

MINISTRY OF EDUCATION AND SCIENCE  
OF THE RUSSIAN FEDERATION

Federal State State-Financed Educational Institution of High Professional Education  
South Ural State University (National Research University)  
Faculty of Computational Mathematics and Informatics  
Department of System Programming

THESIS IS CHECKED

Reviewer, Director of Grid-Engineering

V.A. Dorokhov

\_\_\_\_\_ 2016



ACCEPTED FOR THE DEFENSE

Head of the department, Dr. Sci., Prof.

\_\_\_\_\_ L.B. Sokolinsky

“13” 06 2016

**DEVELOPMENT OF CLASSIFIEDS WEB APPLICATION FOR  
THE CITY OF BAGHDAD**

GRADUATE QUALIFICATION WORK  
SUSU-02.04.2016.115-119.GQW

Supervisor

PhD., Assoc. Prof.

\_\_\_\_\_ G.I. Radchenko

Author,

the student of the group VMI-216

\_\_\_\_\_ A.A. Al-Waeli

Normative control

\_\_\_\_\_ O.N. Ivanova

“10” 10 2016

Chelyabinsk, 2016

MINISTRY OF EDUCATION AND SCIENCE OF THE RUSSIAN FEDERATION

Federal State State-Financed Educational Institution of High Professional Education  
South Ural State University  
(National Research University)  
Faculty of Computational Mathematics and Informatics  
Department of System Programming

ACCEPTED

Head of the department,  
Dr. Sci., Prof.

  
L.B. Sokolinsky

10.02.2016г.

**TASK**

**of the master graduate qualification work**

for the student of the group VMI-216

Al-Waeli Ahmed Abbood

in master direction 02.04.02

“Fundamental Informatics and Information Technologies”  
(master program “Database Technologies”)

1. **The topic** (approved by the order of the recto from 15.04.2016 No. 661)  
Development of classifieds web application for the city of Baghdad.
2. **Thesis basic data**
  - 2.1. Ebay web-shop [Electronic resource] URL: <http://www.ebay.com/> (date of access 16.04.2015).
  - 2.2. Amazon web-shop [Electronic resource] URL: <http://www.amazon.com/> (date of access 16.04.2015).
  - 2.3. Erik R. Getting Started with ASP.NET 4.5 Web Forms and Visual Studio 2013. E-book Gallery for Microsoft Technologies. – USA: Microsoft Corporation, 2014. – 200 p.
3. **The list of the development issues**
  - 3.1. Provide a review of web applications platforms.
  - 3.2. Provide a review of classifieds web applications.
  - 3.3. Develop a use-case model for classifieds web application.
  - 3.4. Develop an architecture for classifieds web application.
  - 3.5. Implement and test classifieds web application.
4. **Issuance date of the task:** 09.02.2016

**Supervisor**

PhD., Assoc. Prof.

The task is taken to perform



G.I. Radchenko

Al-Waeli Ahmed Abbood

Student: Al-Waely Ahmed Abbood Ali

Supervisor: PhD, Assoc. Prof. G.I. Radchenko

Topic: Development of classified web application for the city of Baghdad

**The calendar plan  
of the execution of master graduate qualifying work (GQW)**

No	Phase	Duration	Deadline	Report	Actual date of execution	Supervisor's signature
1.	Introduction and literature review	1 month	February, 25 <sup>th</sup>	1. Task of the master graduate qualification work 2. Text of Introduction 3. References	25.02	
2.	Development of the model, design of the system	1 month	March, 15 <sup>th</sup>	1. Text of chapter 1 (theoretical part).	15.03	
3.	Implementation of a system	1 month	April, 15 <sup>th</sup>	1. Software system 2. Text of chapter 2 (implementation part).	15.04	
4.	Testing and debugging of the system, experiments	2 weeks	May, 1 <sup>st</sup>	1. Set of tests 2. Text of chapter 3 (experimental part).	1.05	
5.	Full text	2 weeks	May, 15 <sup>th</sup>	1. Full text of GQW	15.05	
6.	Validation of the text by supervisor	1 week	May, 22 <sup>nd</sup>	1. Electronic version of the GQW text checked by the supervisor	22.05	
7.	Normative control	3 days	May, 25 <sup>th</sup>	1. Twisted text of GQW signed by student, supervisor and normative controller	25.05. 10.06.2016	
8.	Proposal defense	1 week	May, 25 <sup>th</sup> – June, 1 <sup>st</sup>	1. Twisted text of GQW signed by student, supervisor and normative controller for the signature of the Head of the Department Head about accepting for the defense 2. A signed review of the supervisor 3. A review of the reviewer, signed and notarized at his place of work 4. Implementation act (if exists) 5. Presentation of the report in PowerPoint	01.06.2016 1.06	

# CONTENTS

ACKNOWLEDGEMENTS .....	4
INTRODUCTION.....	5
1. RELATED WORKS .....	6
1.1. Web applications platforms.....	6
1.1.1. Python and Django platform .....	6
1.1.2. ASP.NET platform.....	7
1.1.3. Microsoft Azure platform .....	8
2. CLASSIFIEDS WEB APPLICATIONS REVIEW .....	10
2.1. Mredy .....	10
2.2. Avito.....	11
2.3. Ebay.....	12
2.4. Amazon .....	13
3. REQUIREMENTS DEVELOPMENT .....	14
4. SYSTEM ARCHITECTURE AND IMPLEMENTATION.....	16
4.1. ER-Diagram .....	16
4.1.2. Database Structure.....	16
4.2. Web shop-Application development Database.....	17
4.2.1. Product class.....	17
4.2.2. Category Class.....	18
4.2.3. Database Initializer class.....	19
4.2.4. Shopping Cart Actions .....	20
4.3. SQL Codes .....	20
5. WEB SHOP IMPLEMENTATION AND TEST.....	21
5.1. Main Page.....	21
5.2. Login Page.....	22
5.3. Catalog page.....	23
5.4. Item description.....	25
5.5. Add or remove items in the products' list .....	26
5.6. Add or remove items in the shopping cart .....	27
REFERENCES .....	29

## **ACKNOWLEDGEMENTS**

I would like to express thanks to God who helped me for doing this work.

I want to thank my deepest thanks from the bottom of my heart to PhD, associate professor, G.I. Radchenko, my supervisor for his help, efforts, an support and continuous enthusiasm and encouragement throughout the project is also very grateful and extend my sincere thanks to all staff working in the department of system programming of the faculty of computational mathematics and informatics in the South Ural State University.

I would like express my thanks from the depth of my heart to Russia and her people who gave me this chance in order to improve my life and build my future.

Finally many thanks to my family, father, mother, sisters and brother and all people who encouraged me.

## **INTRODUCTION**

I would like to create a web application for online advertisement (classifieds web application) to make help the people who they live in Baghdad city to benefit from its services to search their various desirable goods without going to the shops and wasting their time in transportation between the several shops.

Using this service, the advertisers should be able to show their goods and services to be seen to all people who use internet service. Advertisers would be able to provide description of their items and services, add photos and suggest the price and other details of their items.

We will provide a joint between both buyer and seller together who will be registered on our web site and manage this connection between them. Buyers would be able to search for goods and services they want. Sellers and buyers would be able to rate each other for the quality of service they provided.

This application is important to the people because it simplify much efforts of searching about any requirement in the city shops and shows and will change the searching method from physically to electronically.

To reach the goal I should achieve the following objectives:

- 1) make analysis of web applications platforms;
- 2) make an analysis of classifieds web applications;
- 3) develop a use-case model for classifieds web application;
- 4) develop an architecture for classifieds web application;
- 5) implement and test classifieds web application.

# 1. RELATED WORKS

## 1.1. Web applications platforms

### 1.1.1. Python and Django platform

Python [1] is a widely used general-purpose, high-level programming language. Its design philosophy emphasizes code readability, and its syntax allows programmers to express concepts in fewer lines of code than would be possible in languages such as C++ or Java. The language provides constructs intended to enable clear programs on both a small and large scale.

Python supports multiple programming paradigms, including object-oriented, imperative and functional programming or procedural styles. It features a dynamic type system and automatic memory management and has a large and comprehensive standard library.

Python interpreters are available for installation on many operating systems, allowing Python code execution on a majority of systems. Using third-party tools, such as Py2exe [20] or Pyinstaller [21], Python code can be packaged into stand-alone executable programs for some of the most popular operating systems, allowing for the distribution of Python-based software for use on those environments without requiring the installation of a Python interpreter.

CPython, the reference implementation of Python, is free and open-source software and has a community-based development model, as do nearly all of its alternative implementations. CPython is managed by the non-profit Python Software Foundation [1].

