“___”___________ 2019

Chelyabinsk–2019
TASK
of the master graduate qualification work
for the student of the group CE-229
Albadran Zainalabdeen D. Aldeen Salim
in master direction 02.04.02
“Fundamental Informatics and Information Technologies”
(master program “Database Technologies”)

1. The topic (approved by the order of the rector from 25.04.2019 No. 899)
   Development of a web application “Digital market”.

2. The deadline for the completion of the work: 05.06.2019.

3. The source data for the work

4. The list of the development issues
   4.1. Install DBMS MS SQL Server.
   4.2. Design and create a database.
   4.3. Design “Digital market” web application connected to the database.
   4.4. Design tests and test the application.

5. Issuance date of the task: 08.02.2019.

Supervisor
Cand. Sci, Assoc. Prof. P.S. Kostenetskiy

The task is taken to perform Z.D.S. Albadran
TABLE OF CONTENTS

INTRODUCTION ........................................................................................................... 5

1. REVIEW OF TECHNOLOGIES ............................................................................ 7
   1.1. Web Application .......................................................................................... 7
   1.2. Web programming technology .................................................................... 8
   1.3. The Request and Response Objects ............................................................ 10

2. DATABASE DESIGN .............................................................................................. 16
   2.1. Database schema ......................................................................................... 16
   2.2. The structure of Users table ....................................................................... 16
   2.3. The structure of ForgotPassRequests table ................................................ 17
   2.4. The structure of cart table ........................................................................... 18
   2.5. The structure of Items table ....................................................................... 19
   2.6. The structure of Item Images ..................................................................... 20
   2.7. The structure of table Categories ............................................................... 21
   2.8. The structure of table Subcategories .......................................................... 21
   2.9. The structure of table Brands .................................................................... 22
   2.9. The Structure of Table Transaction ............................................................ 22

3. SOFTWARE DESIGN ............................................................................................ 23
   3.1. Description of the Subject Area .................................................................. 23
   3.2 Use case model ............................................................................................. 23

4. SOFTWARE DEVELOPMENT .............................................................................. 26
   4.1. The Home page .......................................................................................... 26
   4.2. Implementation of the Sign UP page ............................................................ 26
   4.3. Implementation of log in page ..................................................................... 27
INTRODUCTION

The Internet is a global system of interconnected computer networks that use the standard Internet protocol suite (TCP/IP) to serve billions of users worldwide. It is a network of networks that consists of millions of private, public, academic, business, and government networks, of local to global scope. Internet has been the most useful technology of the modern times, Internet marketing, or online marketing, refers to advertising and marketing efforts that use the Web and email to drive direct sales via electronic commerce, in addition to sales leads from websites or emails [6].

Internet marketing and online advertising efforts are typically used in conjunction with traditional types of advertising such as radio, television, newspapers and magazines.

Internet Marketing is any means that use to market your business online. How to market to people has radically changed over the last few years and it can be very confusing to businesses. Social media in particular is an area that has radically changed how you market your business. Consumers are much smarter and are not as interested in messages being broadcast to them and are therefore engaging more with social networks. Through these they are finding out what products to buy, services to use, and so on, based on trusted recommendations from their friends and family. It is estimated that 78% of the time if you read a recommendation from friends online you trust that recommendation but what’s more interesting is that if you read that recommendation from someone you don’t know you still trust it 60% of the time.

Properly understood, it can be harnessed to complement your existing marketing practices, extend operations and create new opportunities. The key to successful marketing over the Internet is applying the strengths of the medium to proven traditional marketing practices in innovative ways.
Research goal and objectives

The goal of the research is making a web application to the online markets who want to make a discount on the Items “Digital market”. For the reaching the goal I must achieve following objectives.

1. Analyse the subject area
2. Install DBMS MS SQL Server.
3. Design and create a database.
4. Design “Digital market” web application connected to the database.
5. Design tests and test the application.

The practical significance

This project useful, because it contains important features to assist easy transaction online:

1) users can create and edit specifications of their Items,
2) users can make deal with web-application easily.
3) the web-application has no limit on the number of users.

Structure of the thesis

The thesis comprises of four chapters, introduction, review of technologies design of software appropriate and reference list.

In chapter one, the review of the technologies of the Internet and how the user deal with the Internet and include domain of problem. Also, there is the overview of what are the best cur- rent web development technologies, which we used to develop the project. In chapter Two there is description to the Database

In chapter three, there is design of software and analysis of the system, there is a description of functional requirements, non-Functional requirements, Use Case diagram, database scheme and the design the application interfaces.

The chapter four is Software Development there is all the interfaces and how the web site is work

Chapter fife is devoted to the functional testing of the application.
1. REVIEW OF TECHNOLOGIES

1.1. Web Application

To build a static HTML website before, or are coming from a PHP or ASP background, it will be probably used to the idea of the web server (Apache or IIS, for example) serving your static files so that a browser can view them over the network. For example, to create the file about.html, and put it in the proper directory, it can then navigate to http://localhost/about.html. Depending on your web server configuration, you might even be able to omit the .html, but the relationship between URL and filename is clear: the web server simply knows where the file is on the computer, and serves it to the browser [20].

Node offers a different paradigm than that of a traditional web server: the app that you write is the web server. Node simply provides the framework for you to build a webserver. “But I don’t want to write a web server,” you might be saying! It’s a natural response: you want to be writing an app, not a web server. However, Node makes the business of writing.

