

Web Based Technologies – Midterm

In this exam, you will design a shopping website that displays items information and allows user to add products into shopping cart.

You have to design **four** web pages (**Default.aspx**, **Login.aspx**, **ProductInfo.aspx**, and **Cart.aspx**) and a class page (**Product.cs**) that holds product information.

Preliminary Work

1. Create a web project for the exam.
2. Download the sample product-cover images from the web site and put these images into a folder named **images** in your web folder.

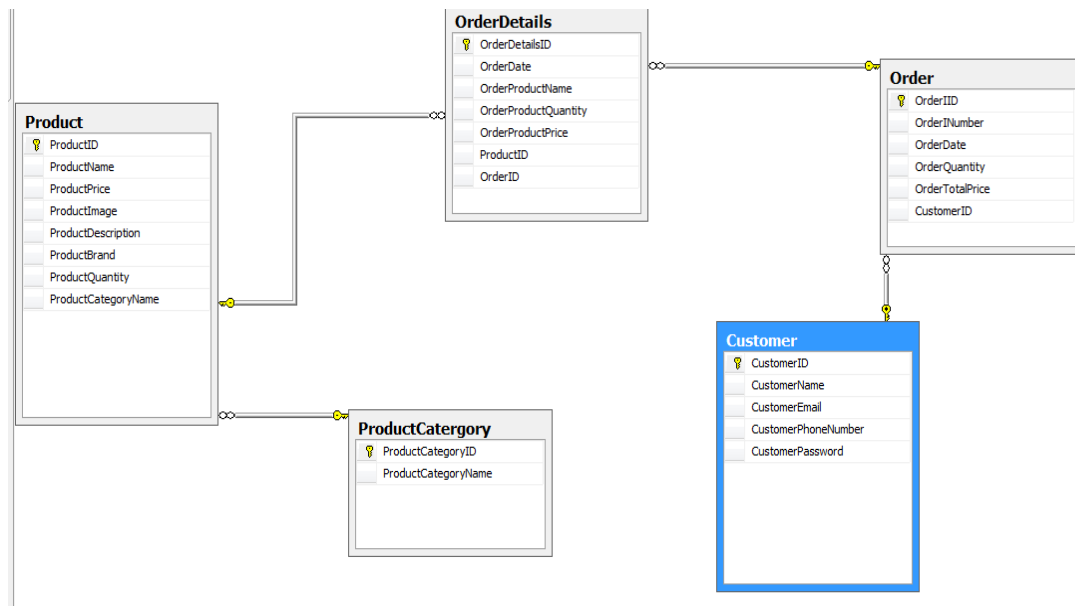
Product.cs

1. Add a class file named **Product.cs** into your project (Add New Item, Class). This file will contain information about the products. If you encounter a message which states that this file should be in **App_Code** folder, then select **Yes** and let Visual Studio create the **App_Code** folder for you and put **Product.cs** file in this folder. After then, you will be able to use the **Product** class in your application like any other classes of .NET library.
2. Add member variables to the class **Product** which are **ProductID** (integer), **Category** (string), **Writer** (string), **Director** (string), **Price** (Double), and **ProductImage** (string).
3. Define a constructor for the **Product** class which takes all of its member variables as parameter (**Hint**: In Visual Studio, type **ctor** and press **TAB** key. This constructs an empty constructor for the class. After then, change the constructor parameters and implementation).
4. Finally, your Product class should be like this:

```
public class Product
{
    public int ProductID;
    public string ProductName;
    public double ProductPrice;
    public string ProductImage;
    public string ProductDescription;
    public string ProductBrand;
    public int ProductQuantity;
    public string ProductCategoryName;

    public Product(int ProductID, string ProductName, double ProductPrice,
        string ProductImage, string ProductDescription, string ProductBrand,
        int ProductQuantity, string ProductCategoryName
    )
    {
        this.ProductID = ProductID;
        this.ProductName = ProductName;
        this.ProductPrice = ProductPrice;
        this.ProductImage = ProductImage;
        this.ProductDescription = ProductDescription;
        this.ProductBrand = ProductBrand;
        this.ProductQuantity = ProductQuantity;
        this.ProductCategoryName = ProductCategoryName;
    }
}
```

Design of Database



Design of Default.aspx

When loaded, **Default.aspx** page should check whether the user is logged in or not. This check should be accomplished by a **cookie** check. The cookie should contain only the **first name** and **last name** of the user. If no such cookie is defined, then **Default.aspx** should display a **message** and a **link** to the **Login.aspx** as in Fig. 1.