Python offers many choices for web development:

- frameworks such as Django [4] and Pyramid [14];
- micro-frameworks such as Flask [2,3] and Bottle [22];
- advanced content management systems such as Plone [23] and Django CMS.

Python's standard library supports many Internet protocols:

- HTML and XML;
- JSON;

- e-mail processing;
- support for ftp, imap, and other internet protocols;
- easy-to-use socket interface.

Django [2] is a high-level Python Web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes care of much of the hassle of Web development, so you can focus on writing your app without needing to reinvent the wheel. It's free and open source.

### **1.1.2. ASP.NET platform**

ASP.NET [19] is an open-source server-side web application framework designed for web development to produce dynamic web pages. It was developed by Microsoft to allow programmers to build dynamic web sites, web applications and web services.

It was first released in January 2002 with version 1.0 of the .NET Framework [19], and is the successor to Microsoft's Active Server Pages (ASP) technology. ASP.NET is built on the Common Language Runtime (CLR), allowing programmers to write ASP.NET code using any supported .NET language.

ASP.NET is in the process of being re-implemented as a modern and modular web framework, together with other frameworks like Entity Framework. The new framework will make use of the new open-source .NET Compiler Platform (code-name "Roslyn") and be cross platform. ASP.NET MVC, ASP.NET Web API, and ASP.NET Web Pages (a platform using only Razor pages) will merge into a unified MVC 6. The project is called ASP.NET vNext.

ASP.NET Web pages, known officially as Web Forms, are the main building blocks for application development in ASP.NET. There are two basic methodologies for Web Forms, a web application format and a web site format. Web applications need to be compiled before deployment, while web sites structures allows the user to copy the files directly to the server without prior compilation. Web forms are contained in files with ".aspx" extension; these files typically contain static (X) HTML markup or component markup.

The component markup can include server-side Web Controls and User Controls that have been defined in the framework or the web page.

Microsoft recommends dealing with dynamic program code by using the code-behind model, which places this code in a separate file or in a specially designated script tag.

Code-behind files typically have names like *MyPage.aspx.cs* or *MyPage.aspx.vb* while the page file is *MyPage.aspx* (same filename as the page file (ASPX), but with the final extension denoting the page language). This practice is automatic in Visual Studio and other IDEs however the user can change the code-behind page. Also, in the web application format, the page name.aspx.cs is a partial class that is linked to the page name.designer.cs file. The designer file is a file that is auto generated from the aspx page that allows the programmer to reference components in the aspx page from the cs page without having to declare them manually as in versions prior to ASP.Net version 2. When using this style of programming, the developer writes code to respond to different events, like the page being loaded, or a control being clicked, rather than a procedural walkthrough of the document.

ASP.NET's code-behind model marks a departure from Classic ASP in that it encourages developers to build applications with separation of presentation and content in mind. In theory, this would allow a Web designer, for example, to focus on the design markup with less potential for disturbing the programming code that drives it. This is similar to the separation of the controller from the view in model-view-controller (MVC) frameworks.

### **1.1.3. Microsoft Azure platform**

Microsoft Azure [5] is a cloud computing platform and infrastructure, created by Microsoft, for building, deploying and managing applications and services through a global network of Microsoft-managed datacenters. It provides both PaaS and IaaS services and supports many different programming languages, tools and frameworks, including both Microsoft-specific and third-party software and systems. Azure was released on 1 February 2010.

Microsoft Azure uses a specialized operating system, called Microsoft Azure, to run its "fabric layer": a cluster hosted at Microsoft's data centers that manages computing and storage resources of the computers and provisions the resources (or a subset of them) to applications running on top of Microsoft Azure. Microsoft Azure has been described as a "cloud layer" on top of a number of Windows Server systems, which use Windows Server 2008 and a customized version of Hyper-V, known as the Microsoft Azure Hypervisor to provide virtualization of services.

Scaling and reliability are controlled by the Microsoft Azure Fabric Controller so the services and environment do not crash if one of the servers crashes within the Microsoft data center and provides the management of the user's web application like memory resources and load balancing.

Azure provides an API built on REST, HTTP, and XML that allows a developer to interact with the services provided by Microsoft Azure. Microsoft also provides a client-side managed class library which encapsulates the functions of interacting with the services. It also integrates with Microsoft Visual Studio, Git, and Eclipse.

## 2. CLASSIFIEDS WEB APPLICATIONS REVIEW

### 2.1. Mredy

Mredy [6] is the most popular advertising website in Iraq (see fig. 1).

On the main page of this website you can choose one of the most common categories of advertisements:

- cars;
- real estate;
- furniture;
- electronics;
- clothing;
- others.

The user can choose a city from the title page to go to the list of advertisements.

When the user click on the advertisement he sees the details about the item description and it's price and he can also contact the advertiser through the registration and login to this website.

The advertiser also must register himself in the web in order to upload and show his items to the other users and clients, but in this site there is no ability to pay and shopping online.

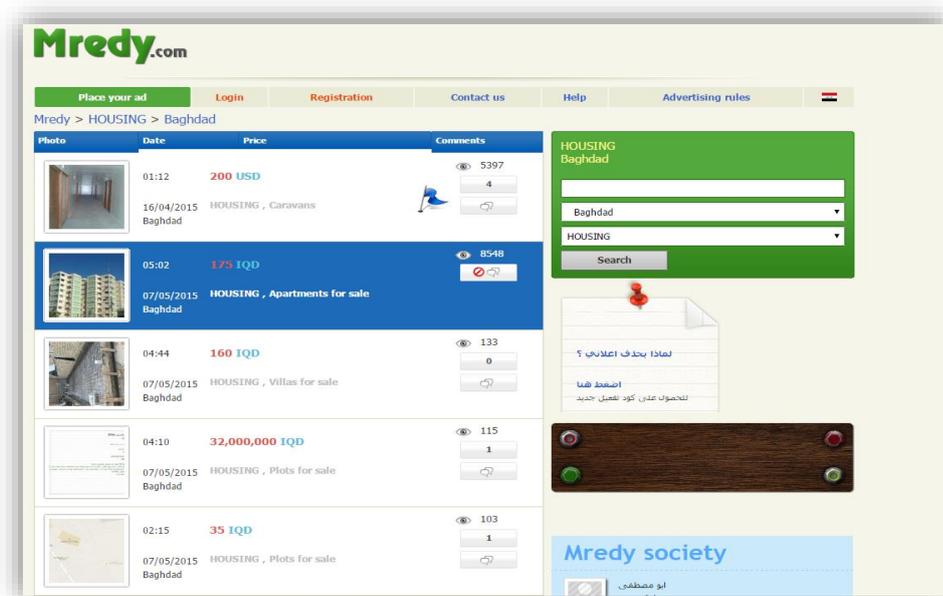


Fig. 1. Web interface of Mredy web site (translated to English)

## 2.2. Avito

Avito [7] is the most popular advertising site in Russia (see fig. 2).

On the main page of the Avito.com the user can choose one of the Russian cities and your preferable category of advertisements:

- transport;
- work;
- real estate;
- work;
- services;
- paraphernalia;
- home and garden;
- consumer electronics;
- hobbies and leisure;
- animals;
- for business.

When the user clicks on the suitable advertisement he sees the full description and the price of the advertised item.

The visitor or client can contact the advertiser or rate him as a favorite. Also he can view the phone number of the advertiser or send him a message, but he must register and login on this website.

This website is specialized for advertising only, and with shopping online availability.



Fig. 2. Web interface of Avito web site (translated to English)

## 2.3. Ebay

Ebay [15] is the most popular shopping website in Europe and many other countries around the world (see fig. 3).

