Syrian Private University
Faculty of Informatics & Computer Engineering

ELECTRONIC COMMERCE WEBSITE WITH COLLABORATIVE FILTERING TO MAKE RECOMMENDATION SYSTEM.

Junior Project Presented to the Faculty of Computer and Informatics Engineering

In Partial Fulfillment of the Requirements for the Degree of Bachelor of Engineering

In Software & Information Systems.

Prepared By

Abdul Kareem Eisa Balta

Mohammad Ismaeel Al-Tahan

Hanadi Ahmad Hmaideh

Samer Ahmad Al-Yaseen

Under The Supervision Of

Dr. Basem Qusaibah

Eng. MHD Mousa Hamad

Table of contents:

LIST OF TABLES:

LIST OF FIGURES:

Abstract

The invention of the World Wide Web made it possible to buy everything online without going to stores or malls, the main goal of the project is to simulate an online store website that sells electronic devices and its accessories, the website is able to be developed to sell more and more things.

The main characteristic of the project is: the website using some algorithms for rating, and to predicate items that may be purchased by each customer, depending on the previous procurement cases.

This project was built using ASP.NET MVC 4 that make the working with MVC technique simple to make a Dynamic website, and there is some algorithms used for making Recommendation system that gives each customer a different list of recommended product for every customer logs in to the website.

ASP.Net MVC 4 is developed in Microsoft Corporation to use MVC technique in the websites, the used algorithms is the same algorithms used in AMAZON.com but we used Pearson Correlation Score to calculate the similarity between products, AMAZON.com used Cosine Based method to calculate similarity.

Chapter 1: Electronic Commerce

1.1: WHAT IS ELECTRONIC COMMERCE?

Even today, some considerable time after the so called ‘dot com/Internet revolution’, electronic commerce (e-commerce) remains a relatively new, emerging and constantly changing area of business management and information technology. There has been and continues to be much publicity and discussion about e-commerce. Library catalogues and shelves are filled with books and articles on the subject. However, there remains a sense of confusion, suspicion and misunderstanding surrounding the area, which has been exacerbated by the different contexts in which electronic commerce is used, coupled with the myriad related buzzwords and acronyms.

Electronic commerce has existed for over 40 years, originating from the electronic transmission of messages during the Berlin airlift in 1948. From this, electronic data interchange (EDI) was the next stage of e-commerce development. In the 1960s a cooperative effort between industry groups produced a first attempt at common electronic data formats. The formats, however, were only for purchasing, transportation and finance data, and were used primarily for intra-industry transactions.

It was not until the late 1970s that work began for national Electronic Data Interchange (EDI) standards, which developed well into the early 1990s.

With the advent of the Internet, the term e-commerce began to include:

1- Electronic trading of physical goods and of intangibles such as information.
2- All the steps involved in trade, such as on-line marketing, ordering payment and support for delivery.
3- The electronic provision of services such as after sales support or on-line legal advice.
4- Electronic support for collaboration between companies such as collaborative on-line design and engineering or virtual business consultancy teams.

1.2: DEFINISIONS OF ELECTRONIC COMMERCE:

Some of the definitions of e-commerce often heard and found in publications and the media are:

1- Electronic Commerce (EC) is where business transactions take place via telecommunications networks, especially the Internet.

2- Electronic commerce describes the buying and selling of products, services, and information via computer networks including the Internet.
3- Electronic commerce is about doing business electronically.
4- E-commerce, ecommerce, or electronic commerce is defined as the conduct of a financial transaction by electronic means.

1.3: SUMMARY:

In the near future, E-commerce will become something essential in most countries of the world.
So that anyone can buy what he need from one place he sat in his house.
And this is the idea that our project talks about.

Chapter2: Collaborative Filtering

Ever browsed Amazon and noticed the "Customers That Bought This Item Also Bought These Items" list, or after you've selected a movie on Netflix, recommendations for similar flicks? The ability to find trends in the browsing habits and choices of users has become a must for many customer facing websites. Collaborative Filters are the magic that makes these features possible.

2.1: INTRODUCTION:

With the vast amount of data that the world has nowadays, institutions are looking for more and more accurate ways of using this data.
Companies like Amazon use their huge amounts of data to give recommendations for users. Based on similarities among items, systems can give predictions for a new item's rating. Recommender systems use the user, item, and ratings information to predict how other users will like a particular item.
Recommender systems are now pervasive and seek to make profit out of customers or successfully meet their needs. However, to reach this goal, systems need to parse a lot of data and collect information, sometimes from different resources, and predict how the user will like the product or item. The computation power needed is considerable.
This project uses the Item-Based collaborative filtering. And it is an approach of machine learning through:

1- Applying machine learning On-Line using Collaborative Filtering.
2- Parsing data retrieved from a database and predicting user preference.

What we were trying to do is to build a system that collects information and then uses the stored data in a machine learning algorithm. Predicting users' preferences using data may give more accurate results than any algorithm that does not use previous data. Most systems like Amazon, eBay, and others suggest things to users based on similarities among users, items, or both. This will make those systems more personalized and efficient from a user's perspective. Commercial and trading systems gain trust and profits using such systems if they successfully predict what users want at what time and where.
In the collaborative filtering algorithm, the system has a recorded set of items and users and how the users rated those items. Then algorithm is used to predict the rating for a user who has not rated the item yet. A rating for an item can be predicted from the ratings given to the item by users who are similar in taste to the given user.

2.2: AMAZON'S ITEM-TO-ITEM COLLABORATIVE FILTERING:

Amazon uses Item-to-item Collaborative Filtering. This method, according to Linden (2003) is scalable and produces high quality recommendations. Also, they provide a feature where you can see what is recommended to you and edit your filter of recommendations by product line and subject.
In addition they can access those products as well as their previous purchases to rate them.

Linden describes Amazon's method Item-to-item by taking the item that a user has rated highly then finding similar items. Then, the system will construct a matrix of similar items that users tend to rate highly. After that, the system will use this matrix to recommend new items for users. This process is done offline and the matrix is built by iterating through items and building a similarity table. The pseudo code is given as follows :(Linden et al., 2003)

```
For each item in product catalog, $I_1$
For each customer $C$ who purchased $I_1$
For each item $I_2$ purchased by customer $C$
Record that a customer purchased $I_1$ and $I_2$
For each item $I_2$ Compute the similarity between $I_1$ and $I_2$
```

2.2.1: Similarity between Items:

The similarity values between items are measured by observing **all the users who have rated both the items**.

As shown in the diagram below, the similarity between two items is dependent upon the ratings given to the items by users who have rated both of them:

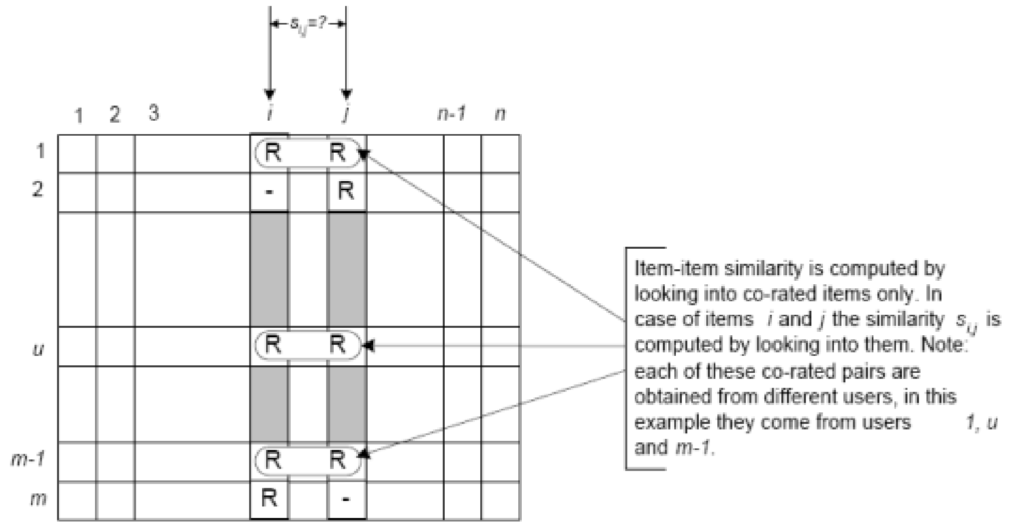

$s_{ij}=?$
1
2
3
i
j
n-1
n
1
2
u
m-1
m
R
-
Item-item similarity is computed by looking into co-rated items only. In case of items i and j the similarity s_{ij} is computed by looking into them. Note: each of these co-rated pairs are obtained from different users, in this example they come from users 1, u and m-1.

Figure 2.1 Similarity between items.

2.2.2: Similarity Measures:

There are many different mathematical formulations that can be used to calculate the similarity between two items. As can be seen in the formula below, each formula includes terms summed over the set of common users U. Here some of this formulas:

- Cosine-Based Similarity.
- Adjusted cosine similarity.
- Pearson (correlation)-based similarity.

2.3: PEARSON CORRELATION SCORE:

The algorithm we're going to be using to find the trends in our data is called the Pearson Correlation Score. The reason for this choice is simple it effectively can handle and balance un-normalized data.

The similarity between different items is measured as:

$$sim(I_a, I_b) = \frac{\Sigma_i^m (O_{ia} - \overline{O_i}) \times (O_{ib} - \overline{O_i})}{\sqrt{\Sigma_i^m (O_{ia} - \overline{O_i})^2} \sqrt{\Sigma_i^m (O_{ib} - \overline{O_i})^2}}$$

Where Oi is the average of the i-th user's ratings [1, 2].

The prediction on an item i for a user j can be computed by using the sum of the ratings of the user to items weighted by similarity between different items as:

$$O_{ij} = \frac{\Sigma_{c=1}^{k} O_{ic} \cdot sim(I_j, I_c)}{\Sigma_{c=1}^{k} sim(I_j, I_c)}$$

Where Ij identifies the j-th item Ic identifies nearest neighbors of the j-th item. Oij represents the i-th user's rating on the j-th item [1].

2.4: SUMMARY:

By using all the previous formulas (Item-to-Item Collaborative Filtering) and the Pearson (correlation) Score formulation to compute similarity between Items, our recommendation system is ready to give a recommended Items for every user in the website and this recommended list depends on the previous Items the customer have purchased.

CHAPTER3: PC STORE

3.1: PROJECT DOCUMENT:

Here we will talk about the Use Case Description, Database design and How to insert Data into the Database.