The core philosophy behind Node is that of event-driven programming. What that means for you, the programmer, is that you have to understand what events are available to you and how to respond to them. Many people are introduced to event-driven programming by implementing a user interface: the user clicks on something, and you handle the “click event.” It is a good metaphor, because it has understood that the programmer has no control over when, or if, the user is going to click something, so event-driven programming is really quite intuitive. It can be a little harder to make the conceptual leap to responding to events on the server, but the principle is the same. in the previous code example, the event is implicit: the event that’s being handled is an HTTP request. The http. Create Server method takes a function as an argument; that function will be invoked every time an HTTP request is made.
1.2. Web programming technology

Two terms thrown around a lot in the web industry are front-end and back-end. It can be a little frustrating since the difference between the front-end and back-end isn't always perfectly clear. They're terms often used to describe aspects of the web industry. The front-end is also referred to as the client-side and is sometimes considered "web design". The back-end of the web industry is often called the server-side. Often when someone says they're a "web developer" they're saying they work on the back-end of sites. While that explanation seems simple, the line between the two is often blurry. Here are some basic guidelines to help tell the difference between the front-end and back-end or at least be able to begin to understand what someone does when they say they're a "front-end developer."

Front-End: The front-end is everything involved with what the user sees, including design and some languages like HTML and CSS. There are a lot of different jobs associated with the front-end. A common front-end job title is "web designer". A web designer, you guessed it, designs websites. The job title of web designer is pretty broad, though. A web designer could just be someone who designs the sites in a program like Photoshop or Fireworks and will never touch the code. But in another location, a web designer could do all the design comps in Photoshop, and then be responsible for creating all the HTML and CSS (and sometimes even JavaScript) to go along with it. A User Interface (UI) Designer is basically a visual designer and is generally focused on design. They're not usually involved in the implementation of the design, but they might know light HTML and CSS so they can communicate their ideas more effectively to the developers.

User Experience (UX) designers work in the front-end and study and researches how people use the sites and make changes through a lot of testing. A front-end designer or developer can create a site without any back-end development. The sites they would create without a web developer, or using the back-end, is a static site. A static site is something like a site for a restaurant or hair salon. It doesn't require any information to be stored in a database. The pages will usually stay the same, unless it is time for a redesign.
A front-end developer may be required to have a grasp on testing, as well as be well versed in HTML, CSS and JavaScript. This person may or may not have experience with creating the design in a design program. A different version of this title is front-end engineer. Specific front-end languages like "JavaScript developer" are also considered front-end developers.

**Back-End:** The back-end, or the "server-side", is basically how the site works, updates and changes. This refers to everything the user can't see in the browser, like databases and servers. Usually people who work on the back-end are called programmers or developers. Back-end developers are mostly worried about things like security, structure and content management. They usually know and can use languages like HTML and CSS, but that's definitely not their focus. Back-end developers, or at least back-end development, is required to create a dynamic site. A dynamic site is a site that's constantly changing and updated in real-time. Most sites are dynamic sites, as opposed to static sites. Facebook, Google Maps and this blog are all considered dynamic sites. Blogs are dynamic sites, since their content is constantly changing and updating. A dynamic site requires a database to work properly. All information, like user profiles or images they've uploaded, or blog posts, are stored in the database. Web Developers work with programming languages like PHP or .Net, since they need to work with something the database understands. The code they write communicates with the server and then tells the browser what to use from the database. It gives the user the ability to update dynamically without going back to the designer. In some web applications, the Back End developer designs a control panel to the user to give the ability to add and remove news or photos, etc.

Data access layer (back end) of a piece of software, or the physical infrastructure or hardware. In the client–server model, the client is usually considered the front end and the server is usually considered the back end, even when some presentation work is actually done on the server itself. The Front End and back end Architecture showed in figure 1.
1.3. The Request and Response Objects

URLs are one of the most commonly used technology concepts today. Essentially, they’re the addresses you’d use to access various resources — most of the time, you’re using one to access a particular website available on the internet. Because URLs are "handled" so frequently by users, it’s important to take care when choosing your domain, as well as the folder structure of your website, since the decisions you make are apparent to users and impact their experience in navigating through your site. URLs are also significant when it comes to search engine optimization [6]. See figure 2.

![Fig. 1. The Parts of a URL](image-url)
Protocol: The protocol determines how the request will be transmitted. We will be dealing exclusively with http and https. Other common protocols include file and ftp.

Host: The host identifies the server. Servers on your computer (local host) or a local net-work may simply be one word, or it may be a numeric IP address. On the Internet, the host will end in a top-level domain (TLD) like .com or .net. Additionally, there may be subdomains, which prefix the hostname. Ww is a very common subdomain, though it can be anything. Subdomains are optional.

Port: Each server has a collection of numbered ports. Some port numbers are “special,” like 80 and 443. If you omit the port, port 80 is assumed for HTTP and 443 for HTTPS. In general, if you aren’t using port 80 or 443, you should use a port number greater than 1023. It’s very common to use easy-to-remember port numbers like 3000, 8080, and 8088.

Path: The path is generally the first part of the URL that your app cares about (it is possible to make decisions based on protocol, host, and port, but it’s not good practice). The path should be used to uniquely identify pages or other resources in your app.