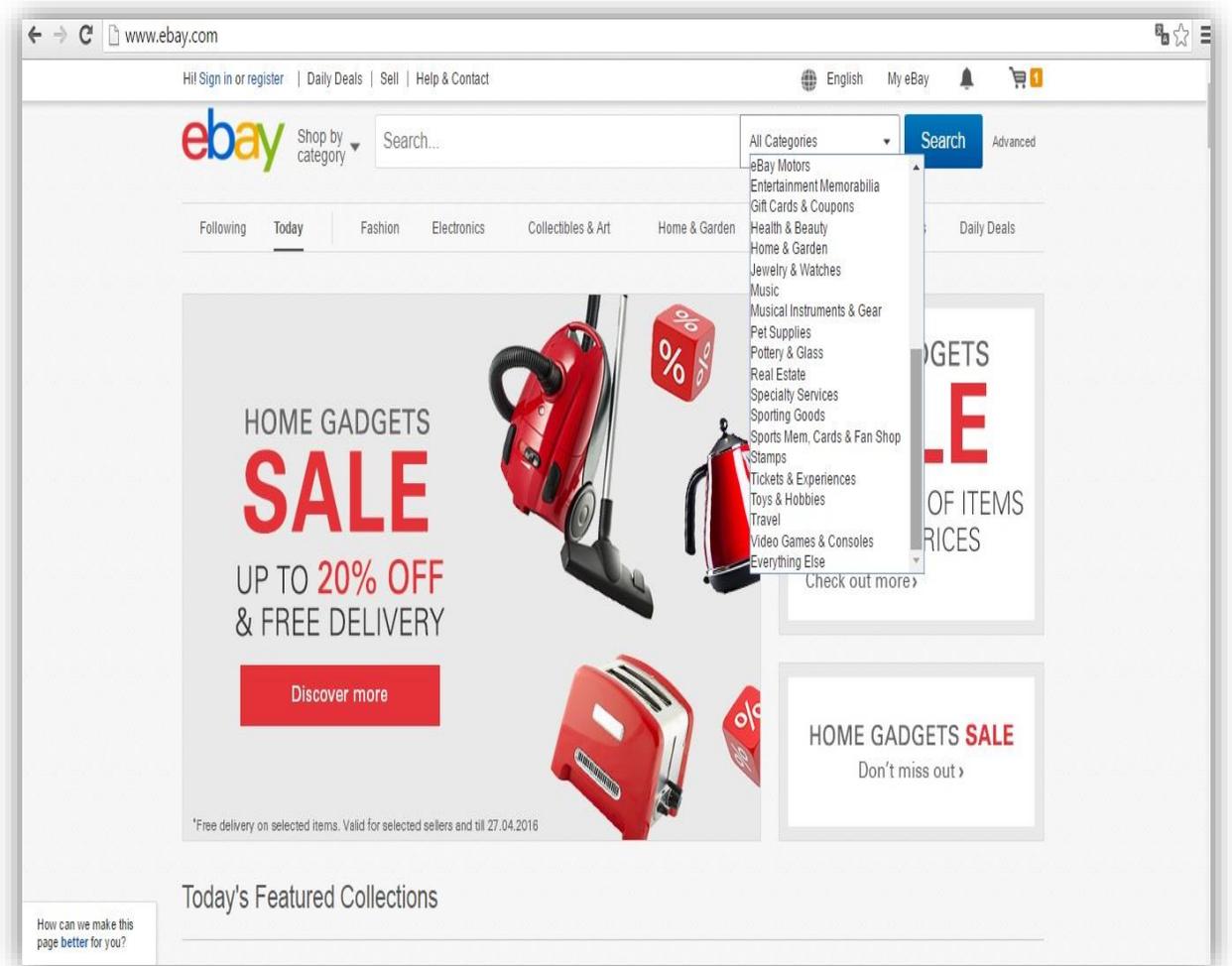
On the main page of this site, the visitor can view a huge list of different categories and choose one of them.

Any visitor can view the preferable item and it's full description and price.

The advertiser also must register on this site in order to view his items to the clients.

The client can make his shopping directly online after registration and login process.

The client can shop can buy many items together at the same time through the Shopping Cart.



**Fig. 3.** Web interface of Ebay web site (translated to English)

## 2.4. Amazon

Is a very well-known shopping and advertising website in America and many a lot of countries (see fig. 4).

When we open the main page of this site, we see the logo on the top of left side of and the navigation category list is on the middle top of the window.

Also, there are some visible advertised products appears temporarily and sequentially on the page because, they are much visited by the users.

There is a link for registering or signing in to this website, in order to be distinguished in the administration database if the user liked to buy something online.

The departments (categories) list is very huge and contains sub departments also that enable the user to sail in them and choose his desired product that is completely described.

In case that the user chose any product, he can add this product to his own shopping cart on this website and buy it by following the steps of payment. Also he can delete it or add another items for the shopping cart.

Any user can advertise about his goods but, he must firstly register and sign in to his account.

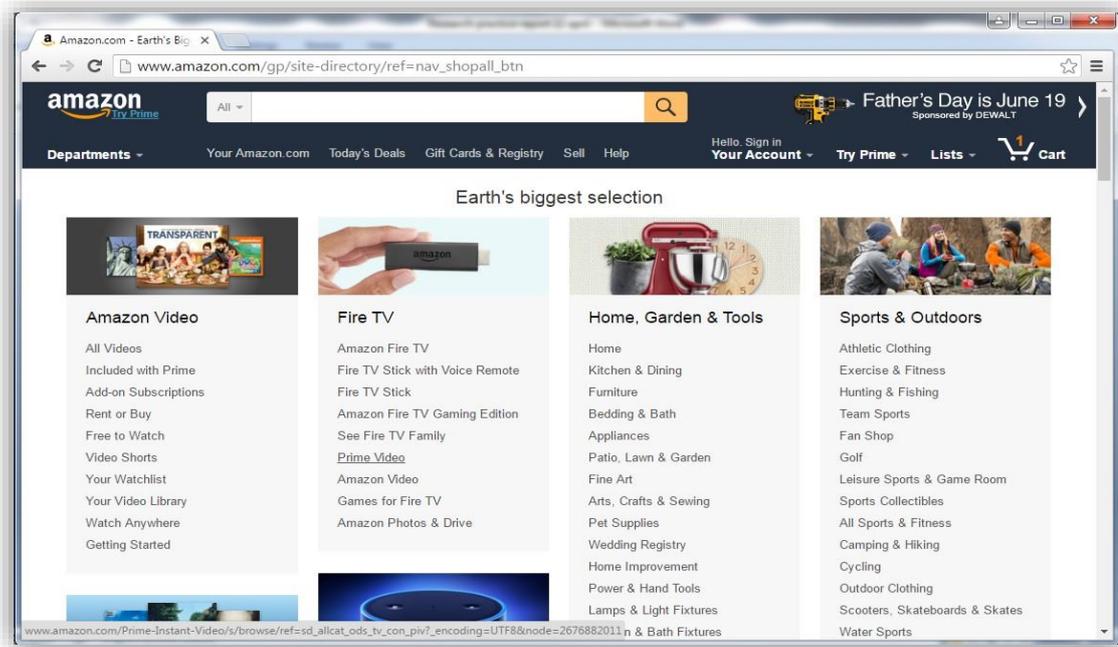
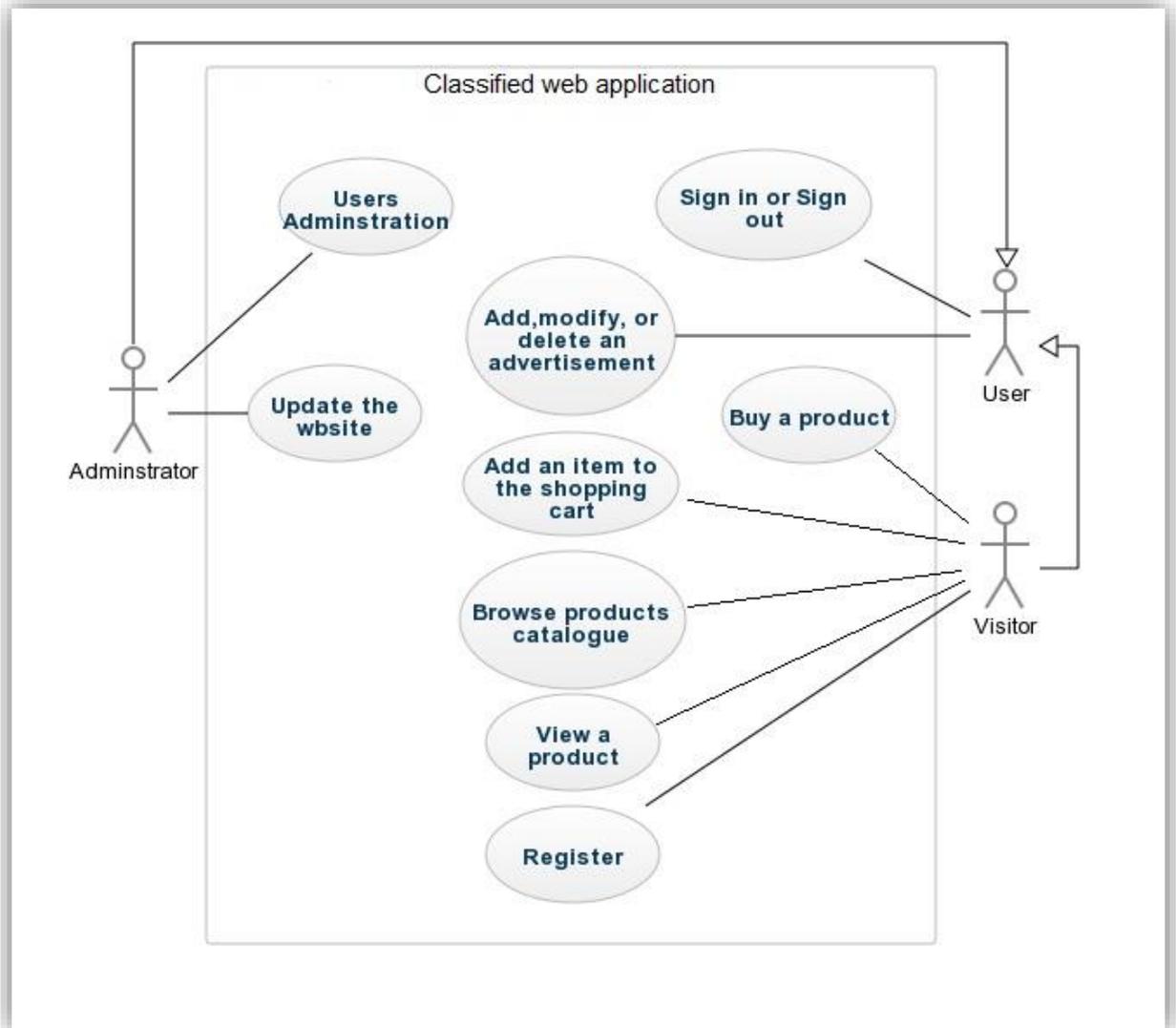