Use Case Diagram:

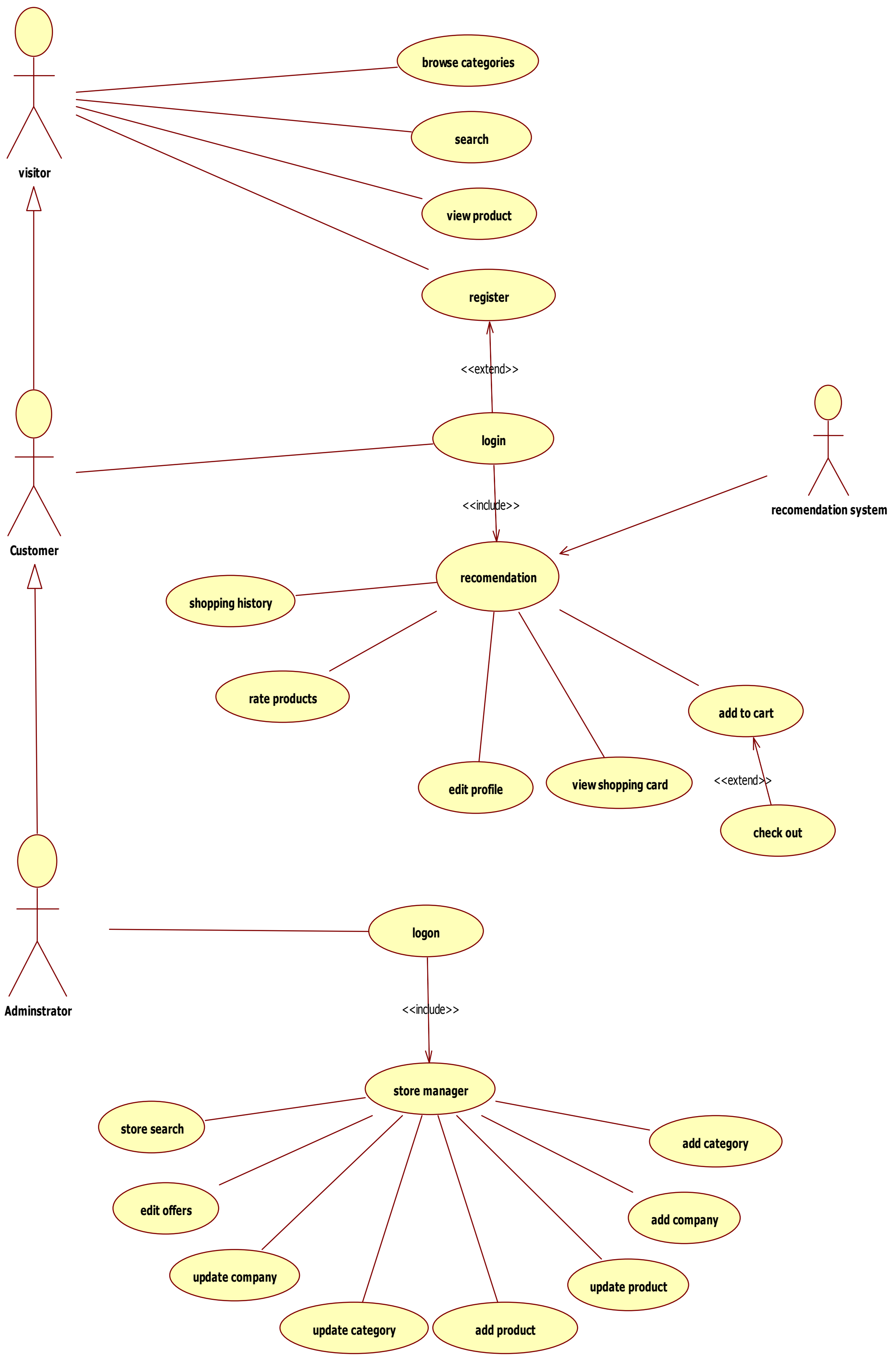


Figure 3.1 Use Case diagram.

3.1.1: Use Case Description:

Number	1
Name	Login.
Summary	The customer log as a customer.
Actors	Primary: visitor, customer. Secondary: none.
Pre-conditions	The visitor should be registered.
Scenario	When the visitor enter to the site he must enter a valid user name and valid password, if he is not in the website he must to register.
Exceptions	Internet connection is lost by either the user or the system.
Post-condition	None.

Table 3.1 Login.

Number	2
Name	Register.
Summary	Register in the website as customer.
Actors	Primary: visitor. Secondary: none.
Pre-conditions	Register a new customer.
Scenario	The visitor must full a valid form to register in the website.
Exceptions	Internet connection is lost by either the user or the system.
Post-condition	None.

Table 3.2 Register.

Number	3
Name	Browse categories.
Summary	The user can browse the categories in the website.
Actors	Primary: visitor, customer, administrator.

	Secondary: none.
Pre-conditions	None.
Scenario	The user can browse categories to find the item and add it to his cart if he wants to buy.
Exceptions	None.
Post-condition	None.

Table 3.3 Browse Categories.

Number	4
Name	Search
Summary	Search for product
Actors	Primary: visitor, customer, administrator. Secondary: none.
Pre-conditions	None
Scenario	Al the actors can search a keyword in the products using the search method.
Exceptions	None.
Post-condition	None.

Table 3.4 Search.

Number	5
Name	View product.
Summary	The website views the product.
Actors	Primary: visitor, customer, administrator. Secondary: none.
Pre-conditions	None.
Scenario	The website gives a description for a product to add it to the cart to check it out.
Exceptions	None.
Post-condition	None.

Table 3.5 View Product.

Number	6
Name	Recommendation.
Summary	Show the recommended list.
Actors	Primary: recommendation system. Secondary: none.
Pre-conditions	Login.

Scenario	When the customer logs in to the website the recommendation system starts to work and gives the customer a list of recommended products for each user.
Exceptions	Internet connection is lost by either the user or the system.
Post-condition	None.

Table 3.6 Recommendation.

Number	7
Name	Shopping history
Summary	Show shopping history.
Actors	Primary: customer. Secondary: none.
Pre-conditions	Login.
Scenario	The customer order to view the products who bought before.
Exceptions	Internet connection is lost by either the user or the system.
Post-condition	None.

Table 3.7 Shopping history.

Number	8
Name	Rate product.
Summary	Give a rate for product.
Actors	Primary: customer. Secondary: none.
Pre-conditions	Login, add to cart.
Scenario	When the customer wants to buy anything from the store he must give a rate for product.
Exceptions	Internet connection is lost by either the user or the system.
Post-condition	None.

Table 3.8 Rate product.

Number	9
Name	Edit profile.
Summary	Change profile data.
Actors	Primary: customer. Secondary: none.
Pre-conditions	Login, registered.
Scenario	The customer can access to his data ad can modify it (name, password, city, country, payment methods, E-mail.).
Exceptions	Internet connection is lost by either the user or the system.
Post-condition	Profile data changed.

Table 3.9 Edit profile.

Number	10
Name	View shopping cart.
Summary	View the items that added to cart.
Actors	Primary: customer, visitor. Secondary: none.
Pre-conditions	None.
Scenario	The user can shows the added products to the basket and take a detailed report about the items and prices.
Exceptions	Internet connection is lost by either the user or the system.
Post-condition	None.

Table 3.10 View shopping cart.

Number	11
Name	Add to cart.
Summary	Add a product to the basket.
Actors	Primary: visitor, customer, administrator. Secondary: none.
Pre-conditions	None.
Scenario	Add the product to the basket after viewing it.
Exceptions	None.
Post-condition	None.

Table 3.11 Add to cart.

Number	12
Name	Check out.
Summary	Confirm the bought operation.
Actors	Primary: customer. Secondary: none.
Pre-conditions	Add to cart (extend), cart not empty.
Scenario	The customer enter his personal information, payment information and take the product out from store.
Exceptions	Internet connection is lost by either the user or the system.
Post-condition	Products sells to the customer.

Table 3.12 Checkout.

Number	13
Name	Logon.
Summary	Admin login.
Actors	Primary: administrator. Secondary: none.
Pre-conditions	Administrator registered.
Scenario	The admin can login to the website and make his operations.
Exceptions	Internet connection is lost by either the user or the system.
Post-condition	Administrator logged on.

Table 3.13 Logon.

Number	14
Name	Store search.
Summary	Search in the database.
Actors	Primary: administrator. Secondary: none.
Pre-conditions	Logon.

Scenario	The administrator can use multi search methods for companies, categories and products to modify them.
Exceptions	Internet connection is lost by either the user or the system.
Post-condition	None.

Table 3.14 Store Search.

Number	15
Name	Edit offers.
Summary	Make offers.
Actors	Primary: administrator. Secondary: none.
Pre-conditions	Logon.
Scenario	The administrator can offer a product or update an existing offer.
Exceptions	Internet connection is lost by either the user or the system.
Post-condition	Product offered.

Table 3.15 Edit offers.

Number	16
Name	Update company.
Summary	Update company details.
Actors	Primary: administrator. Secondary: none.
Pre-conditions	Logon.
Scenario	The administrator can update a company information or delete the company.
Exceptions	Internet connection is lost by either the user or the system.
Post-condition	None.

Table 3.16 Update company.

Number	17
Name	Update category.
Summary	Update category details.
Actors	Primary: administrator. Secondary: none.

Pre-conditions	Logon.
Scenario	The administrator can update a category information or delete the category.
Exceptions	Internet connection is lost by either the user or the system.
Post-condition	None.

Table 3.17 Update category.

Number	18
Name	Update product.
Summary	Update product details.
Actors	Primary: administrator. Secondary: none.
Pre-conditions	Logon.
Scenario	The administrator can update a product information or delete the product.
Exceptions	Internet connection is lost by either the user or the system.
Post-condition	None.

Table 3.18 Update product.

Number	19
Name	Add company.
Summary	Add new company.
Actors	Primary: administrator. Secondary: none.
Pre-conditions	Logon.
Scenario	The administrator add new company to the companies list.
Exceptions	Internet connection is lost by either the user or the system.
Post-condition	None.

Table 3.19 Add company.

Number	20
Name	Add category.
Summary	Add new category.
Actors	Primary: administrator. Secondary: none.

Pre-conditions	Logon.
Scenario	The administrator add new category to the categories list.
Exceptions	Internet connection is lost by either the user or the system.
Post-condition	None.

Table 3.20 Add category.

Number	21
Name	Add product.
Summary	Add new product.
Actors	Primary: administrator. Secondary: none.
Pre-conditions	Logon.
Scenario	The administrator add new product to the products list.
Exceptions	Internet connection is lost by either the user or the system.
Post-condition	None.

Table 3.21 Add product.

3.1.2: Actor Specification:

3.1.2.1: Visitor.

Description: normal user not logged or not registered in the website, he can make some processes.

3.1.2.2: Customer.

Description: registered user can bought products and many other processes.

3.1.2.3: Administrator.

Description: website admin can make customer and visitor processes and he can controls the data (modify, delete, add).

3.1.2.4: Recommendation system.

Description: the used algorithm to give a recommended products for all customers, it works after the customer login, and generate product lists for every customer, the list depends on the similarity between products and rates.

3.2: SYSTEM DESIGN:

3.2.1: Database Tables:

We use C# classes to create database as we will see in [(4.6.1) Database access with Entity Framework Code-First]. Here is the used classes to create the database:

AccountModels:

This Model contains three tables (RegisterModel, LogOnModel and ChangePasswordModel) this Model process the user information, register in the website, login data and change the user password.

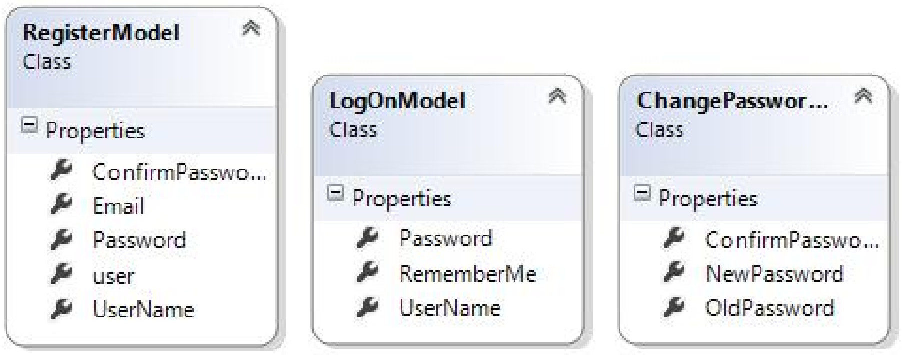


Figure 3.2 Account models.

Cart:

This table contains the basket data, and the products in this basket
PK: CartId.