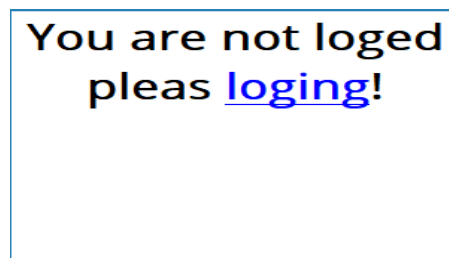


Fig. 1 – Default.aspx when user is not logged in.

The login form contains the following elements:

- Header**: Welcome!
- First Name**: Input field with the value "Şahin".
- Last Name**: Input field with the value "Işık".
- Login**: A green button.
- Footer**: A link "Forgot your password? Click Here!"

Fig. 2 – Login.aspx page

Design of Login.aspx

Login.aspx is very simple web page; it contains only **two textboxes** for **first name** and **last name**, and a **Login** button as in Fig. 2. There will be no password check. Any user that completes this form will be considered as a logged in user.

When the **Login** button is clicked , **first name** and **last name** will be written into a cookie named **UserInfo** for **one month** and the page will be redirected to **Default.aspx** and **Default.aspx** will display **user info** on the left and a **list of products** on the right as in Fig. 3.

Table 1 – Product info

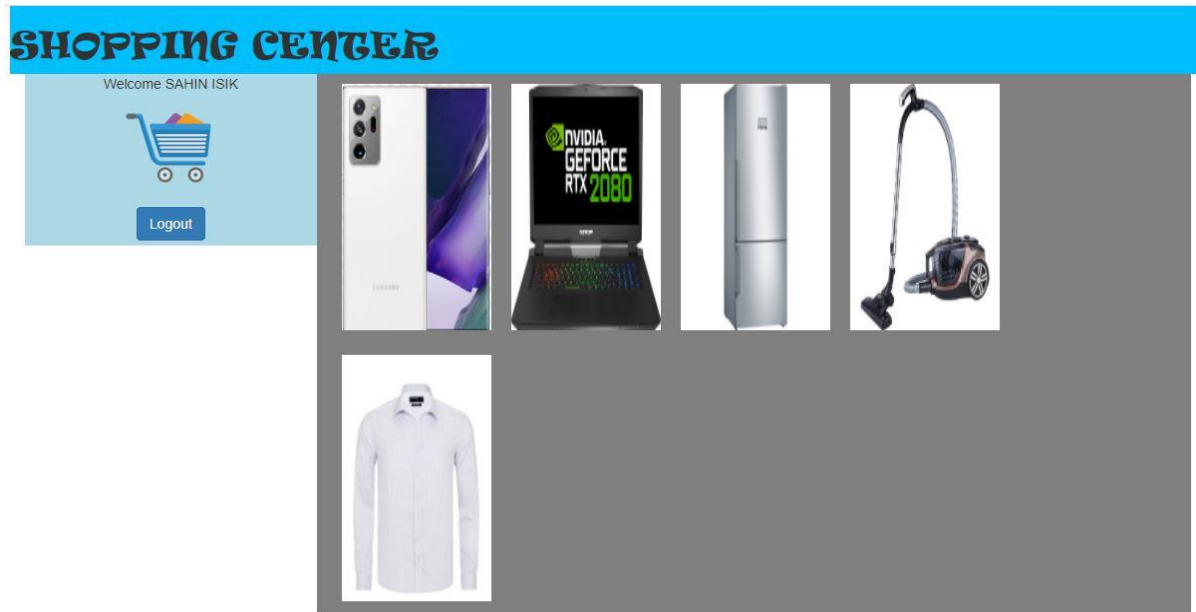


Fig. 3 – Default.aspx page after user is logged in. This just an example. You can use template.