Fig. 4. Web interface of Amazon web site

### 3. REQUIREMENTS DEVELOPMENT

We created a Use Case diagram for the classifieds web application as shown in (fig. 5).



**Fig. 5.** Use Case diagram for the classifieds web application

We defined the following main actors:

- Administrator - a person who will administrate the website and manage all the services and users in this website application;
- User - a registered visitor who will add or delete his advertisement and his details on our website. He can also buy online.
- Visitor - a person who will search and visit our site.

We defined the following main use-cases for our system.

- Register - Any visitor on our website can register himself in order to benefit from the services on this website.

- Sign in or Sign out-any registered user of our web application should Sign In before adding any advertisement or buying any product online. He needs to provide his name, e-mail and choose his password. If he is registered on our web site already, he need to Sign In using his e-mail and password.

- Users Administration- administrator can manage all the registered

- Update the Web Site - administrator can update the web site frequently and periodically as he desire. He also can add more services to +

- Add, Modify, Delete Advertisement - advertiser can add, modify, or delete his advertisement.

- Browse products catalog - Any visitor can browse the full catalog of products or browse them by viewing them in their specialized categories.

- View Advertisement - any customer can view any advertisement and see it details, including description, specifications and pictures of the item.

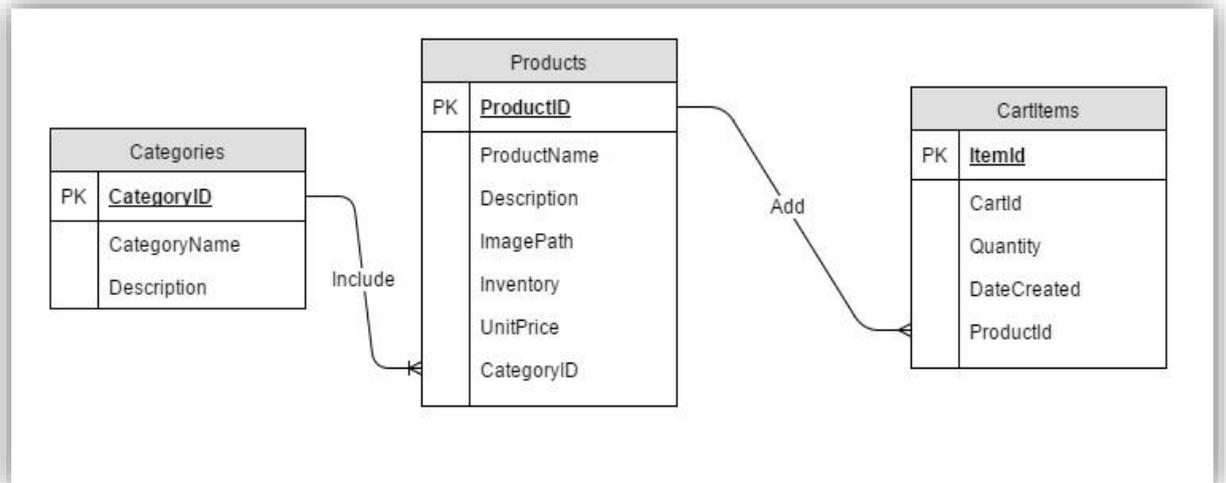
- Add an item to the shopping cart - Any visitor or user can add his required item of the advertised product to his own shopping cart.

- Buy a product - The user registered can buy his product by after adding the desired product to his own shopping cart and following the payment steps.

## 4. SYSTEM ARCHITECTURE AND IMPLEMENTATION

### 4.1. ER-Diagram

The Entity Relation Diagram represents the technique and data connections in our application database. So it identifies entities, determine the relationship between those entities, and analyzes the nature of those relationship (see fig. 6).



**Fig. 6.** ER-Diagram

#### 4.1.2. Database Structure

In our application database, we have three main classes (Products, categories, and CartItems) (see fig. 7).

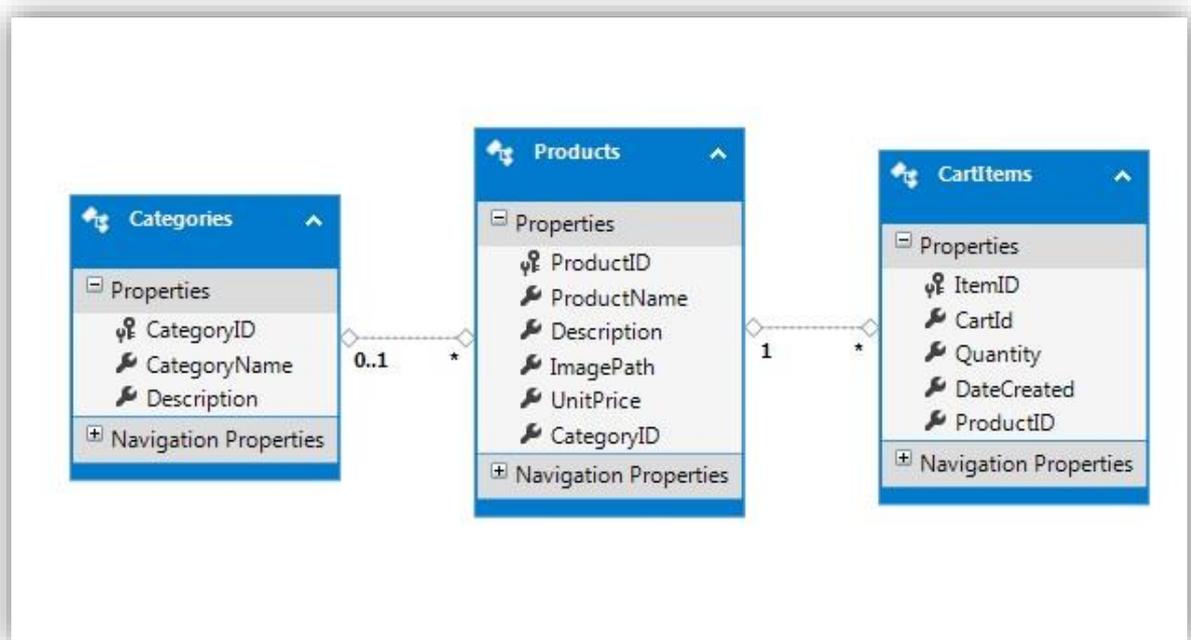
1) Categories - an entity class contains the information about each category as the following:

- CategoryID - a public key that represent a unique number for each category that some products belong to;
- CategoryName - represents the name of each category;
- Description - represents the description of the related product.

2) Products - an entity class contains information about each product as the following:

- ProductID - a public key that represents a unique number for each product.
- ProductName - represents the name of each product;
- Description - represents the description of each product;

- ImagePath - represents a link path of each product;
  - UnitPrice - represents the unit price for each product;
  - categoryID - represents the category number that the product belongs to.
- 3) CartItems - class represents the information about the cart actions of each user.
- ItemID - represents the ID number for each product in the cart;
  - CartID - represents the ID for each user;
  - Quantity - represents the quantity of each product inside the shopping cart;
  - DateCreated - represents the date of adding or removing products to the shopping cart;
  - ProductID - represents the ID number for each product.



**Fig. 7.** Database structure

## 4.2. Web shop-Application development Database

### 4.2.1. Product class

The products class contains the definitions for each product, and represented by the following members (see fig. 8):

- ProductID;
- productName;

- Description;
- ImagePath;
- UnitPrice;
- CategoryID.