FK: Product_id, RecordId.

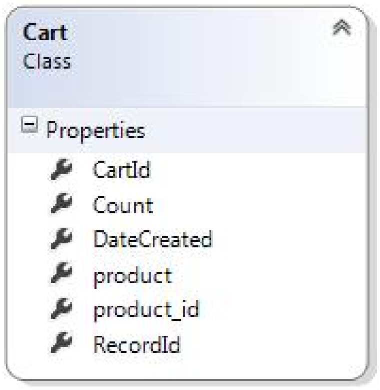


Figure 3.3 Cart.

Category:

This tables contains categories data.
PK: Category_Id.
FK: Products.

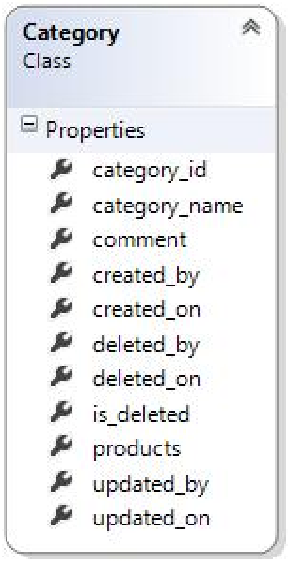
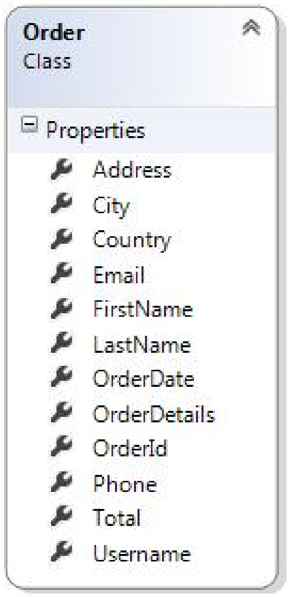


Figure 3.4 Category.

Order:

This table contains the full Customer data to complete the buy process.
PK: OrederId.

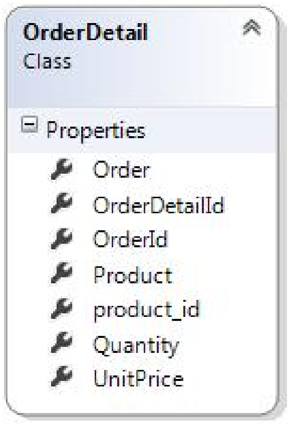


Address
City
Country
Email
FirstName
LastName
OrderDate
OrderDetails
OrderId
Phone
Total
Username

Figure 3.5 Order.

OrederDetail:

This table contains the website Data to complete the buy process.
PK: OrederDetailId.
FK: Product_id.



Order
OrderDetailId
OrderId
Product
product_id
Quantity
UnitPrice

Figure 3.6 Order details.

Product table:

This table contains the full details about products, the rates given by users, and the category details.
PK: Product_id
FK: Category_id, Product_type_id.

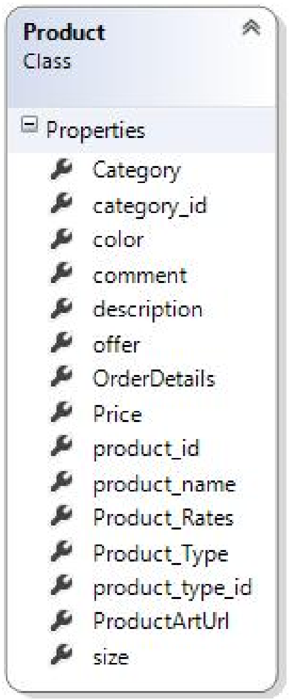


Figure 3.7 Product.

Product_Rate table:

This table contains the Rate data for Customer Rates
PK: product_rate_id.
FK: Product_id.

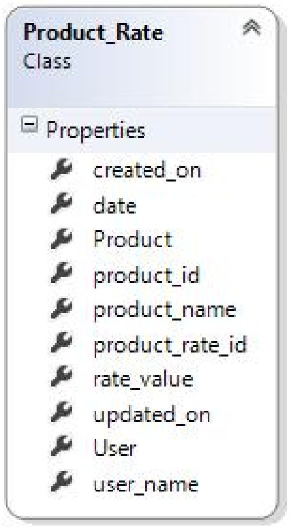


Figure 3.8 Product rate.

Product_Type table:

This table contains the companies for every product.
PK: product_type_id.

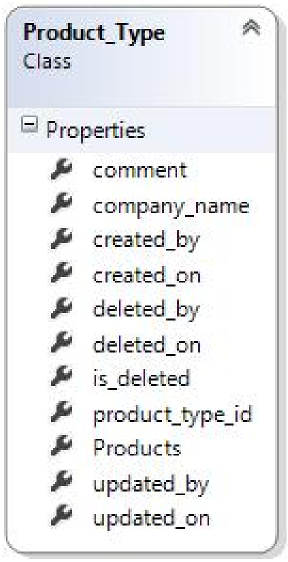


Figure 3.9 Product type.

User Table:

This table contains the users' Data and the orders for every customer.
PK: UserId.

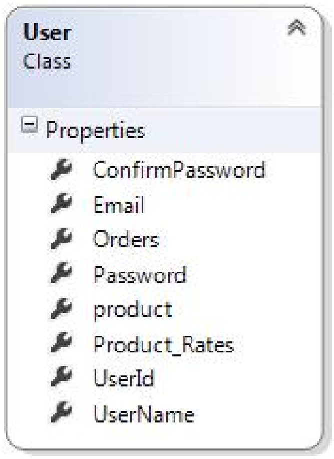


Figure 3.10 User.

ShoppingCart table:

This table contains the full process of the buy operation.

PK: ShoppingCartId.

Fields:

CartSessionKey: User Session number.
StoreDB: stored database.

Methods:

AddToCart: add products to cart.
CreateOrder: converts the shopping cart to an order during the checkout phase.
EmptyCart: removes all items from a user's shopping cart.
GetCart: static method which allows our controllers to obtain a cart object.
GetCartId: handle reading the CartId from the user's session.
GetCartItems: retrieves a list of CartItems for display or processing.
GetCount: retrieves the total number of products a user has in their shopping cart.
GetTotal: calculates the total cost of all items in the cart.
MigrateCart: When a user has logged in, migrate their shopping cart to be associated with their username
RemoveFromCart: delete product from user's cart.

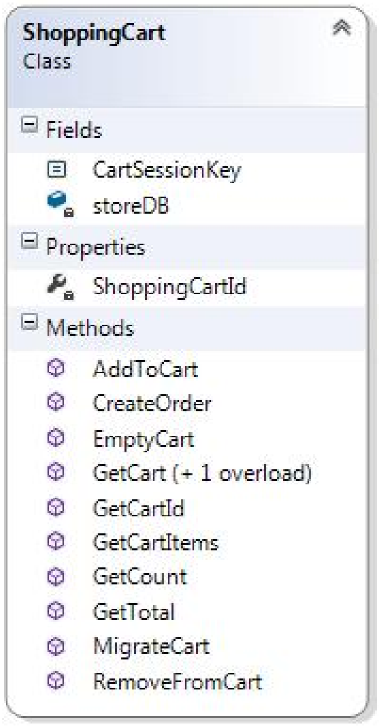


Figure 3.11 Shopping cart.

E_Commercentities:

This class contains the database properties and the relations between tables The Code First technique builds the Database, the created tables in the database is the properties of this class.

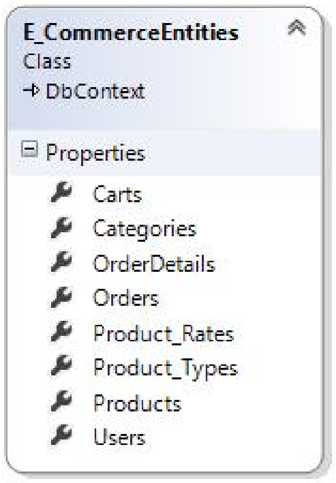


Figure 3.12 Ecommerce entities.

This class is the start of building database, the structure of this class is:

```
using System;
using System.Collections.Generic;
using System.Data.Entity;
using System.Linq;
using System.Web;

namespace E_Commerce.Models
{
    public class E_CommerceEntities:DbContext
    {
        public DbSet<Product> Products { get; set; }
        public DbSet<Category> Categories { get; set; }
        public DbSet<Product_Type> Product_Types { get; set; }
        public DbSet<Cart> Carts { get; set; }
        public DbSet<Order> Orders { get; set; }
        public DbSet<OrderDetail> OrderDetails { get; set; }
        public DbSet<User> Users { get; set; }
        public DbSet<Product_Rate> Product_Rates { get; set; }

    }
}
```

The foreign key in every table written as bellow:

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

       Order: the destination table.

        Order: name of method.
```

Adding our store catalog data:

We will take advantage of a feature in Entity Framework which adds “seed” data to a newly created database. This will pre-populate our store catalog with a list of companies, Categories, and Products.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Data.Entity;


namespace E_Commerce.Models
{
    public class SampleData :
DropCreateDatabaseIfModelChanges<E_CommerceEntities>
    {
        protected override void Seed(E_CommerceEntities context)
        {
            var category = new List<Category>
           {

                new Category{category_name="ACCESSORIES"},
           };



  var company = new List<Product_Type>
           {
                new Product_Type{company_name="-"},

           };

  new List<Product>
           {

                new Product {product_name="CARD READER/WRITER INTERNAL ALL
IN ONE",Product_Type=company.SingleOrDefault(c=>c.company_name=="-"),Price=
      520    ,color=      null   ,offer=      0      ,description="INTERNAL
CARD
READER/WRITER",Category=category.SingleOrDefault(c=>c.category_name=="ACCES
SORIES"),ProductArtUrl=   null   ,comment=    null   },
            }.ForEach(a => context.Products.Add(a));


        }

    }
}
```

This called seed Data the (EF) automatically create the IDs for every row in the database and fill the data into the original Database.
We call this class (SampleData) in the Global.asax file in Project Folder as below:
Add this row to the ApplicationStart () Method

```
System.Data.Entity.Database.SetInitializer(new
E_Commerce.Models.SampleData());
```

This row will initialize the database when we run the project depending on the SampleData.

3.2.2: Web Pages:

Here we will talk about the Web Pages in our website.
The Web Pages written in XHTML 1.0 Transitional, Jquery for the Slider, cascading sheets (CSS), and with the Razor engine (CSHTML) for the Content Pages.

3.1.2.1: Home Page:

This is the root page, this will viewed when we enter to the site.
It contains Search Method, links to all site, list of categories, About Us pages, offered products, the latest products, and the products that have the biggest rate.