Default.aspx should display a link to **Cart.aspx** and a **Logout** button on the left. When the **Logout** button is clicked, the **UserInfo** cookie should be removed and Fig. 1 should be displayed.

Default.aspx should display the titles of the products on the right with a **link** on each one. Each link should navigate to **ProductInfo.aspx** with query string named **id** (e.g. **ProductInfo.aspx?id=5**).

When **Default.aspx** page is first loaded, it should create five **Product** objects and put all them in to the **Session state**.

Design of ProductInfo.aspx

When a product title is clicked in **Default.aspx**, the information about the product should be displayed in **ProductInfo.aspx** page as in Fig. 4. Product ID should be taken from **query string** with name **id**.

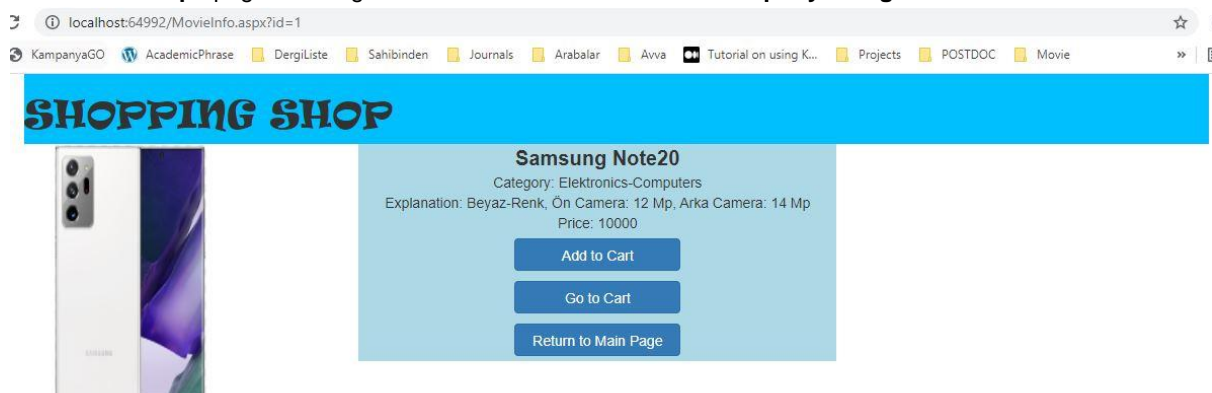


Fig. 4 – ProductInfo.aspx page with a valid product id



Fig. 5 – ProductInfo.aspx when an invalid ID is provided



Fig. 6 – ProductInfo.aspx when no query string is provided

If **ProductInfo.aspx** is requested without any query string or the product ID does not exist, then it should display an appropriate message as in Figures 5 and 6.

If a valid product ID is presented, then **ProductInfo.aspx** page should bring the desired product info from the Session state and display the product info with its cover image.

There also should be an **Add to Cart** button, a link to **Cart.aspx** and a link to **Default.aspx** as in Fig. 4.

When **Add to Cart** button is clicked, the **product** ID should be added in an **ArrayList** object in Session state that holds the selected product IDs. After the product is added into this **ArrayList** object, the user should be informed about this as in Fig. 7.

If that product is already in the cart, an appropriate message should be displayed as in Fig. 7 and Fig. 8

ProductInfo.aspx should also present links to **Cart.aspx** and **Default.aspx**.