It represents the individual products in the database.

```
using System.ComponentModel.DataAnnotations;

namespace HarajSooq.Models
{
    public class Product
    {
        [ScaffoldColumn(false)]
        public int ProductID { get; set; }

        [Required, StringLength(100), Display(Name = "Name")]
        public string ProductName { get; set; }

        [Required, StringLength(10000), Display(Name = "Product Description"),
        DataType(DataType.MultilineText)]
        public string Description { get; set; }

        public string ImagePath { get; set; }

        public int Inventory { get; set; }

        [Display(Name = "Price")]
        public double? UnitPrice { get; set; }

        public int? CategoryID { get; set; }

        public virtual Category Category { get; set; }
    }
}
```

**Fig. 8.** Product class code

#### 4.2.2. Category Class

Category class contains the definitions of each category that any products is belonged to it, and represented by the following members (see fig.9):

- CategoryID;
- CategoryName;
- Description;
- Products.

This class defines the type of the product, which the application is designed through one of the categories (Cars, Electronics and Computers, Furniture, Real estate, Clothes and Shoes, and Others).

```
16 references
public class Category
{
    [ScaffoldColumn(false)]
    6 references
    public int CategoryID { get; set; }

    [Required, StringLength(100), Display(Name = "Name")]
    9 references
    public string CategoryName { get; set; }

    [Display(Name = "Product Description")]
    0 references
    public string Description { get; set; }

    0 references
    public virtual ICollection<Product> Products { get; set; }
}
```

**Fig. 9.** Category class

#### 4.2.3. Database\_INITIALIZER class

We created a class in the (Models) folders as (ProductDatabaseInitializer.cs).

We already created this class before running the application because, the (Application\_Start) method that contained in the (global.asax.cs) file will call this class and it will use the model classes (Category and Product) to create the database.

This class initializes the Database and allows seed data to be added to the database, so, we can directly display the products and categories (see fig. 10, 11).

```
new Category
{
    CategoryID = 1,
    CategoryName = "Cars"
},
new Category
{
    CategoryID = 2,
    CategoryName = "Electronics and Computers"
},
```

**Fig. 10.** Category\_INITIALIZER class codes

```

new Product
{
    ProductID = 6,
    ProductName = "SAMSUNG LED Smart TV Full HD",
    Description = "50' LED Smart TV Full HD 1080p; Color:Black.",
    ImagePath="SamsungTV.jpg",
    UnitPrice = 1300.00,
    CategoryID = 2,
    Inventory = 1
},

```

**Fig. 11.** Products Initializer class codes

#### 4.2.4. Shopping Cart Actions

The *shoppingCart* class will be added to a separate folder in the application.

The (AddToCart) method enables individual products to be included in the shopping cart based on the product ID (see fig 12).

```

public class ShoppingCartActions : IDisposable
{
    12 references
    public string ShoppingCartId { get; set; }

    private ProductContext _db = new ProductContext();

    public const string CartSessionKey = "CartId";

    1 reference
    public void AddToCart(int id)
    {
        // Retrieve the product from the database.
        ShoppingCartId = GetCartId();
    }
}

```

**Fig. 12.** some codes of shopping cart class

#### 4.3. SQL Codes

Inside the (Web.config), the value of the *DataDirectory* is a reserved value and represents the (App\_Data) folder in our project.

This folder where the database that was created from our entity classes is located.

We used 978 codes in this system.

## 5. WEB SHOP IMPLEMENTATION AND TEST

### 5.1. Main Page

When we run the application at the first time, the browser opens and shows the *Default.aspx* page, and the main page will be displayed (see fig. 13):

– The category navigation The database that named (HarajSooq.mdf) is created in the *App\_Data* folder, for this reason we see the category navigation that is generated as a result of retrieving the categories from the database;

In addition to the categories of products, we see the following specifications:

– Logo and Title of our website– are compiled from *Site.Master* page of the application. They represent the main Title of our website;

– Welcome Statement –is compiled from the *Default.aspx* page. It represents a short statement of welcome to the visitors of the site.

The navigation bar, which appears on the top of main page, also keeps appearing on the sub-windows of the website, because it contains the following important links:

- Home;
- About;
- Contact;
- Products;
- Cart;
- Register;
- Log in.

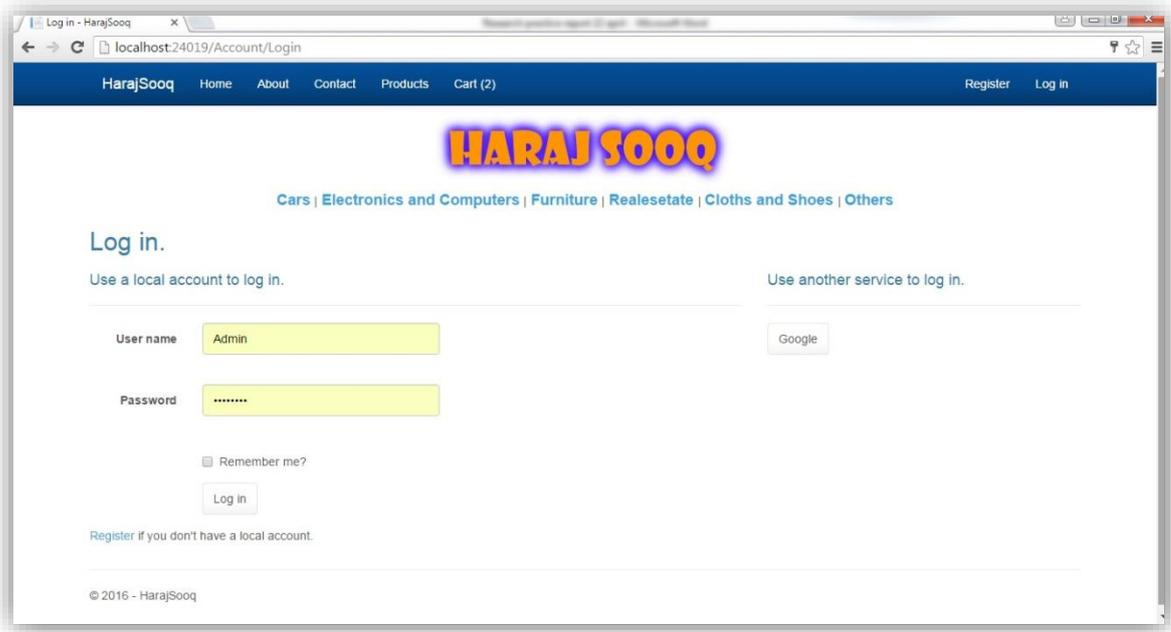


**Fig. 13.** Main page

## 5.2. Login Page

At the right side on the top of each page of our website, we see a button named as (Login) (see fig. 14).

When we click on this button, the (Login.aspx) page will be displayed (see fig. 9)



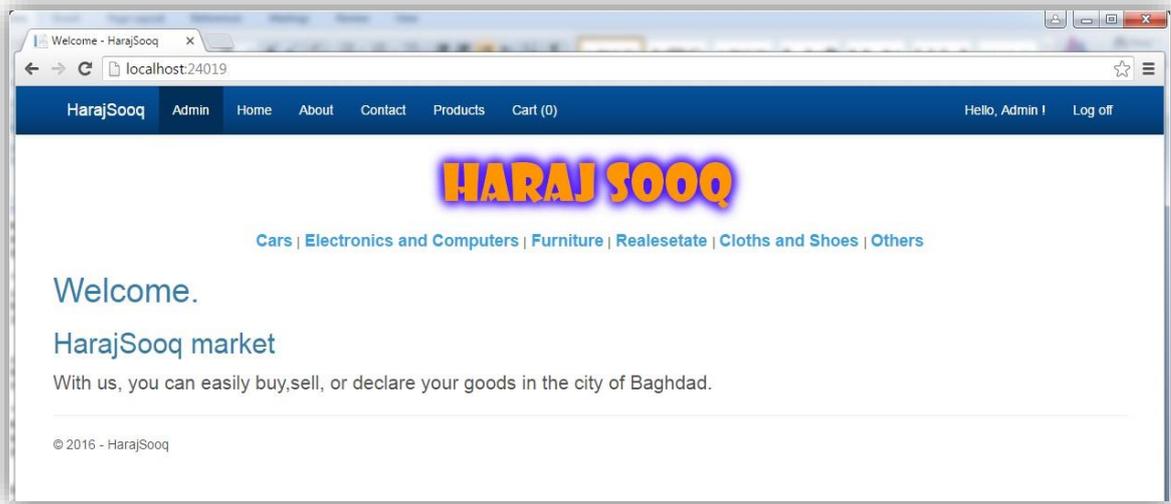
**Fig. 14.** Output of login page

In the Login page we see two fields, the first one is for entering the (User name), and the second one is for entering the (Password).