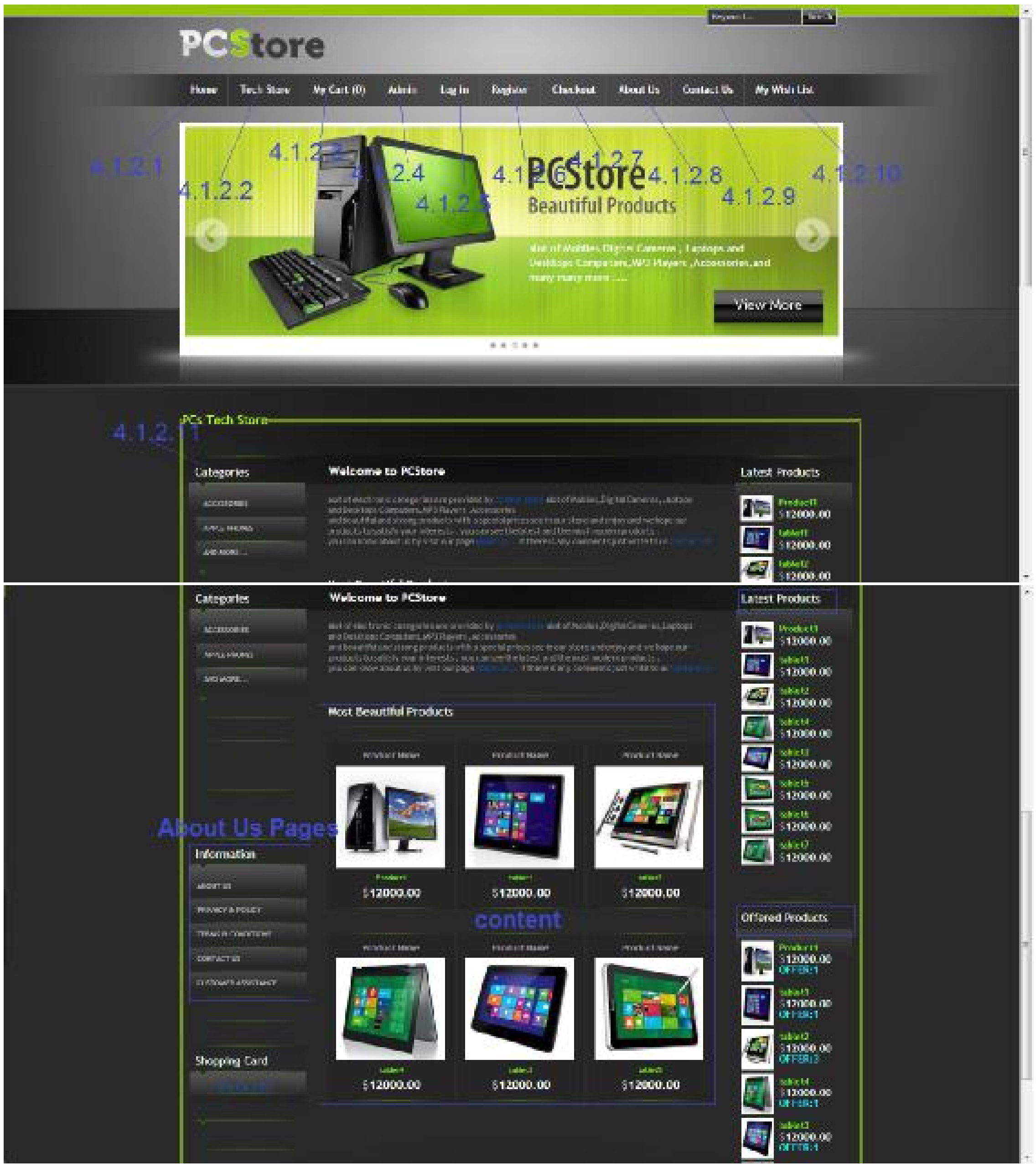
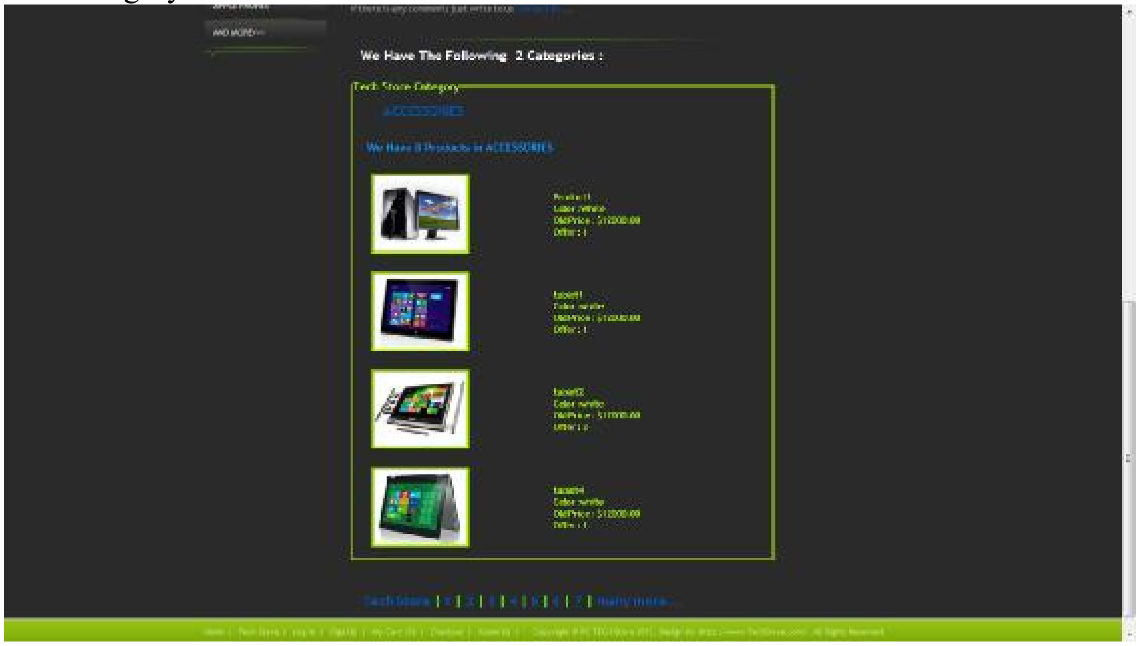


Figure 3.13 Home page.

3.1.2.2: Tech Store Page:

In this page we list the categories in our Website and the products that go under each category.


We Have The Following 2 Categories :

Tech Store Category

ACCESSORIES

We Have 8 Products in ACCESSORIES

Product1
Color: white
OldPrice : $12000.00
Offer : 1

tablet1
Color: white
OldPrice : $12000.00
Offer : 1

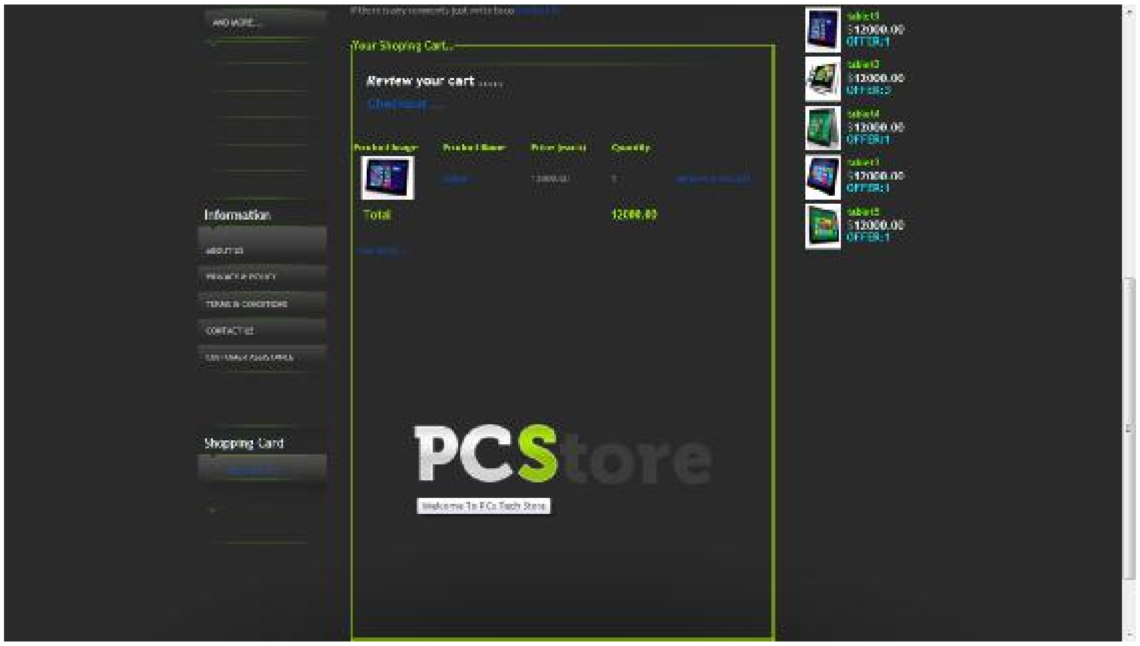
tablet2
Color: white
OldPrice : $12000.00
Offer : 2

tablet4
Color: white
OldPrice : $12000.00
Offer : 1

Figure 3.14 Tech store page.

3.1.2.3: My Cart Page:

This page contain the items that the customer added it to check it out then.


Your Shoping Cart..

Review your cart .....

Checkout ...

Product Image	Product Name	Price	Quantity
	[illegible]	12000.00	1
Total			12000.00

Information

ABOUT US

PRIVACY & POLICY

TERMS & CONDITIONS

CONTACT US

Shopping Card

tablet1
$12000.00
OFFER:1

tablet2
$12000.00
OFFER:3

tablet4
$12000.00
OFFER:1

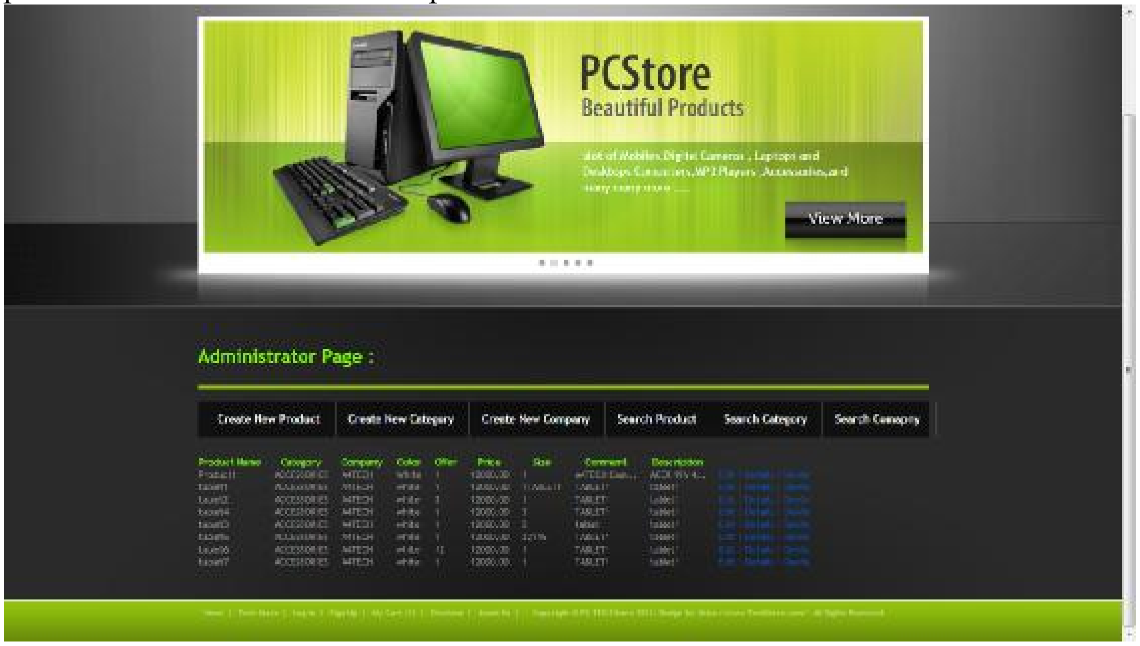
tablet1
$12000.00
OFFER:1

tablet5
$12000.00
OFFER:1

Figure 3.15 My cart page.

3.1.2.4: Admin Page:

This is the admin page, the admin can edit and add companies, categories, products and can make offers in products.