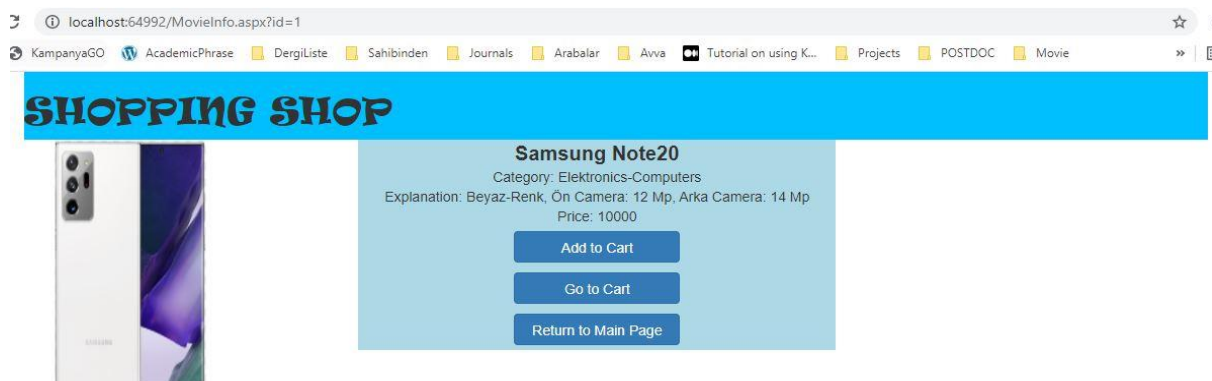


Fig. 7 – ProductInfo.aspx after Add to Cart button is clicked and the product is successfully added to the chart.

If the product is already in the chart, then display the following message at the below of button:
Product was already in the cart.



Fig. 8 – ProductInfo.aspx after Add to Cart button is clicked and it says that the Product is already in the shopping cart.

Basic Information About *ArrayList* Class

ArrayList class can be used to store items in an array in object-oriented manner. You can store product IDs in an **ArrayList** object. An **ArrayList** object is created by the new operator:

```
ArrayList SelectedProductIndices = new ArrayList();
```

An item can be added to the **ArrayList** by **Add()** method:

```
SelectedProductIndices.Add(id);
```

Here, **id** can be of any class. For the demonstration, we assume that **id** is of type **string**.

In order to check whether an item exists in the **ArrayList** or not, **Contains()** method can be used:

```
if (SelectedProductIndices.Contains(id))  
{  
    ...  
}
```

To remove an item, **Remove()** method can be used. If you want to remove all items, you can use **Clear()** method.


Design of Cart.aspx

Cart.aspx should display all products that are added to the shopping cart as. The IDs of the selected products should be taken from **Session** state as described above and information about the products should also be taken from the **Session** state which is put into the **Session** state by **Default.aspx** when it is first displayed. If there is no product in the shopping cart, a message should be displayed.

Cart.aspx page should also presents a link to the main page.

The tasks:

0- Choose one of the following topic for your project. You can also use another topic by sharing your ideas with course teacher.