Query string: The query string is an optional collection of name/value pairs. The query string starts with a question mark (?), and name/value pairs are separated by ampersands (&). Both names and values should be URL encoded. JavaScript provides a built-in function to do that: encode URI Component. For example, spaces will be replaced with plus signs (+). Other special characters will be replaced with numeric character references.

Fragment: The fragment (or hash) is not passed to the server at all: it is strictly for use by the browser. It is becoming increasingly common for single-page applications or AJAX-heavy applications to use the fragment to control the application. Originally, the fragment’s sole purpose was to cause the browser to display a specific part of the document, marked by an anchor tag (<a id="chapter06">).
The request object (which is normally passed to a call back, meaning you can name it whatever you want: it is common to name it ret or request) starts its life as an instance of http. Incoming Message, a core Node object. Express adds additional functionality. Let’s look at the most useful properties and methods of the request object all of these methods are added by Express.

The response object (which is normally passed to a call back, meaning you can name it whatever you want: it is common to name it res, rasp, or response) starts its life as an instance of http. Server Response, a core Node object. Express adds additional functionality.

I had choose the technologies of my Research for the front end and backend.

Front-end

HTML: HTML stands for Hypertext Mark-up Language. It allows the user to create and structure sections, paragraphs, headings, links, and block quotes for web pages and applications. HTML is not a programming language, meaning it doesn’t have the ability to create dynamic functionality. Instead, it makes it possible to organize and format documents, similarly to Microsoft Word. HTML is a format that tells a computer how to display a web page. The documents themselves are plain text files with special "tags" or codes that a web browser uses to interpret and display information on your computer screen.

CSS: The article introduces you to Cascading Style Sheets (CSS), defining what they are, and what they are used for on the Web. Cascading Style Sheets (CSS) is a language for specifying how documents are presented to users. A document is a collection of information that is structured using a mark-up language such as HTML (Hypertext Markup Language). CSS is a foundation technology, used today by most documents on the web [11].

The primary goal of CSS is to allow separation of a document’s presentation characteristics (formatting) from the document’s content. Separating content and formatting simplifies document creation and maintenance, improves presentation flexibility, and allows multiple HTML pages to share formatting by
placing formatting rules in an external file. CSS rules can control virtually every aspect of data formatting, including font, color, weight, spacing, positioning, background colors and images, link characteristics, and more.

**Back-end**

I have chosen C# With ASP.NET as a platform for the implementation of my project, SQL Server 2014 Management Studio as DBMS for the implementation of my project.

C#: is a programming language that works on the .Net platform and is produced by Microsoft and was sold at the same time as the .Net platform, and enables the language of #C of games, desktop applications and smart phone applications. The C # language is derived from the Java developers' efforts, which share many of its features, design and excel in some parts, and combine # languages between C ++, Visual Basic, and C # as other languages. Net is a multilanguage platform that can run programs on all operating systems, the language of C # is a powerful and easy-to-learn language at the same time. It relies on the .Net framework libraries, making it easy to write complex programs. It is one of the fastest and most efficient programming languages in the world. It is easy to learn the language of the #C is not difficult and you can begin to learn if you are a beginner in the field of programming, as it is characterized by easy to learn and understand and speed of learning also, and here are some sources in Arabic and English to learn this language. C# is a general object-oriented programming (OOP) language for networking and Web development. C# is specified as a common language infrastructure (CLI) language.

In January 1999, Dutch software engineer Anders Hejlsberg formed a team to develop C# as a complement to Microsoft’s NET framework. Initially, C# was developed as C-Like Object Oriented Language (Cool). The actual name was changed to avert potential trademark issues. In January 2000, NET was released as C#. Its NET framework promotes multiple Web technologies.
**ASP.Net**: ASP.Net is a tool or technology for developing a powerful web and windows application using different languages. ASP.Net is not a coding language. It is a software that provide plate form for developing powerful website and windows application.is an initial programming language designed by Microsoft for software development, or the latest development of Microsoft technology releases in Active Server Pages (ASP) and Visual Studio .NET ), Based on building applications in the Internet whether static or dynamic sites can be detailed as follows:– Static websites: They are ordinary web sites that contain images, texts or, more simply, websites that are written on HTML so that each page is separated from the rest of the pages and there are no databases. Dynamic websites: sites that allow you to easily change, delete or add images or information from pages through the webmaster without returning to the site designer, where data and information on its pages are displayed and entered by databases, which allows to update and add to the pages continuously and therefore was called "dynamic sites " [10].

**SQL Server 2014 Management Studio**: SQL (Structured Query Language) is a standardized programming language that's used to manage relational databases and perform various operations on the data in them. Initially created in the 1970s, SQL is regularly used not only by database administrators, but also by developers writing data integration scripts and data analysts looking to set up and run analytical queries. is a relational database management system developed by Microsoft. As a database server, it is a software product with the primary function of storing and retrieving data as requested by other software applications—which may run either on the same computer or on another computer across a network (including the Internet). Microsoft markets at least a dozen different editions of Microsoft SQL Server, aimed at different audiences and for workloads ranging from small single-machine applications to large Internet-facing applications with many concurrent users.
SQL Server is chock-full of features that support data management and facilitate data access. Most are respected both far and wide. Others are counted among the despised. A number of them fall into both camps, often among the same users, many of whom openly acknowledge their love/hate relationships. Few SQL Server features come out entirely unscathed. In the first article in this series, we covered some of the more controversial of these features, such as Data Quality Services and Master Data Services, neither of which has a particularly large fan base. In this article, we continue the discussion with a look at another set of features, those that also have their critics and their devoted followers. Few issues are ever black-and-white, and SQL Server is no exception [13].
2. DATABASE DESIGN

2.1. Database schema

A database schema is the skeleton structure that represents the logical view of the entire database. It defines how the data is organized and how the relations among them are associated. It formulates all the constraints that are to be applied on the data [14].