Product Name	Category	Company	Color	Offer	Price	Size	Comment	Description	
Product1	ACCESSORIES	M-TECH	white	1	12000.00	1	M-TECH Com...	ACER 15V 4...	[illegible]
tablet1	ACCESSORIES	M-TECH	white	1	12000.00	[illegible]	TABLET	tablet	[illegible]
tablet2	ACCESSORIES	M-TECH	white	1	12000.00	1	TABLET	tablet	[illegible]
tablet4	ACCESSORIES	M-TECH	white	1	12000.00	1	TABLET	tablet	[illegible]
tablet5	ACCESSORIES	M-TECH	white	1	12000.00	1	tablet	tablet	[illegible]
tablet6	ACCESSORIES	M-TECH	white	1	12000.00	[illegible]	TABLET	tablet	[illegible]
tablet6	ACCESSORIES	M-TECH	white	11	12000.00	1	TABLET	tablet	[illegible]
tablet7	ACCESSORIES	M-TECH	white	1	12000.00	1	TABLET	tablet	[illegible]

Figure 3.16 Admin page.

3.1.2.5: Login Page:

This is the login page where the customer logs to the website using his user name and the password.

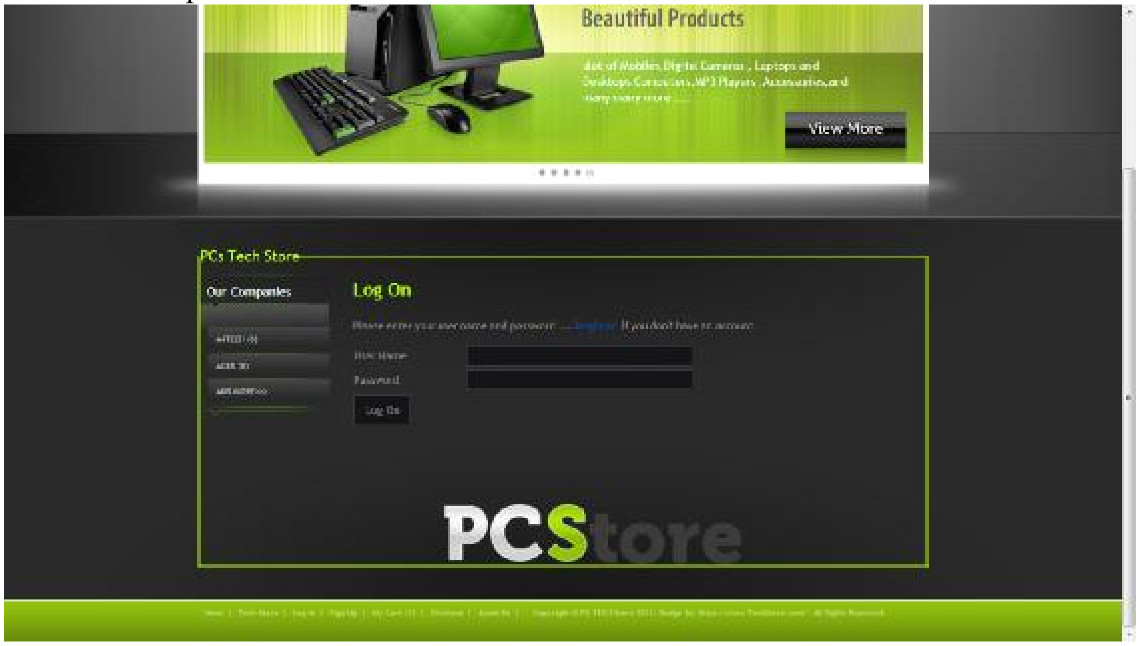
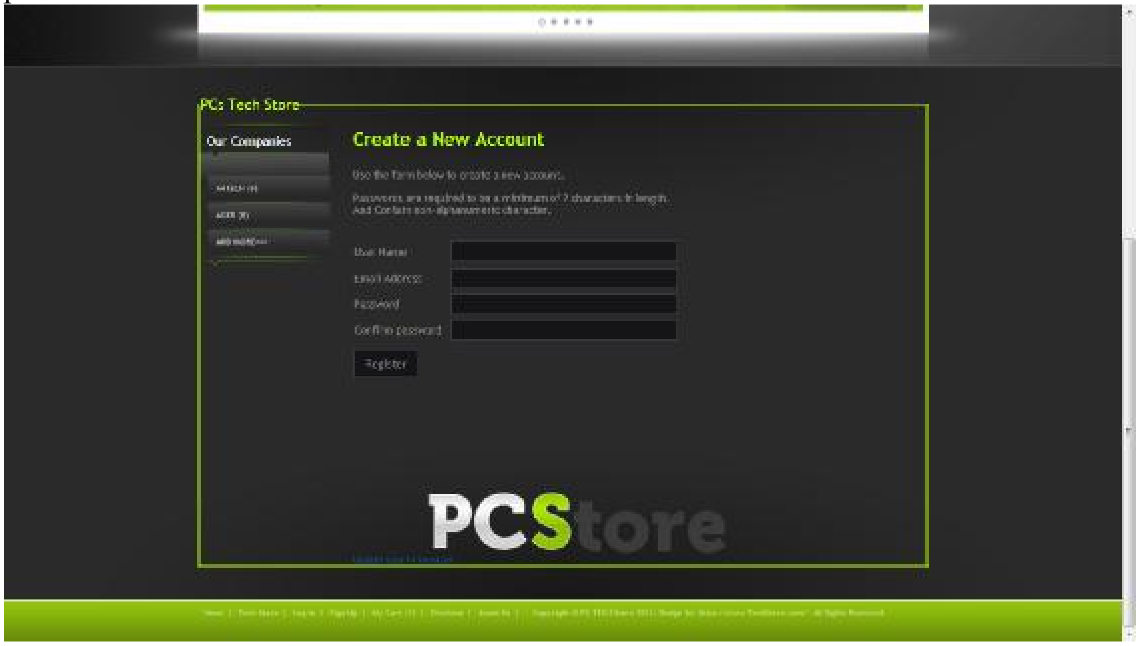


Figure 3.17 Login page.

3.1.2.6: Register Page:

This page where the new user can register in our website by entering his personal information.



PCs Tech Store

Our Companies

Create a New Account

Use the form below to create a new account.

Passwords are required to be a minimum of 7 characters in length.
And Contain non-alphanumeric character.

User Name:
Email Address:
Password:
Confirm password:

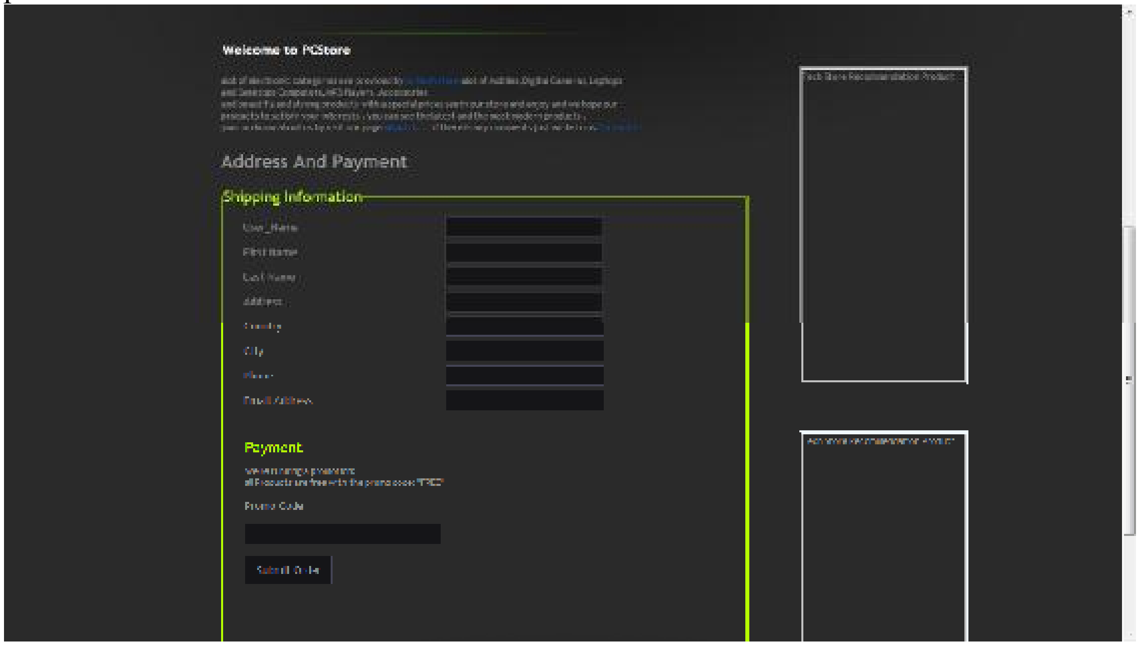
Register

PCStore

Figure 3.18 Register page.

3.1.2.7: Checkout Page:

In this page the user can confirm his bought operation by adding his full personal information.



Welcome to PCStore

[illegible]

Address And Payment

Shipping Information

User_Name
First Name
Last Name
Address
Country
City
Phone
Email Address

Payment

[illegible]
all Products are free with the promo code "FREE"

Promo Code

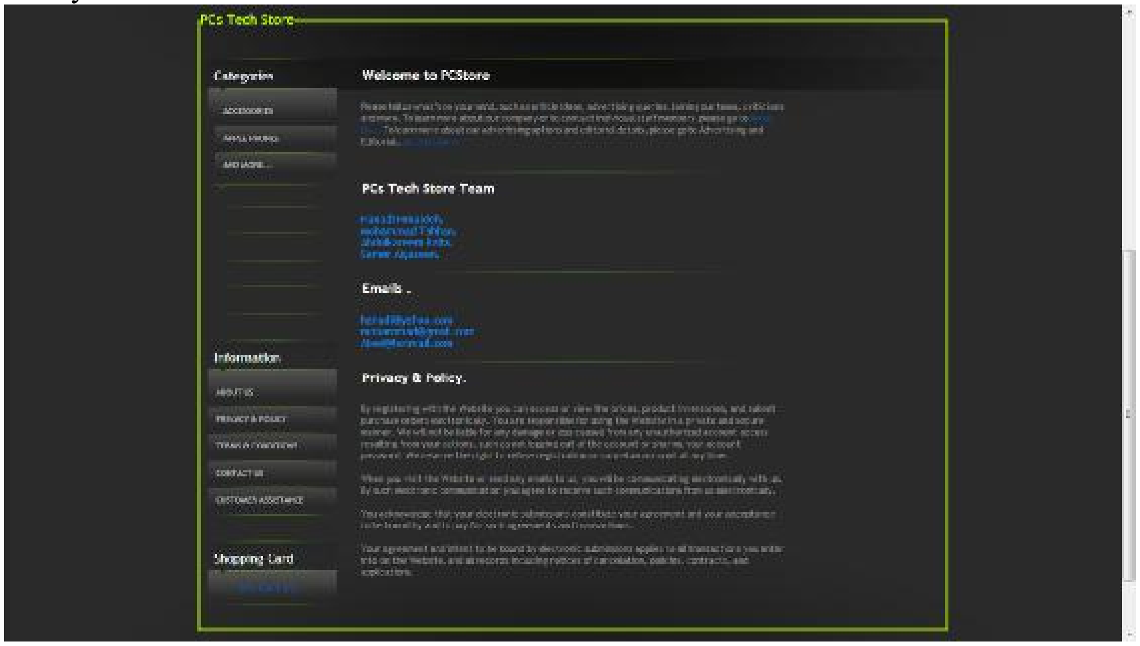
Submit Order

Tech Store Recommendation Product

Figure 3.19 Checkout page.