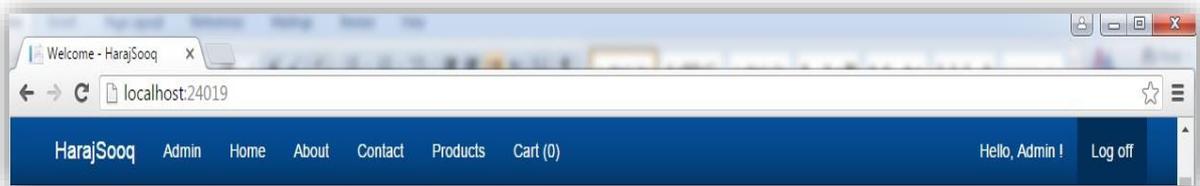
The logging in process can proceed successfully after the checking the validation of the previously registered user name and password.

Once the user logged into the site, the welcome page will appear again, but under the environment of the specific user, so, we can log in as an (Admin), and navigate to the (AdminPage.aspx) (see fig. 15). There are more abilities to the registered user than the usual unregistered visitor, because he can buy online, or declare about his goods in the suitable categories, also his own data will be stored in the server, in order be distinguished from the other users.

After finishing the process of shopping or declaring, the user can log off of his account through the same button on the navigation bar (see fig. 16).



**Fig. 15.** Admin page

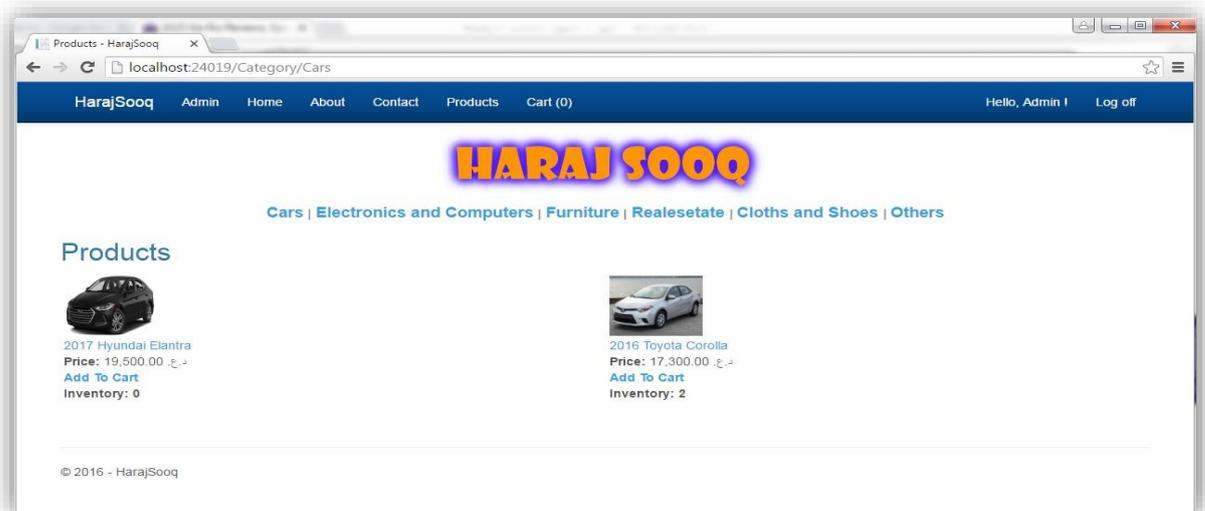


**Fig. 16.** Logging off

### 5.3. Catalog page

When we run the application and main page is displayed, we see the product category navigation menu. We can choose any category of categories menu.

The (ProductList.aspx) page is displayed showing only the products that included this category (see fig.17).



**Fig. 17.** Cars category and its products

Also we can display the products of any other categories (see fig.18).



Fig. 18. Cars category and its products

At the navigation bar, we can select (products) link to view all the entire products of all categories, and the (ProductList.aspx) is displayed again (see fig. 19).

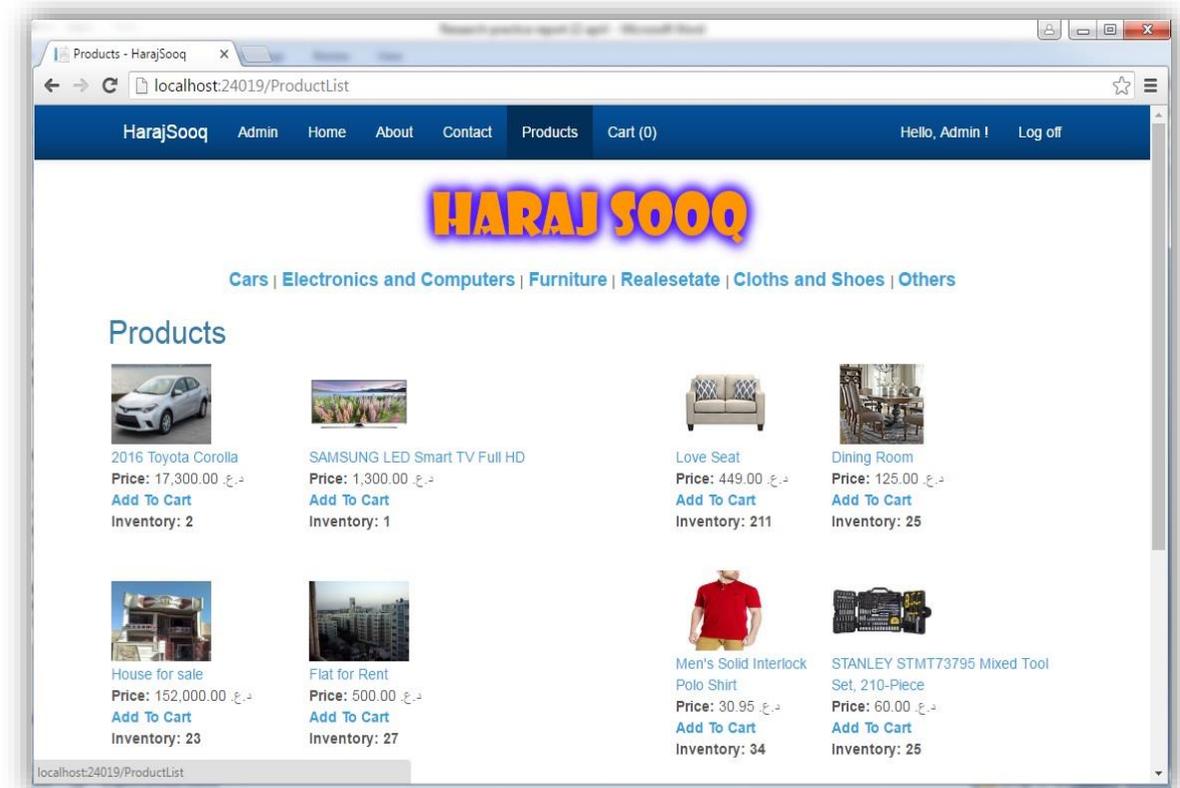
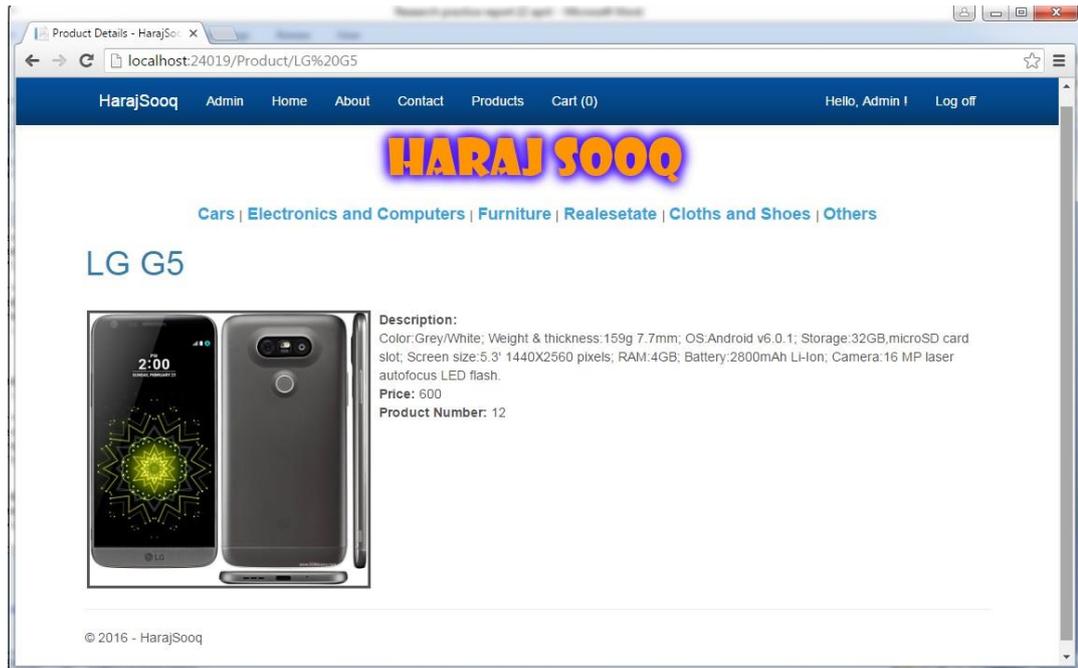