A database schema defines its entities and the relationship among them. It contains a descriptive detail of the database, which can be depicted by means of schema diagrams. It’s the database designers who design the schema to help programmers understand the database and make it useful. The Fig.3 is show the Database schema.

![Database Schema Diagram](image)

Fig. 2. Database Schema

2.2. The structure of Users table

Providing a successful web application is required a good user interface, UI design can make or break the success of your website or application in web design, great user interface, or UI design, is all about helping the user to accomplish a given task as simply and efficiently as possible:

1. Uid: the primary key of the Users table from type (int);
2. Username: the user name of the user from type (nvarchar(max))
3. Password: the password of the user from type (nvarchar(max))
4. Email: the email of the user so they can register to the site from type (nvarchar(max))
5. Name: the name of the user from type (nvarchar(max))
6. Usertype: the type of the user that he registered he will choose seller or buyer from type (nchar(10))

Fig. 4. User Table

2.3. The structure of ForgotPassRequests table.

Sometimes the user forget his password or Username, so he need Recovery to restore his password or username. This table is designed to the users, if the user forgot the password he can get it again

Fig. 5. The ForgotPassRequests Table

1. Id: the primary key of the ForgotPassRequests table from type (uniqueidentifier).
2. Uid: the Id of the user from type (int).
3. RequestDateTime: the time and date from type (datetime).
2.4. The structure of cart table

Cart table contain information about cart. This is used to put what the buyer had choose from the items in it before buying. Its consist of 12 fields. These fields have different types of data: money, datetime. The primary key is the field “Cart_id”.

![Cart Table]

Fig. 6. The Cart Table

Its consist of 12 fields. These fields have different types of data: money, date time. The primary key is the field “Cart_id”, it is automatically incremented, and we could see there is one foreign key “buyerid” that are used to connect this table with “User” table. This table is describe the connection of the cart that Buyer use it to buy the goods.

1. Card_ID: the primary key of the Cart Table from type (bigint).
2. Buyer ID: the foreign key of the cart table from type (int).
3. Cart Amount: the Amount that the cart is connecting with from type (money).
4. Cart Discount: if there is a discount on some goods the buyer can take it from type (money).
5. Total Payed: is the total of the money that the user has pay from type (money).
6. Payment Type: the type of paying and which type of the cart that used to pay from type (nvarchar (50)).
7. Payment Status: the status of the paying its give you the message if the buyer had pay for the good from type (nvarchar (50)).
8. Date of Purchase: it describes the time of the purchase’s time from type (datetime).
9. Name: the user’s name from type (nvarchar (max)).
10. Date of Purchase: it describes the time of the purchase’s time from type (datetime).
11. Name: the user’s name from type (nvarchar (max)).
12. Address: the user’s address from type (nvarchar (max)).
13. Pin code: the main Pin code of the card from type (nvarchar (50)).
14. Mobile number: the user’s mobile number from type (nvarchar (50)).

2.5. The structure of Items table

Every web App. For shopping needs Items to sell or buy the table that contain the items of the X-Market that user operations deals with. Sell or buy.

Where the:
1. Item_ID: is the primary key of the table Items from type(bigint)
2. Item_Name: the name of the Item the user will choose it for example iPhone X or Casio watch from type (varchar (max)).
3. Item_owner_id: It is the Item of the owner.
5. Item_selPrice: It’s the Id of the Item that the Owner that want reduction.
6. Item_selPrice: It’s the Id of the Item that the Owner that want reduction.
7. Item_BrandID: it’s the brand of the Item for example the phone from brand Iphone from type (bigint).
8. Item_categoryID: it’s the category of the Items like phones or
television from type (bigint).

9. Item_subCatID: it’s the sub category of the Item like Iphone from
category phones or Sony from television category from type (bigint).

10. Item_Details: it’s the details of the Item like how much hard space of
the Iphone or how much pixels in the television that the user that write
it from type (nvarchar (max)).

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Data Type</th>
<th>Allow Nulls</th>
</tr>
</thead>
<tbody>
<tr>
<td>Item_ID</td>
<td>bigint</td>
<td></td>
</tr>
<tr>
<td>item_Name</td>
<td>nvarchar(MAX)</td>
<td></td>
</tr>
<tr>
<td>item_owner_id</td>
<td>int</td>
<td></td>
</tr>
<tr>
<td>item_Price</td>
<td>money</td>
<td></td>
</tr>
<tr>
<td>item_SellPrice</td>
<td>money</td>
<td></td>
</tr>
<tr>
<td>item_BrandID</td>
<td>bigint</td>
<td></td>
</tr>
<tr>
<td>item_CategoryID</td>
<td>bigint</td>
<td></td>
</tr>
<tr>
<td>item_SubCatID</td>
<td>bigint</td>
<td></td>
</tr>
<tr>
<td>item_Details</td>
<td>nvarchar(MAX)</td>
<td></td>
</tr>
<tr>
<td>item_Quantity</td>
<td>int</td>
<td></td>
</tr>
</tbody>
</table>