3.1.2.8: About us page:

In this page we list the team members, our numbers, E-mails, and Privacy & Policy Conditions.



PCs Tech Store

Categories

ACCESSORIES

[illegible]

AND MORE...

Welcome to PCStore

[illegible]

PCs Tech Store Team

[illegible]

Emails .

[illegible]

Information

ABOUT US

PRIVACY & POLICY

TERMS & CONDITIONS

CONTACT US

CUSTOMER ASSISTANCE

Privacy & Policy.

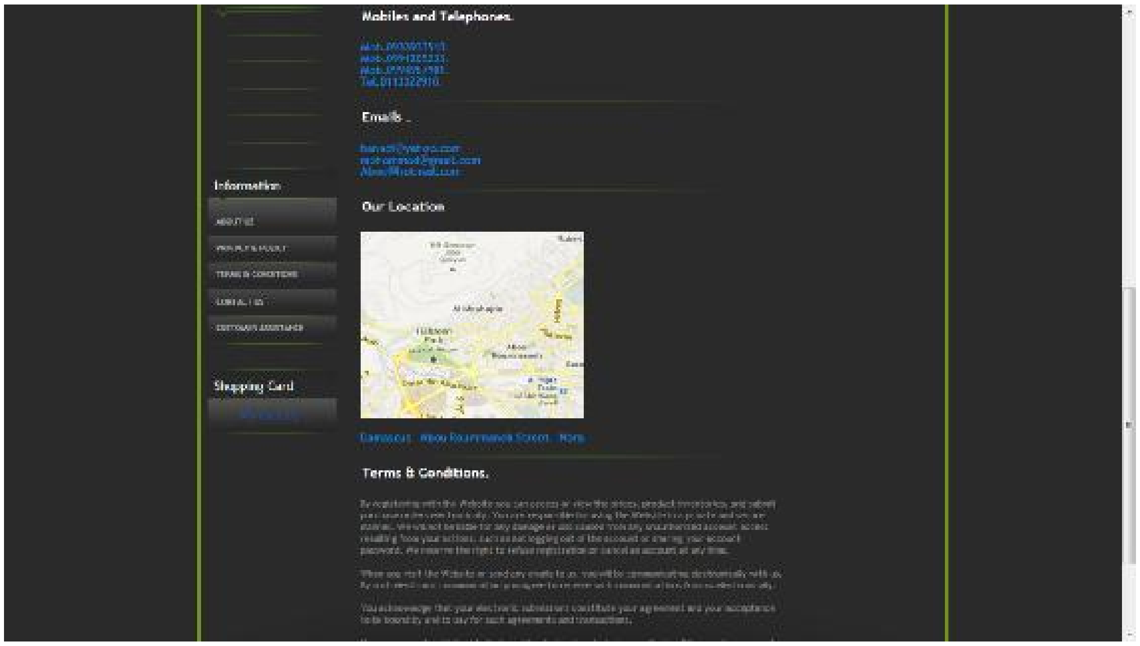
[illegible]

Shopping Card

Figure 3.20 About us page.

3.1.2.9: Contact us Page:

In this page you can to contact us, email us, and take our location, and terms & conditions.



Mobiles and Telephones.

[illegible]

Emails .

[illegible]

Information

ABOUT US

PRIVACY & POLICY

TERMS & CONDITIONS

CONTACT US

CUSTOMER ASSISTANCE

Our Location

Shopping Card

Terms & Conditions.

[illegible]

Figure 3.21 Contact us page.

3.1.2.10: My Wish List Page:

This page listed a list of products recommended for each user depending on our recommendation system, you can to see this page when you login, and the website redirect the customer after login to this Page.

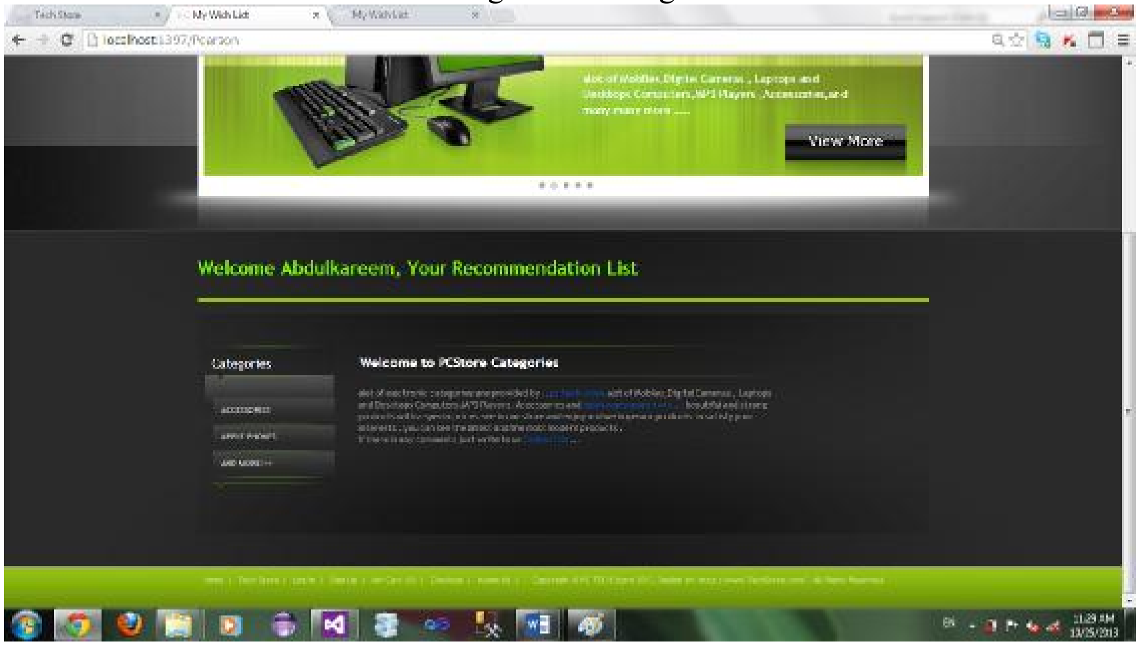

My Wish List
View More
Welcome Abdulkareem, Your Recommendation List
Categories
ACCESSORIES
APPLE PHONES
AND MORE>>
Welcome to PCStore Categories

Figure 3.22 My wish list page.

3.1.2.11: Categories Page:

In this page we list the categories in our Website and the products that go under each category.

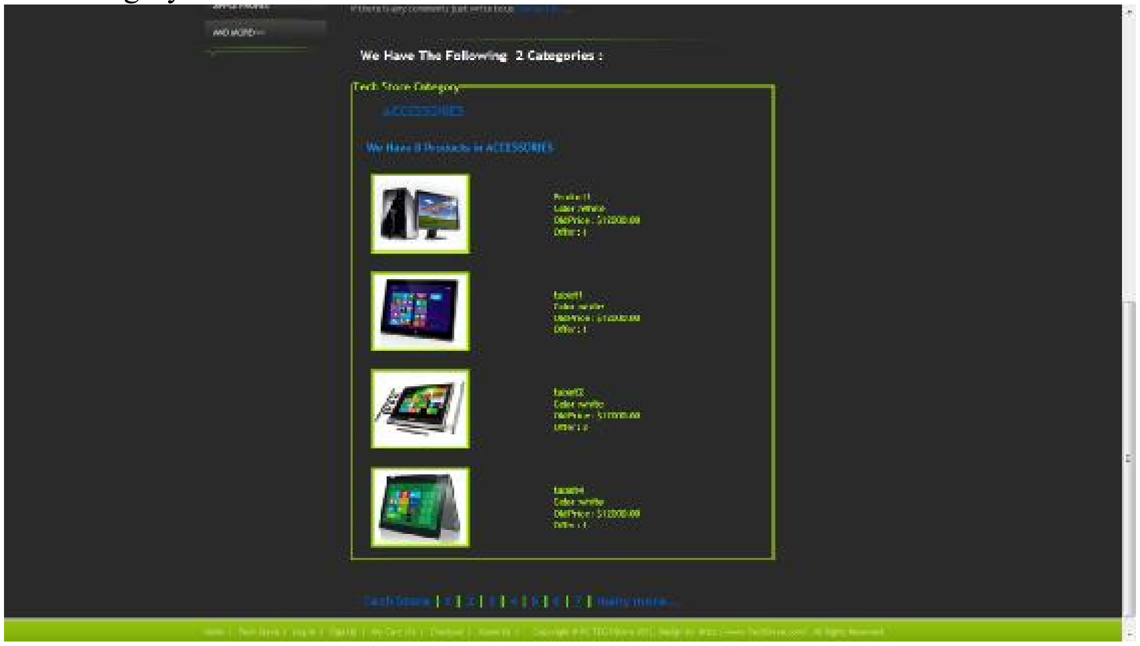
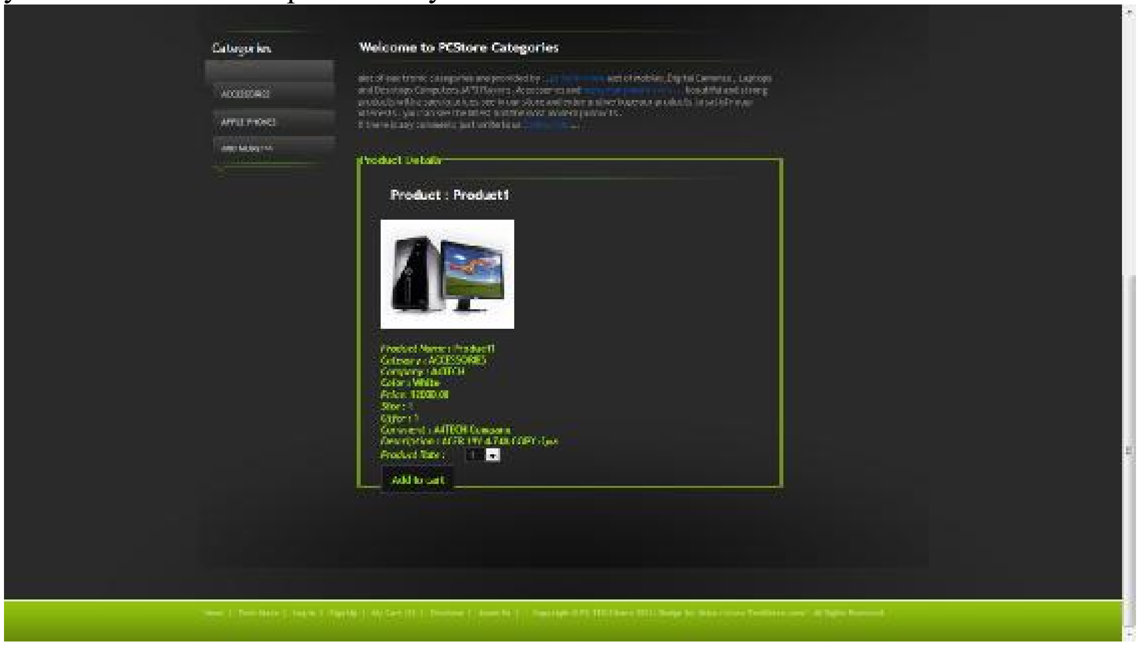


Figure 3.23 Categories page.

3.1.2.12: Details Page:

In this page you can to see the full details about each product and you can give your rate and add the product to your cart.