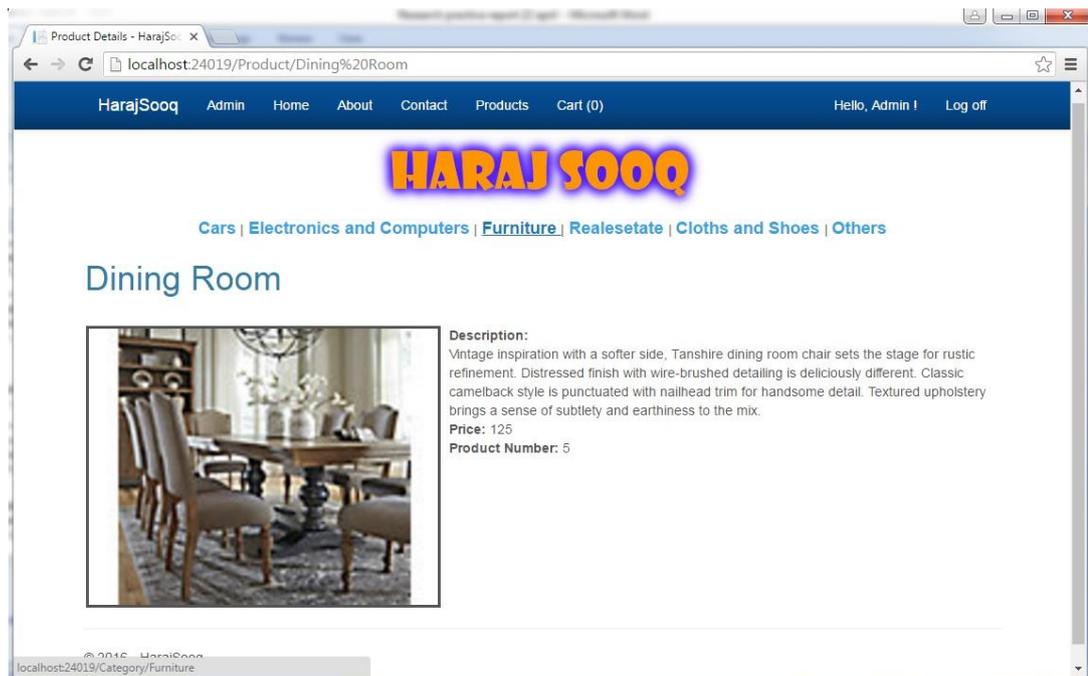
Fig. 19. Products page

## 5.4. Item description

When we open a specific category from the category navigation menu, and the (ProductList.aspx) page is displayed, then we select a distinct product from the category's products, the (ProductsDetails.aspx) page is displayed, and the description of this product will appear (see fig. 20, 21).



**Fig.20.** Product description



**Fig.21.** Product description

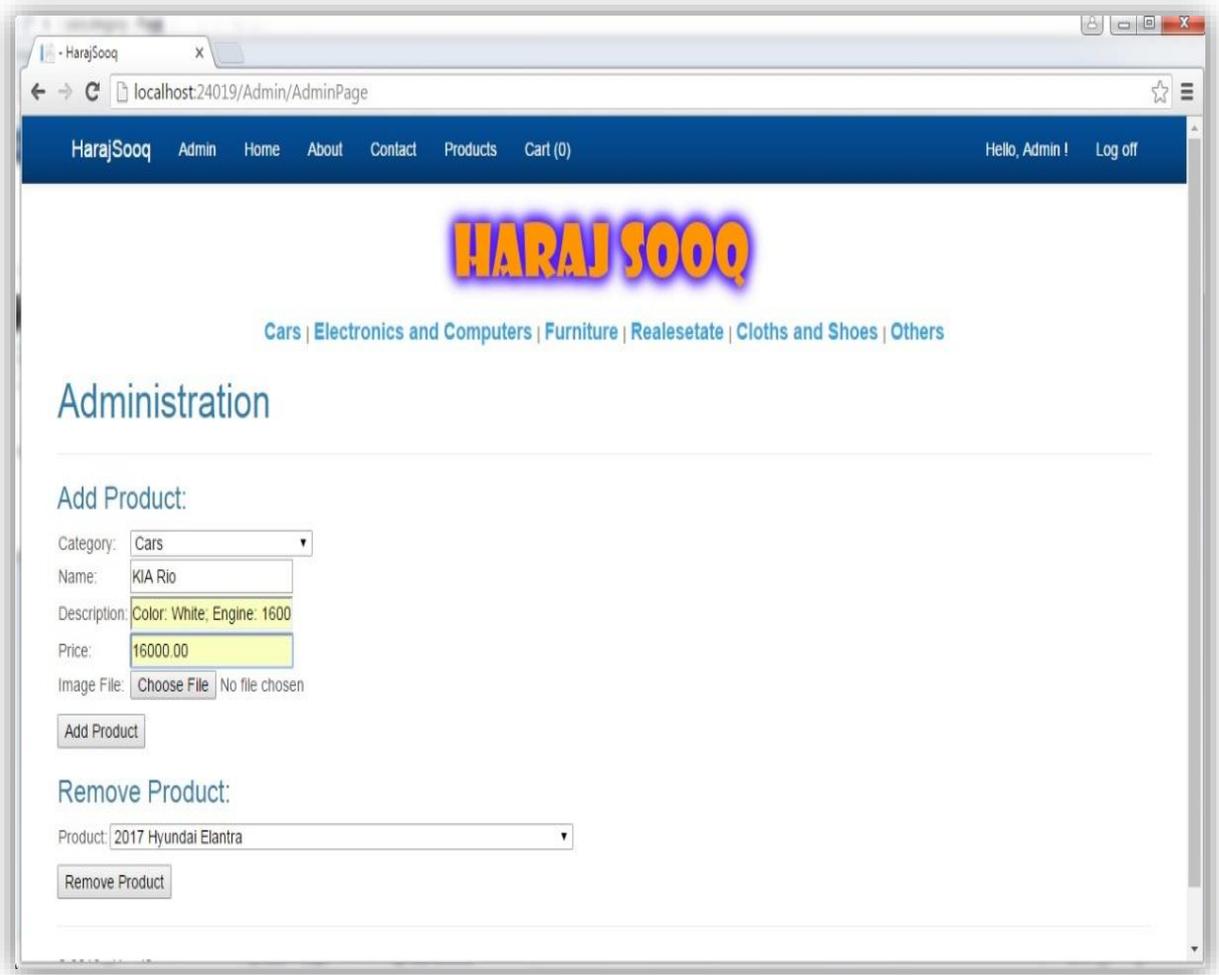
## 5.5. Add or remove items in the products' list

When we logged in the Admin page, and select the (Admin) link at the left side of the navigation bar, we will navigate to the page of (AdminPage.aspx), then, the Administration page will appear.

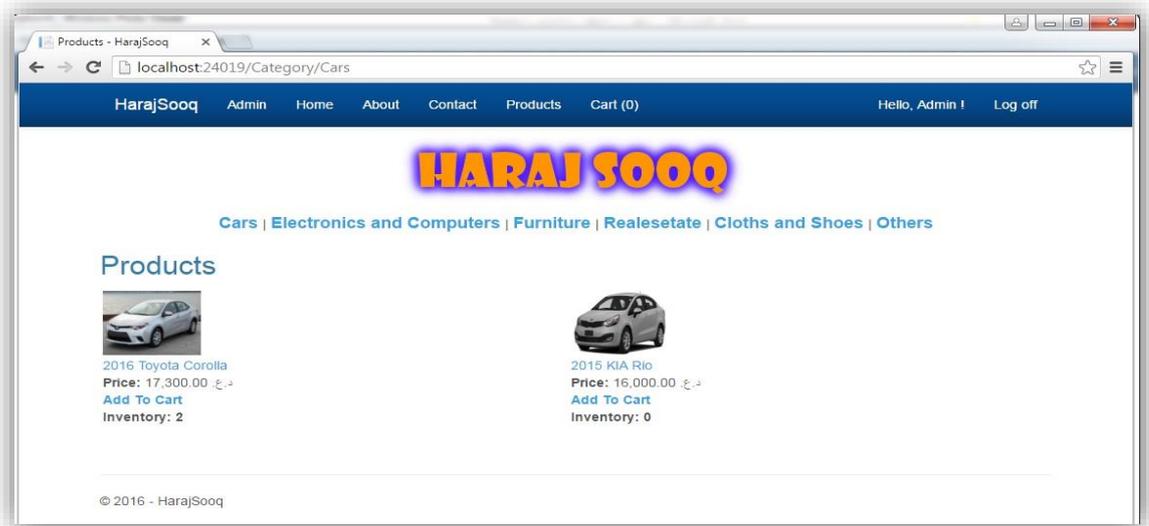
There are two main partitions on this page (see fig. 22):

- Add Products– used for adding new products by determining the suitable (Category, Name, Description, Price, and Image File);
- Remove Products – used for removing products from the field of products list.