Fig.7. Items table

2.6. The structure of Item Images.

Every Item that seller upload it needs image so can buyer see the Item This
table is contain the images of the Items that the user uploaded it

Where:

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Data Type</th>
<th>Allow Nulls</th>
</tr>
</thead>
<tbody>
<tr>
<td>[item_IMGID]</td>
<td>bigint</td>
<td></td>
</tr>
<tr>
<td>item_ID</td>
<td>bigint</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>nvarchar(MAX)</td>
<td></td>
</tr>
<tr>
<td>Extention</td>
<td>nvarchar(50)</td>
<td></td>
</tr>
</tbody>
</table>

Fig.8. The table ItemImages
1. [item.IMAGID]: is the primary key of the table from type (bigint).
2. Item_ID: it’s the Id of the item(foreign key) from type(bigint)
3. Name: the name of the image from type (nvarchar).
4. Extension: it is to store all the Extension of the image

2.7. The structure of table Categories

Category table contain information about different types of categories of products. Its consist of 2 fields, see table 6.the primary key is the field “CatID”, it is automatically incremented field. It’s the category of the Items like phones or televisions.

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Data Type</th>
<th>Allow Nulls</th>
</tr>
</thead>
<tbody>
<tr>
<td>CatID</td>
<td>bigint</td>
<td></td>
</tr>
<tr>
<td>CatName</td>
<td>nvarchar(MAX)</td>
<td></td>
</tr>
</tbody>
</table>

Fig.9. The table of categories

1. Cat ID: the primary key of the table from type (bigint).
2. CatName: the name of category like phones televisions from type (nvarchar).

2.8. The structure of table Subcategories

It’s the sub category of the Item like Iphone from category phones or Sony from televisions category where:

1. SubCatID: it’s the primary key of the table form type (bigint).
2. SubCatName : it’s the name of the subcategories
3. MainCatID: it’s the main categoryID

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Data Type</th>
<th>Allow Nulls</th>
</tr>
</thead>
<tbody>
<tr>
<td>SubCatID</td>
<td>bigint</td>
<td></td>
</tr>
<tr>
<td>SubCatName</td>
<td>nvarchar(MAX)</td>
<td></td>
</tr>
<tr>
<td>MainCatID</td>
<td>bigint</td>
<td></td>
</tr>
</tbody>
</table>

Fig 10. SubCategory Table
2.9 The structure of table Brands

![Fig 11. Table Brands](image1)

1. BrandID: it’s the primary key of the table from type (bigint).
2. Name: it’s the name of the brand from type (nvarchar).

2.9. The Structure of Table Transaction

It’s the table that describe the transaction on the web app. That can the Admin know all the reports and Actions on the web app.

![Fig.12. the Table Transaction](image2)

Where:
1. Tr_id: it’s the primary key of the table Transaction from type (int).
2. Seller_id: it’s the seller ID from table users from type (int).
3. Buyer_id: it’s the buyer ID from table users from type (int).
4. Item_id: it’s the item ID from type Items from type (bigint).
5. Date_of_transaction: it’s the date of the transactions operations. It’s about when the seller are sell the item this table for the admin from type (date).
3. SOFTWARE DESIGN

3.1. Description of the Subject Area

I want to create a web Application because I need this system to Make finding the discounts on Items in Markets more Easy.

In the Web Application the buyer and seller (user) they have a profile so the buyer can register to the web application and the seller too. And each user have his own profile.

The buyer will see the all the Items that he want to buy study, the prices, Photos of the Items and he have A cart to put all the Items that he want to buy all marks that cannot login into the teacher seller page.

When the seller login into this site he can see all functions for Adding the Items or adding the brands of the Item.

So he can add his Item and the buyer can see his Items

This web application is based on C# web platform so it is a reliable website because it is considered one of the best in building websites, quicker and there is no user can change or hack it

This chapter is devoted to design of the system. There is a description of the system and we will show a design of users interface in Fig.13.

3.2 Use case model

A use case is a methodology used in system analysis to identify, clarify, and organize system requirements. The use case is made up of a set of possible sequences of interactions between systems and users in a particular environment and related to a particular goal. It consists of a group of elements (for example, classes and interfaces) that can be used together in a way that will have an effect larger than the sum of the separate elements combined. The use case should contain all system activities that have significance to the users. A use case can be thought of as a collection of possible scenarios related to a particular goal, indeed, the use case and goal are sometimes considered to be synonymous.
A use case is a description of set of sequence of actions that a system performs that yields an observable result of value to a particular actor.

An actor models a role played by external entities (such as human users or other systems) that interact with the subject by interchanging signals and data. In any case, the entity represented by an actor is external to the subject [16].

A use case diagram is a graphic depiction of the interactions among the elements of a system.

A use case is a methodology used in system analysis to identify, clarify, and organize system requirements. In this context, the term "system" refers to something being developed or operated, such as a mail-order product sales and service Web site. Use case diagrams are employed in UML (Unified Modelling Language), a standard notation for the modelling of real-world objects and systems. Use case diagram are shown in (Figure 13.). Talk about actors of the system.

![Use Case Diagram](image)

Fig.13. Use Case Diagram
List of Functions.