[illegible]

Product : Product1

Product Name : Product1
Category : ACCESSORIES
Company : AATECH
Color : White
Price: 12000.00
Size : 1
Other : 1
Comment : AATECH Company
Description : [illegible]
Product Rate : 1

Figure 3.24 Details page.

Chapter4: ASP.NET MVC 4

4.1: WHAT IS ASP.NET MVC?

ASP.NET MVC provides an alternative to Web Forms for building web applications on the .NET platform. It was first unveiled by Microsoft in November 2007 and has since had four major releases. The third major version, ASP.NET MVC 3, was released at the end of January 2011 and is the first version of ASP.NET MVC to take a dependency on .NET 4. ASP.NET MVC 4 works with .NET 4 as well as .NET 4.5, which has not released at publishing time MVC stands for Model-View-Controller, a design pattern that's very popular in the web development space.
As an alternative to Web Forms, ASP.NET MVC takes a different approach when it comes to structuring web applications. This means you won't be dealing with ASPX pages and controls, post backs or view state, or complicated event lifecycles. Instead, you'll be defining controllers, actions, and views. The underlying ASP.NET platform is the same, however, so things like HTTP handlers and HTTP modules still apply, and you can mix MVC and Web Forms pages in the same application. Both ASP.NET Web Forms and ASP.NET MVC sit alongside each other on top of the core ASP.NET platform.

4.2: THE MVC PATTERN:

The Model-View-Controller pattern originated in the Smalltalk development community in the 1970s, although it was popularized for use on the web with the advent of Ruby on Rails in 2003.
There are three pieces to the MVC pattern:
The model—the domain that your software is built around. If you were building a blog, your models might be post and comment. In some contexts, the term model might refer to a view-specific model a representation of the domain for the specific purpose of being displayed in the user interface.

The view—the visual representation of a model, given some context. It's usually the resulting markup that the framework renders to the browser, such as the HTML representing the blog post.

The controller—the coordinator that provides the link between the view and the model. The controller is responsible for processing input, acting upon the

model, and deciding on what action should be performed, such as rendering a view or redirecting to another page.
Continuing the blog example, the controller might look up the most recent comments for a post (the model) and pass them to the view for rendering. As shown in figure (2.1).

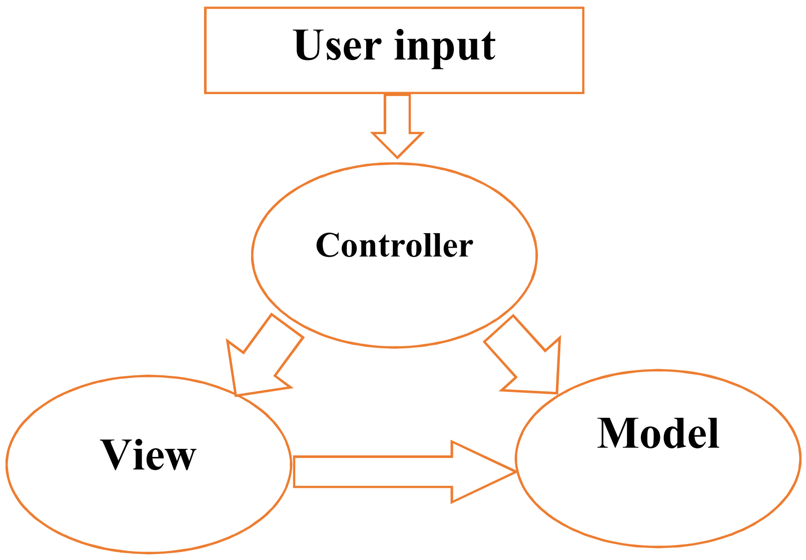


Figure 4.1 MVC Pattern.

4.3: BENEFITS OF ASP.NET MVC:

ASP.NET MVC addresses many of the shortcomings of ASP.NET Web Forms, which can often make it a better choice for developing new applications on the .NET platform.

4.3.1: CLOSER TO THE PROTOCOL:

While ASP.NET Web Forms attempts to completely hide the stateless nature of HTTP, ASP.NET MVC doesn’t. By embracing the MVC pattern and mapping a single HTTP request to a single method call, ASP.NET MVC provides a development experience that is far more familiar to anyone with a web development background. The model is also drastically simplified—gone are the complex page lifecycle events of Web Forms, and the abstractions over HTTP are minimal.

4.3.2: SEPARATION OF CONCERNS

While ASP.NET Web Forms tightly couples the user interface to its code-behind, ASP.NET MVC encourages a design where the user interface (the view) is kept separate from the code that drives it (the controller). When implemented well, this means that applications can be easier for developers to navigate, and it also makes the application easier to maintain—making a change to a controller doesn't necessarily mean you also have to modify the user interface.

4.3.3: TESTABILITY

By separating application logic from the user interface, ASP.NET MVC makes it easier to test individual components in isolation. Controller classes can be tested without testing the actual user interface. Unlike Web Forms, MVC controllers do not have a direct dependency on the infamously untestable HttpContext class and instead rely on an abstraction, which makes it far easier to write automated unit tests.
Now that you've seen some of the benefits of ASP.NET MVC, we'll briefly explore what's new in the third release of the framework.

4.4: WHAT'S NEW IN ASP.NET MVC 3/4?

MVC 3 and 4 come with many improvements and several new features in addition to the new dependency on .NET 4. These new features include

■ The Razor view engine.
■ Package management with NuGet.
■ Improved extensibility.
■ Global action filters.
■ Dynamic language features.
■ Partial page output caching.
■ Ajax improvements.
■ Enhancements to the validation infrastructure.
■ Mobile templates.
■ Web API.

4.5: THE RAZOR VIEW ENGINE

One of the core components of the new ASP.NET Web Pages technology is the Razor view engine. This engine provides a concise way to mix code and markup within the same file.
ASP.NET MVC applications can also make use of the Razor view engine as an alternative to the Web Forms view engine that was available in both ASP.NET MVC 1 and 2.
As an example, the following code snippet shows a simple page that constructs a list of product names using the older Web Forms view engine:

```
<%@ Page Language="C#"
Inherits="System.Web.Mvc.ViewPage<Product[]>" %>
<ul>
<% foreach(var product in Model) { %>
<li><%: product.Name %></li>
<% } %>
</ul>
```

This is quite verbose. The Page declaration at the top and the code nuggets (`<%` and `%>`) that are used to switch between code and markup add a lot of additional characters to the page markup. By contrast, Razor provides a much cleaner way to achieve the same result:

```
@model Product[]
<ul>
@foreach(var product in Model) {
<li>@product.Name</li>
}
</ul>
```

As you can see, Razor does not require code nuggets to transition between code and markup, which helps to keep view logic much more focused on the page's markup.

4.6: DATA ACCESS:

For data access to our project we have used the SQL Server Compact Edition (SQL CE), it is a free, embedded, file database that doesn't requires any installation or configuration.

4.6.1: Database access with Entity Framework Code-First:

We'll use the Entity Framework (EF) support that is included in ASP.NET MVC 3 projects to query and update the database. EF is a flexible object relational mapping (ORM) data API that enables developers to query and update data stored in a database in an object-oriented way.

Entity Framework version 4 supports a development paradigm called code-first. Code-first allows you to create model object by writing simple classes (also known as POCO from "plain-old" CLR objects), and can even create the database on the fly from your classes.

And here is a brief example on how Products table created:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.ComponentModel.DataAnnotations;
using System.Web.Mvc;
using System.ComponentModel;

namespace E_Commerce.Models
{
      public class Product
    {
        public int product_id { get; set; }
        public int product_type_id { get; set; }
        public int category_id { get; set; }
        public string product_name { get; set; }
        public string comment { get; set; }
        public string description { get; set; }
        public Nullable<double> offer { get; set; }
        public decimal Price { get; set; }
        public string ProductArtUrl { get; set; }
        public string color { get; set; }
        public string size { get; set; }
        public virtual Category Category { get; set; }
        public virtual Product_Type Product_Type { get; set; }
        public virtual List<OrderDetail> OrderDetails { get; set; }
        public virtual List<Product_Rate> Product_Rates { get; set; }
    }
}
```

4.6.2: Using Data Annotations for Model Validation:

We have a major issue with our Create and Edit forms: they're not doing any validation. We can do things like leave required fields blank or type letters in the Price field, and the first error we'll see is from the database.
We can easily add validation to our application by adding Data Annotations to our model classes. Data Annotations allow us to describe the rules we want applied to our model properties, and ASP.NET MVC will take care of enforcing them and displaying appropriate messages to our users.

And here is the main Data Annotation attributes:

■ **Required** – Indicates that the property is a required field.

■ **DisplayName** – Defines the text we want used on form fields and validation messages.

■ **StringLength** – Defines a maximum length for a string field.

■ **Range** – Gives a maximum and minimum value for a numeric field.

■ **Bind** - Lists fields to exclude or include when binding parameter or form values to model properties.

■ **ScaffoldColumn** - Allows hiding fields from editor forms.

After making the validation using Data Annotation on the model, the Product class (table) will be as shown:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.ComponentModel.DataAnnotations;
using System.Web.Mvc;
using System.ComponentModel;

namespace E_Commerce.Models
{
   [Bind(Exclude = "product_id")]
    public class Product
    {
        [Key]
        [ScaffoldColumn(false)]
        public int product_id { get; set; }
        [DisplayName("Product_Type")]
        public int product_type_id { get; set; }
        [DisplayName("Category")]
        public int category_id { get; set; }
        [Required(ErrorMessage = "An Product Name is required")]
        [StringLength(200)]
        public string product_name { get; set; }
        public string comment { get; set; }
        public string description { get; set; }
        public Nullable<double> offer { get; set; }
        [Required(ErrorMessage = "Price is required")]
        [Range(0.01, 100.00,
        ErrorMessage = "Price must be between 0.01 and 100.00")]
        public decimal Price { get; set; }
        [DisplayName("Product Art URL")]
        [StringLength(1024)]
        public string ProductArtUrl { get; set; }
        public string color { get; set; }
        public string size { get; set; }
        public virtual User user { get; set; }
        public virtual Category Category { get; set; }
        public virtual Product_Type Product_Type { get; set; }
        public virtual List<OrderDetail> OrderDetails { get; set; }
        public virtual List<Product_Rate> Product_Rates { get; set; }
    }
}
```

4.7: SUMMARY:

This chapter talks about ASP.NET MVC history, what is ASP.NET MVC, How to work with MVC, how to create our Database and validate it, and other new techniques used in our project.

Chapter 5: SYSTEM VALIDATION

5.1: BACKGROUND:

The algorithm we're going to be using to find the trends in our data is called the Pearson Correlation Score. The reason for this choice is simple - it effectively can handle and balance un-normalized data.

5.2: WHY RECOMMENDATION SYSTEM USING COLLABORATIVE FILTERING:

• Based on real activity
the biggest benefit of recommendation systems is that they record, and then base their recommendations on actual user behavior. Their recommendations are not based on guesswork, but on an objective reality. This is the holy grail of design: watching people in their natural environment and making design decisions based directly on the results. Recommendation systems are not perfect, but because they predict the future based on the past, they are remarkably good.

• Great for discovery
sometimes it can feel like we're the victims of horribly inefficient advertising. This is apparent when we go to the movies and none of the previews are interesting, or when none of the music on the radio is aligned to our tastes. Recommendation systems help alleviate this problem because they allow us to discover things that are similar to what we already like. And they can make some pretty surprising recommendations that we probably wouldn't have found out about otherwise.