After adding or removing some products, the related categories will be modified according to our updates (see fig. 23).



**Fig. 22.** Add and remove page

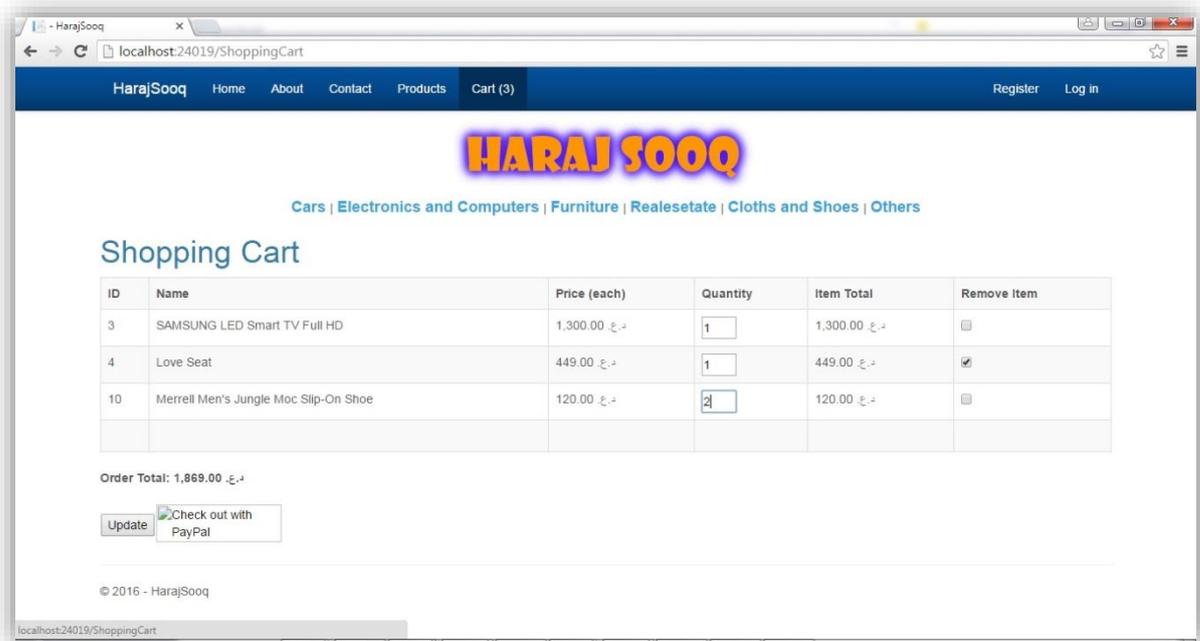


**Fig. 23.** Updates after adding or removing products

### 5.6. Add or remove items in the shopping cart

We already added a counter as a link to the shopping cart in the Site.Master page. So, when we open any category and select any product from this category, we can click on the **Add To Cart** button that located under each product. At once the *ShoppingCart.aspx* will be displayed (see fig. 24).

We can determine the quantity of each selected item, or remove the item from the shopping cart.



**Fig. 24.** Add, remove or update items in the shopping cart

## CONCLUSION

In this thesis, the development of classifieds web application for the city of Baghdad was proposed to help the people who live in Baghdad city who suffer from the difficulties of searching about their requirements from the markets because, the advertiser or the seller cannot inform the far people about his goods, and the searcher also cannot reach to all sellers.

All of the similar websites in this city does not provide the online payments, and this also considered as a big problem, so that, a deep analysis is done to the sites that is specialized for thesis services in many other countries and cities.

After analyzing of all the givens. The best and simplest idea was to create this web application by using the asp.net environment with C# programming language because they are familiar.

The application is implemented by the specialized classes and databases that were constructed according to the requirements.

The application was tested many times and very time of testing we found new extra requirements from the users, then we made it finally suitable to all because, it deals with the people who want to buy through the electronic payment.

The achievement of the website can briefly expressed as Real and comfortable join between the advertiser and the seller, activate the option of online shopping by designing a payment methodology throughout the applications that used to build it.

In this work we provide an overview of development and implementation of web application on the basis of the ASP.NET programming environment. During my work the following issues was solved:

- 1) current classified web applications was investigated;
- 2) ASP.NET frame work was learned;
- 3) architecture of the classified web application was developed;
- 4) the classified web application based on ASP.NET frame work was implemented.

## REFERENCES

1. Python.org web site [Electronic resource] URL: <https://www.python.org/> (date of visit 30.04.2015).
2. Flask: a Python micro framework web site [Electronic resource] URL: <http://flask.pocoo.org/> (date of access 15.04.2016).
3. Grinberg M. Flask Web Development: Developing Web Applications with Python. USA: O'Reilly, 2014. 258 p.
4. Django web site [Electronic resource] URL: <https://www.djangoproject.com/> (date of visit 30.04.2015).
5. "Windows Azure General Availability". The Official Microsoft Blog. Microsoft. [Electronic resource] URL: <http://blogs.microsoft.com/blog/2010/02/01/windows-azure-general-availability/> (date of visit 30.04.2015).
6. Mredy web-shop [Electronic resource] URL: [www.mredy.com/](http://www.mredy.com/) (date of access 01.05.2015).
7. Avito web-shop [Electronic resource] URL: <https://www.avito.ru/> (date of access 01.05.2015).
8. MacDonald, Matthew; Szpuszta, Mario (2005). Pro ASP.NET 2.0 in C# 2005 (1st ed.). Apress. ISBN 1-59059-496-7
9. Anne Boehm: Murach's ASP.NET 3.5 Web Programming with VB 2008, July 21, 2008, Mike Murach and Associates, ISBN 978-1-890774-47-9
10. Stephen Walther: ASP.NET 3.5 Unleashed, December 28, 2007, Sams Publishing, ISBN 0-672-33011-3 ISBN 0-672-33011-3
11. Stephen Walther: Data Access in the ASP.NET 2.0 Framework (Video Training), September 26, 2007, Sams Publishing, ISBN 0-672-32952-2
12. Israel B. Ocbina: Mastering VB.NET and C#. 7th Edition. October 22, 2004. by Cyberocbina© Café. A .NET Developers Edition
13. Getting Started with ASP.NET 4.5 Web Forms and Visual Studio 2013. [Electronic resource] URL: <https://www.asp.net/web->

forms/overview/getting-started/getting-started-with-aspnet-45-web-forms/introduction-and-overview (date of access 18.10.2015).

14. Pyramid.org web site [Electronic resource] URL: <http://www.pyllonsproject.org/>(date of access 15.04.2015).

15. Ebay web-shop [Electronic resource] URL: <http://www.ebay.com/> (date of access 16.04.2015).

16. Amazon web-shop [Electronic resource] URL: <http://www.amazon.com/> (date of access 16.04.2015).

17. Laudon, Kenneth C.; Guercio Traver, Carol (2014). E-commerce. business. technology. society. 10th edition. Pearson. ISBN 978-013-302444-9.

18. Chaudhury, Abijit; Kuilboer, Jean-Pierre (2002). e-Business and e-Commerce Infrastructure. McGraw-Hill. ISBN 0-07-247875-6.

19. Asp.net web site [Electronic resource] URL: <https://www.asp.net/> (date of access 18.04.2016).

20. Py2exe.org web site [Electronic resource] URL: [https://www.py2exe.org](https://www.py2exe.org/) / (date of access 18.04.2015).

21. Pyinstaller.org web site [Electronic resource] URL: <https://www.pyinstaller.org/> (date of access 18.04.2016).

22. Python web frame work [Electronic resource] URL: <http://bottlepy.org/docs/dev/index.html>(date of access 18.04.2016).

23. Plone.org [Electronic resource] URL: <https://www.plone.org/> (date of access 18.04.2016).