1. Function “Sell” is available for the customer “Seller” so he can Fully Managing sales. (CRUD)
2. Function “Upload Item” is available for the customer “Seller” so he can Fully Managing Upload of the Items.
3. Function “Upload Photo” is available for the customer “Seller” so he can Fully Managing Upload of photo of Items
4. Function “Brows” is available for the customer “Buyer”, “Seller” and “so he can Fully Browsing in the Application.
5. Function “Buy” is available for the customer “Buyer” so he can fully buy from the Application.
6. Function “Input Item to cart” is available for the customer “Buyer” so he can fully input the Item that he Want to The Cart in the Application.
7. Function “See Report of Item Transaction” is available for the customer “Admin” so he can All the information of Item sells

So we can name the seller and buyer like (Users) in the Database to be more eyes to describing the Database of the system that I made it by using the Microsoft SQL Server
4. SOFTWARE DEVELOPMENT

4.1. The Home page

Providing a successful web application is required a good user interface, UI design can make or break the success of your website or application in web design, great user interface, or UI design, is all about helping the user to implement views of a website. Shows the main page for the web site of a system.

It contains the navigation bar in the top of the page: ”Home”, “Sign in”, “sign in” see (Fig.14).

![Fig.14. Home Page](image)

4.2. Implementation of the Sign UP page

The user must enter his username, name, email, Password and confirm password after that press bottom “sign up”. It’s accessible to all user. All the users can sign up ether as Sellers or Buyers, so the buyer can sign up as a buyer and seller can sign up as seller or both, that’s mean the user can sign in as a buyer and seller in the same time, its contain the (username of the user, the password of the user, the confirm of the password, the name of the user, the E-mail of the user, the type of the user) in the type of the user, the user can choose ether buyer or seller. See (Fig.15)
4.3. Implementation of log in page

Describe how the user can log in into the Web App. After he sing up, it’s contain the username that he registered it in the sign up page and the password, That he already choose it in the sig up page too, and if the user didn’t sign up, he will back again to the signup page again(Fig.16).

The data of the Users will store in the database automatically, then the Admin can see all the information of the users and manage them in table Users see (Fig.17).
The seller have a Functions and profile so he can change the name and the E-mail, the Functions that he can do (Manage, Items), see (Fig.18).

We can see the Functions (Home, Logout) where the “home” helps to back to the home page and “Logout” helps to logout from the profile. Each (manage, Items) have Functions, the Manage have three Functions (Add Brand, Add Category, Add Sub Category), see, (Fig.19).
Fig. 19. Function Manage

The Function (Add brand) helps the user to choose the brand of the Item that he wants to sell like he wants to sell a phone for example there is a phone from apple brand so he will write Apple to add the brand for his Item so he needs to add the brand name see (Fig.20).

Fig. 20. Function Brand

The user’s information for Adding brand will store in the table of brand Table See (Fig.21)
The Function (Add category) helps the seller to add the category of the Item for example Computer, televisions see (Fig.22).

All the information to add category of the Item will store in the table category (Fig.23)

The Function (Add sub category) this function helps the seller to add the category of the Item for example which type of phones he will add wire or wireless phones so he need to add the main category name and the sub category see for
example the main category will be mobile device but the sub category will be wireless phone (Fig.24).

![Fig.24. Add Subcategory Function](image)

When you press on Main Category All the Categories that you save it will called already see (Fig.25).

![Fig.25. The function main category](image)
The information of this function will store in sub category table (Fig. 26).

<table>
<thead>
<tr>
<th>SubCatID</th>
<th>SubCatName</th>
<th>MainCatID</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Iphone</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>Galaxy</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>Laptop</td>
<td>2</td>
</tr>
<tr>
<td>NULL</td>
<td>NULL</td>
<td>NULL</td>
</tr>
</tbody>
</table>

Fig. 26. The information of subcategory of table

The function Item have one function this function are the Important, It helps the seller to upload his Item, when the seller press on the Items he will have Function “Add Item”, see (Fig. 27).

Fig. 27. Item Function in seller

This functions have (Name, Quantity, price, selling price, Brand, category, sub category, Details, five images for the Items that he want to sell). The Name helps the seller to add a name to his Item, the Quantity helps him to add how much Items that he want to sell, the Price helps to add the price of the Item that he want to sell, Selling price helps the seller to add reduction if he want, the Brand helps to choose the brands that he already Add, the Category helps to choose the
category that he already add the *Subcategory* helps to choose the sub category that he already add, the *Details* helps to add some details to his Item for example, how much memory on the phone, and five images for the *Items* that he want to sell, for example he add phones and laptops and televisions so he need to add some images see (Fig.28 and Fig.29). The page that accessible by register user.

![Fig.28. The function of Add Items (part 1)](image)

![Fig.29. The function of Add Items (part 2)](image)
All the information of the Function Add Item will store in the Item table (Fig.30)

<table>
<thead>
<tr>
<th>Item_ID</th>
<th>Item_Name</th>
<th>Item_owner_id</th>
<th>Item_Price</th>
<th>item_SellPrice</th>
<th>Item_BrandID</th>
<th>Item_CategoryID</th>
<th>Item_SubCatID</th>
<th>Item_Details</th>
<th>Item_Quantity</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>7 Plus</td>
<td>2</td>
<td>1000.0000</td>
<td>800.0000</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>64GB</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>Note9</td>
<td>2</td>
<td>1000.0000</td>
<td>800.0000</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>64GB</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>LAPTOP</td>
<td>4</td>
<td>500.0000</td>
<td>490.0000</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>500 HD</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>NULL</td>
<td>NULL</td>
<td>NULL</td>
<td>NULL</td>
<td>NULL</td>
<td>NULL</td>
<td>NULL</td>
<td>NULL</td>
<td>NULL</td>
</tr>
</tbody>
</table>

Fig.30. The information of the table Item

4.5. Implementation of Buyer profile page

We can see the profile of the buyer he can Edit his profile he can change his name and E-mail.