• Always up to date
Recommendation systems are dynamically updated, and therefore are always up-to-date. Some new, interesting news? It will be up to date within minutes. An interesting new product on Amazon? It quickly gets recommended as long as people rate it highly. The ability for a recommendation system to bubble up activity in real time is a huge advantage because the system is always on.

• Reduced organizational maintenance
Building recommendation systems is quite different from how we've built information-rich web sites in the past. For many designers the primary task of building an information-rich web site is creating navigation systems built on

top of an underlying taxonomy. The taxonomy is built out of the designer's knowledge of users and the domain, generated from observations made during field research, insights from persona creation, or knowledge gained from other design techniques. Most of the organizational maintenance of a site is keeping the navigation system and taxonomy in line with the users' changing needs.

5.3: SYSTEM TEST:

Here, we have a collection of products, and each product has a collection of reviews. Each review has a value from 1 to 5 describing how well they liked the product. This value can be based on what you are trying to profile. For example, you could use a value of 1 to represent a user that bought a particular product (with a 0 showing users who did not purchase a product). As long as we have a way to convert the data to a numerical value, the data can be analyzed.

Here we use a Dictionary Data Structure to represent product and users who buy the product and rate it.

```
Dictionary<string, List<Product_Rate>> productRecommendations = new
Dictionary<string, List<Product_Rate>> ();
```

The parameters of this dictionary is:
First parameter is the name of product.(string).
Second parameter is a list of users who bought the product and his rates.

```
foreach (string product in products)
            {
                productRecommendations.Add (product, db.Product_Rates.Where(x
=> x.Product_Name == product).ToList ());
            }
```

User_Name	Rate_Value	Product_Name
Mohammad	4.5	CARD READER/WRITER INTERNAL ALL IN ONE
Abdulkareem	2.5	CARD READER/WRITER INTERNAL ALL IN ONE
Hanadi	5	CARD READER/WRITER INTERNAL ALL IN ONE
Hasan	2	CARD READER/WRITER INTERNAL ALL IN ONE
Mohammad	5	CABLE POWER
Abdulkareem	3.5	CABLE POWER
Hanadi	1	CABLE POWER
Hasan	3.5	CABLE POWER
Razan	1	CABLE POWER
Mohammad	1	CABLE PS/2 KEY
Hanadi	5	CABLE PS/2 KEY
Abdulkareem	3.5	CABLE PS/2 KEY

Hasan	4	CABLE PS/2 KEY
Razan	4	CABLE PS/2 KEY
Abdulkareem	3.5	CABLE USB Printer 3M
Hanadi	4	CABLE USB Printer 3M
Hasan	5	CABLE USB Printer 3M
Razan	2	CABLE USB Printer 3M
Mohammad	4.5	MOUSE PAD CEC
Abdulkareem	5	MOUSE PAD CEC
Hasan	3	MOUSE PAD CEC

Table 5.1 Test set.

Here we compare between every product that user bought and rate with every product that other users bought and rate.
The result of every comparison give us a number between [-1, 1], and hence the number is near to (1) the result is better.
For example when we run the algorithm for Mohammad:

Muhammad has bought [CARD READER/WRITER INTERNAL ALL IN ONE], this product will be compared with all other products in the table, and the values we get from the algorithm is:

1- [CABLE POWER] = -0.307254933899513
2- [CABLE PS/2 KEY] = -0.133074827804663
3- [CABLE USB Printer 3M] = -0.339422116651.67
4- [MOUSE PAD CEC] = 0.45392064951602

And Mohammad bought [CABLE POWER], and the result given from algorithm depending on this product is:

1- [CARD READER/WRITER INTERNAL ALL IN ONE] = -0.307254933899513
2- [CABLE PS/2 KEY] = - 0.8316378415802933
3- [CABLE USB Printer 3M] = 0.57735026918962573
4- [MOUSE PAD CEC] = 0.27735009811261468

And Mohammad bought [CABLE PS/2 KEY], and the result given from algorithm depending on this product is:

1- [CARD READER/WRITER INTERNAL ALL IN ONE] = -0.133074827804663
2- [CABLE POWER] = - 0.8316378415802933
3- [CABLE USB Printer 3M] = 0.18543452998910614
4- [MOUSE PAD CEC] = -0.4234151591787097

And Mohammad bought [MOUSE PAD CEC], and the result given from algorithm depending on this product is:

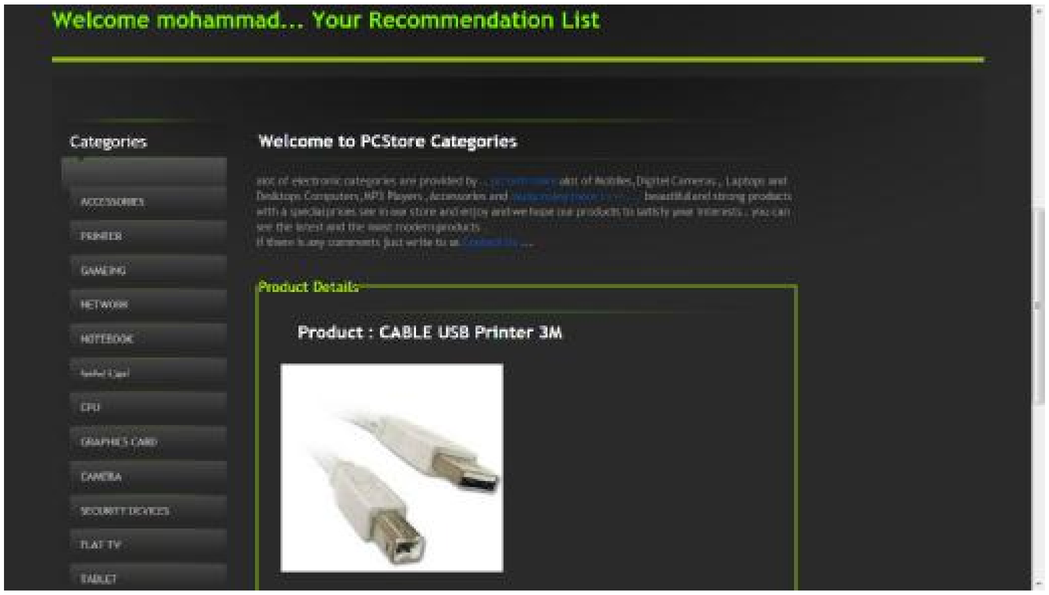
1- [CARD READER/WRITER INTERNAL ALL IN ONE] = 0.45392064951602
2- [CABLE POWER] = 0.27735009811261468
3- [CABLE USB Printer 3M] = -1.0
4- [CABLE PS/2 KEY] = -0.4234151591787097

We have to loop through each product, then order results by descending then select distinct value (bigger value) from each group of Dictionaries.

If result = 1 => this product bought before. Otherwise the product will be suggested to a customer according to the bigger value.

1- “CABLE POWER” = 1
2- “MOUSE PAD CEC” = 1
3- “CARD READER/WRITER INTERNAL ALL IN ONE” = 1
4- “CABLE PS/2 KEY” = 1
5- “CABLE USB Printer 3M” = 0.55470019622522915

And hence Mohammad bought all previous product except “CABLE USB Printer 3M” this product will be suggested to Mohammad because it is the highest and not bought yet.



Welcome to PCStore Categories

alot of electronic categories are provided by [illegible] alot of Mobiles, Digital Cameras , Laptops and Desktops Computers, MP3 Players , Accessories and [illegible] beautiful and strong products with a special prices see in our store and enjoy and we hope our products to satisfy your interests , you can see the latest and the most modern products
If there is any comments just write to us [illegible] ...

Product : CABLE USB Printer 3M

Figure 5.1 Recommendation1.

If someone enter to the website and buy and rate a new product and gives high rate it should to be in recommendation list to Mohammad:
“GAMING GAMEPAD GENIUS MAXFIRE G-08XU USB” this product is rated by Hanadi, Abdulkareem, and Hasan. This users rated this product as Abdulkareem=3, Hanadi=3, and Hasan=4.
And the new recommended list is:

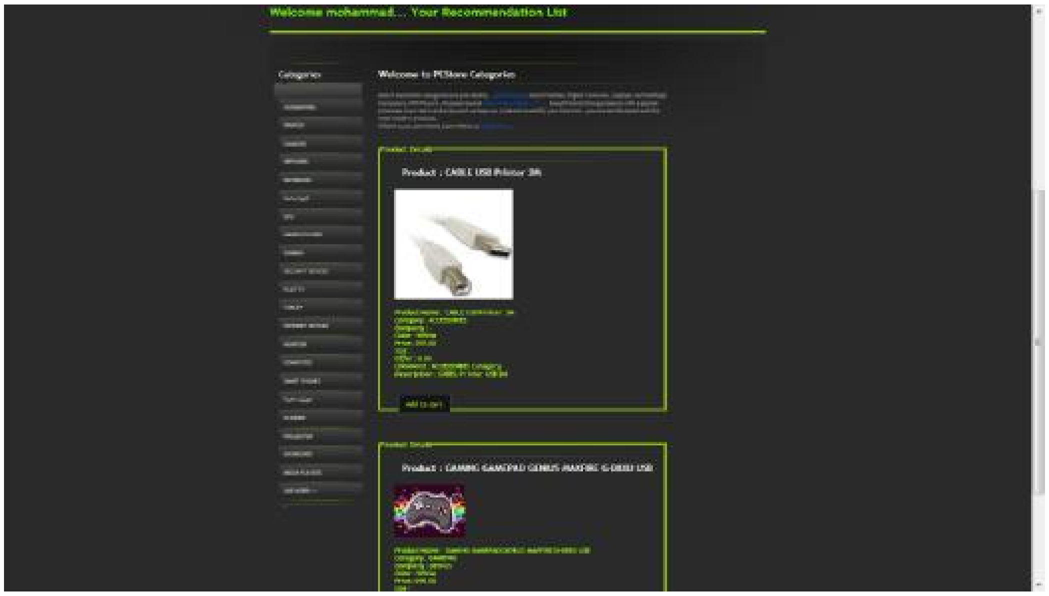

Welcome mohammad... Your Recommendation List
Categories
Welcome to PCStore Categories
Product : CABLE USB Printer 3M
Product : GAMING GAMEPAD GENIUS MAXFIRE G-08XU USB

Figure 5.2 Recommendation2.

The probability of the new item in recommended list is = 0.500000000001
It is smaller than the old product so that the old product comes first.

This how our system still up to date. And every time the user logs in have a new recommendation.

References:

1- Intro to ASP.NET MVC4 with Visual Studio (Beta) .Author Rick Anderson and Scott Hanselman 2011.

2- ASP.NET MVC4 in Action .Author Jeffrey Palermo and Eric Hexter 2012.

3- Amazon.com Recommendations Item-to-Item Collaborative filtering Greg Linden, Brent Smith, and Jeremy York • *Amazon.com.*

4- Item Based Collaborative Filtering Recommendation Algorithms , Group Lens Research Group/Army HPC Research Center Department of Computer Science and Engineering University of Minnesota.

5- A Review on Personalized Information Recommendation System Using Collaborative Filtering, Department of Computer Science and Engineering Manipal Institute of Technology.

6- www.Asp.net/mvc : the official website for asp.net by Microsoft corporation.

7- The StackOverflow Site http://stackoverflow.com/questions/asp-mvc

8- Codeproject.com or some problems.

9- The e-Commerce Safety Guide 2003. Author. Robert Chesnut.

10- Conceptualizing the Role of Collaborative E-Commerce .Author Charles Steinfield.

11- The W3Schoole Site : www.w3schools.com/aspnet/mvc_intro.asp .