Food Shop	
Grocery Shop	
Car Shop	
Shoes Shop	
Book Store	

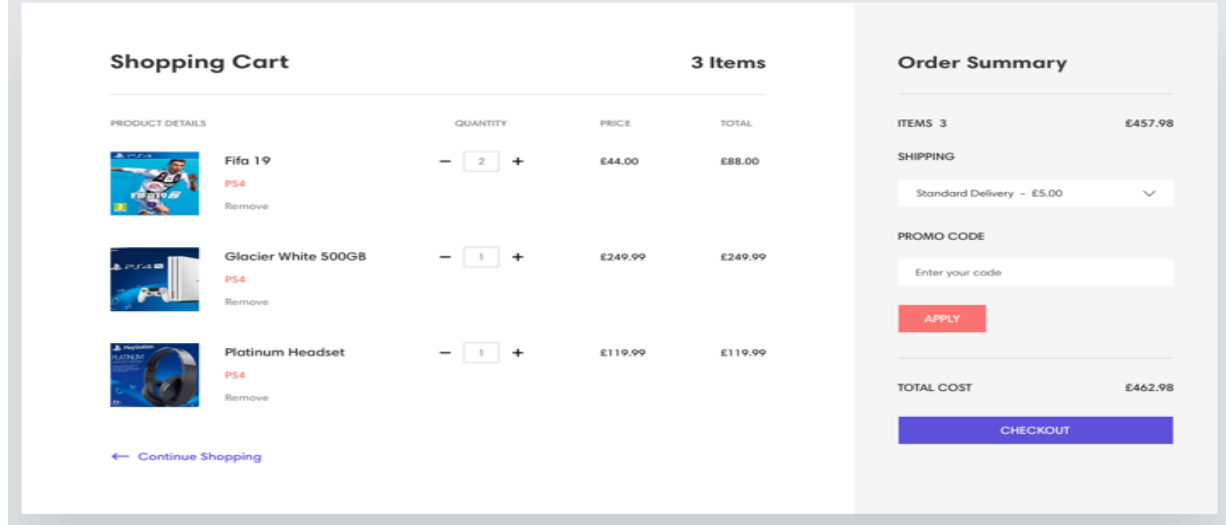
1- Implement a shopping website that covers the tables given database. You are expected implement all details given in tables. For example; show the quantity of product in ProductInfo.aspx. The database must be access (mdb).

2- Add rate bar for all products (ProductInfo.aspx). Keep Average rated for all products.



Also save the rating score of customer in Product Table.

3- You should list user cart as like below.



4- You are expected to google ads to your web site.

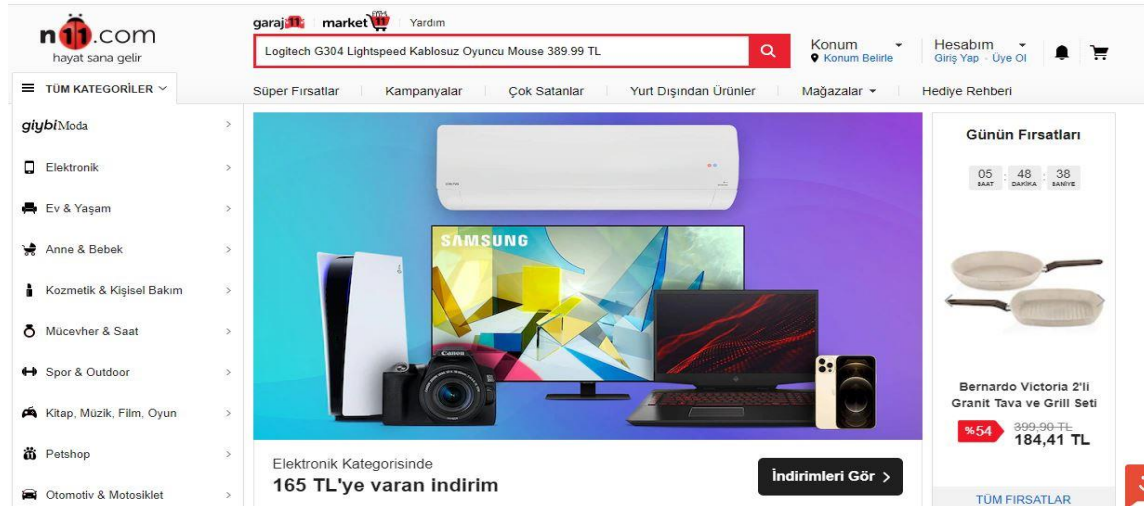
5- Your website should include at least 30 products and 10 customers.

6- You are expected to add comments to your code.

7- You are expected to log the errors.

8- You are expected to show the product names below the product images in Home.aspx. You have to make a good design for home.aspx. You can use templates.

9- You are expected show the categories and tabs in your home page.



10- You are expected show the search bar in your website.



11- The design of website will be graded over 50/100 points.

Submitting the Midterm

You have to prepare a small video (up to 5 minutes) to demonstrate your midterm exam.

Example; Name-Surname-Final.Zip