The buyer have two functions “Items, Cart” (Fig 31)

Fig.31. The buyer profile
When the buyer click on Items Function the buyer will have Function “All Items” See (Fig. 32)

![All Items Function](image1)

**Fig.32. All Items Function**

The Function “All Items” it have all the Items that Seller Already uploaded (Fig.33).

![All Items Page](image2)

**Fig.33. the page of All Items**

So he can Brows on the Web App. And choose the Item that he want to buy by click on the Item (Fig.34)
When the buyer want to buy the Item he will click on button “Add to Cart” he will add the Item to the Cart see (Fig.35)

Then the cart page will contain all the Item that the buyer want to buy, so the buyer when click on Cart Function he will go to cart page see (Fig.36)
The cart contain a button “Buy Now”, when the buyer is ready to buy the Items that he choose he will Click on the button to buy, this button have a Functions (Name, Address, Mobile Number). See (Fig.37)

The Name is the name of the buyer and the Address is the Address of the buyer to get his Items and, Mobile Number is the Number of the buyer.

The information of the cart page will store in the cart table (Fig.38)
4.6. Implementation of Admin profile page

Every Web Application it should have an Admin Can Manage the Application but in This case the Admin can see all the Items in Application and see all the buyers and Sellers so in this case he can know which buyer that buy and seller that sell and see The Total Price for all the Items that sell, see Figure 39.

![Admin Page]

4.7. Implementation of source Code

Several fragments of C# code

Fig. 40 is showing code describing how the users can sign in to the Application Successfully, its helps the user to Use his own Account, if the user
didn’t write the User Name and Password right, the user will back again for Sign in Page to Write the Username and password successfully.

```csharp
public partial class SignIn : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        if (!IsPostBack)
        {
            if (Request.Cookies["UNAME"] != null & Request.Cookies["PWD"] != null)
            {
                UserName.Text = Request.Cookies["UNAME"].Value;
                Password.Text = Request.Cookies["PWD"].Value;
                CheckBox1.Checked = true;
            }
        }
    }
}
```

Fig. 40. Code for Login Implementations

Fig. 41 it’s showing the Implementation of the Page of the Items that the seller uploaded it, and the buyer can see it in one special page

```csharp
public partial class AddBrand : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        if (Session["USERID"] == null)
        {
            Response.Redirect("~/SignIn.aspx");
        }
        else
        {
            if (!IsPostBack)
            {
                //BindBrandsRptr();
            }
        }
    }
}
```

Fig. 41. Code of Implementations Seller Items

Fig. 42 showing the Implementation of the Buyer Page when the user wants to sign in like Buyer so will have a special Page for buying the Items he can see
all the Items and the discount on the Items, and this page contain several of Functions the buyer can use it.

Fig. 42. The Implementation of the Buyer Page

Fig.43 is showing the Code of Add Category it’s to save the Category of Items That the Seller had already uploaded it and have chosen in the Database, its helps the seller to choose the category of the Items that he wants and save the Name of Category in the Database.

Fig.44 is showing the Cart Implementation, The Cart is helping the buyer to choose a several Items and Buy them, when the buyer put the Items in Cart, The Items will Stored in The Database, and this Code bring the Items from Database to Web Page.
```csharp
private void BindMainCategory()
{
    string CS = ConfigurationManager.ConnectionStrings["MyDatabaseConnectionString"].ConnectionString;
    using (SqlConnection con = new SqlConnection(CS))
    {
        SqlCommand cmd = new SqlCommand("select * from tblCategories", con);
        con.Open();
        SqlDataAdapter sda = new SqlDataAdapter(cmd);
        DataTable dt = new DataTable();
        sda.Fill(dt);

        if (dt.Rows.Count != 0)
        {
            ddlCategory.DataSource = dt;
            ddlCategory.DataTextField = "CatName";
            ddlCategory.DataValueField = "CatID";
            ddlCategory.DataBind();
            ddlCategory.Items.Insert(0, new ListItem("-Select-", "0"));
        }
    }
}
```

Fig. 43. The Code of Add Category Implementation

```csharp
public partial class Cart : System.Web.UI.Page
{
    0 references
    protected void Page_Load(object sender, EventArgs e)
    {
        if (Session["USERID"] == null)
        {
            Response.Redirect("~/SignIn.aspx");
        }
        else
        {
            if (Session["PPay"] == "Paid")
            {
                Session["PPay"] = "not";
                mpePopup.Show();
            } else if (IsPostBack)
            {
                BindCartProducts();
            }
        }
    }
```

Fig.44. the Cart Implementation Code
5. TESTING

Software testing is very important for prove how the application, it is an investigation conducted to provide stakeholders with information about the quality of the software product or service under test. Software testing can also provide an objective, independent view of the software to allow the business to appreciate and understand the risks of software implementation. Test techniques include the process of executing a program or application with the intent of finding software bugs (errors or other defects), and verifying that the software product is fit for use [22].

Software testing can be conducted as soon as executable software (even if partially complete) exists. The overall approach to software development often determines when and how testing is conducted. For example, in a phased process, most testing occurs after system requirements have been defined and then implemented in testable programs. In contrast, under an agile approach, requirements, programming, and testing are often done concurrently.

Table 1. The protocol of functional testing of the site

<table>
<thead>
<tr>
<th>№</th>
<th>Test name</th>
<th>Steps</th>
<th>Expected Result</th>
<th>Is the test passed?</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Verify user authorization.</td>
<td>Unauthorized user logs on to any pages of the web application.</td>
<td>The web application redirects the user to the login page.</td>
<td>YES</td>
</tr>
<tr>
<td>2</td>
<td>Adding items</td>
<td>1. Seller open Items page.</td>
<td>New Item added to Item table and available to the customers.</td>
<td>YES</td>
</tr>
<tr>
<td>№</td>
<td>Test name</td>
<td>Steps</td>
<td>Expected Result</td>
<td>Is the test passed?</td>
</tr>
<tr>
<td>----</td>
<td>-------------------------------</td>
<td>----------------------------------------------------------------------</td>
<td>----------------------------------------------------------------------------------</td>
<td>---------------------</td>
</tr>
<tr>
<td>3</td>
<td>Removing Items</td>
<td>1. Seller open Items Page.</td>
<td>Item Removed from the page of Items in the Seller Page</td>
<td>YES</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2. Seller Remove Item Item Removed from the page of Items in the Seller Page</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Updating Items</td>
<td>1. Seller open the Items Page</td>
<td>Item Updated From the page of Items From the page of the seller</td>
<td>YES</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2. Seller can Update the Items</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Reading Items</td>
<td>1. The seller open the Items Page</td>
<td>The Seller can See and read All the Items in the page seller items</td>
<td>YES</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2. The seller Can see All Items in the page</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Deleting Items</td>
<td>1. Seller open The Items Page</td>
<td>The Seller can Delete the Items From The page of Items</td>
<td>YES</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2. The Seller Can Delete the Items</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Uploading the Photos of Items</td>
<td>1. Seller open The page of Items</td>
<td>The seller can Add photo of the Items that he want to Sell to show it to the Buyer</td>
<td>YES</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2. the seller click on Add Items</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>3. the Seller can add photo</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Browsing</td>
<td>The user can Brows in the Web Application successfully</td>
<td>All the users can Brows in the Web Application and see own Functions</td>
<td>Yes</td>
</tr>
<tr>
<td>№</td>
<td>Test name</td>
<td>Steps</td>
<td>Expected Result</td>
<td>Is the test passed?</td>
</tr>
<tr>
<td>----</td>
<td>---------------------------</td>
<td>----------------------------------------------------------------------</td>
<td>---------------------------------------------------------------------------------</td>
<td>---------------------</td>
</tr>
<tr>
<td>9</td>
<td>Buying Item</td>
<td>1. Buyer open his own page</td>
<td>Buyer can buy the Item that Seller already uploaded and the seller can know the buyer from his own Page to contact with him</td>
<td>YES</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2. buyer will choose All Items function</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>3. buyer add the Item to the Cart</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>4. buyer can buy the Item from the cart</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>User Logout</td>
<td>The user can logout from Web Application by choosing the Function logout from his profile.</td>
<td>The user Can logout when he finish his buying or selling</td>
<td>YES</td>
</tr>
<tr>
<td>11</td>
<td>Seeing the information about selling and buying</td>
<td>The Admin Can go to his page and open it</td>
<td>The admin can see all the Items that buy and sell</td>
<td>YES</td>
</tr>
<tr>
<td>12</td>
<td>Calculate all the transaction</td>
<td>The Admin can go to his page and open it</td>
<td>The Admin can see Total price of the Items</td>
<td>Yes</td>
</tr>
</tbody>
</table>
CONCLUSION

I have created a web application to add online markets who wants to make a discount on the Items in the web application. It is based on ASP.Net platform. ASP.net takes care of much of the hassle of web development, so you can focus on writing your app without needing to reinvent. It’s free and open source.

During the developing of the web application, I solved the following tasks:
1. I installed Database management studio MS SQL Server.
2. Designed and created a database for my project.
3. Design “Digital market” web application connected to the database.
4. I studied C# programing language and ASP.Net platform and database SQL server.
5. I created the software design that has a description of the subject area for research, the entity relationships diagram for the Web Application and the database schema for Web Application.
6. Software development that has created the project in ASP.net Platform, the database in SQL server, the models, and web pages for users who work in this Online Market.
7. Software testing that has a test for every user in the site including, Users and Admin and I have given the permission for every user to access their own information.
REFERENCES

8. Code analogies [Electronic resource] URL:
9. Search sql server [Electronic resource]URL:
https://searchsqlserver.techtarget.com/definition/SQL (the date of access: 16.10.2019).
11. Github [Electronic resource] URL:
   https://webplatform.github.io/docs/tutorials/learning_what_css_is (the date of access: 29.10.2019).
12. Hostinger [Electronic resource] URL:
13. Techopedia [Electronic Resource] URL:
   https://www.techopedia.com/definition/26272/c-sharp (The date of access: 2.11.2019).
14. Tutorials point: Database Schema [Electronic resource] URL:
15. Echtarger [Electronic resource] URL:
   https://searchsoftwarequality.techtarget.com/definition/use-case (the date of access: 19.11.2019).
16. Techtarget [Electronic resource] URL:
   https://whatis.techtarget.com/definition/use-case-diagram (the date of access: 19.11.2019).
