ASP.NET Fundamentals Exam - 16 August 2023

SoftUniBazar

Problems for the exam for the "ASP.NET Core Fundamentals" course @ SoftUni

SoftUniBazar is an online platform that the students of SoftUni use to sell and buy different useful items.

1. Technological Requirements and Overview

Use the provided skeleton – **SoftUniBazar_Skeleton**. All of the needed packages have been installed.

The Technological Requirements are ABSOLUTE. If you do not follow them, you will NOT be scored for other Requirements.

The provided skeleton consists of:

- Areas/Identity/Pages you have to scaffold Identity here
- **Controllers** you should implement the controllers logic here
- Data you should hold the entities models here
- Models you should implement the models here
- Views you are provided with the needed views. Your task is to implement some logic regarding the loggedin/logged-out user
- appsettings.json don't forget to change the your connection string
- **Program.cs** you should fulfil the security and password requirements here

NOTE: You should seed the database with provided in advance data regarding the Category entity. In order to do this, remove the comments from the block of code in the protected override void OnModelCreating(ModelBuilder builder) method of the DbContext.

NOTE: Don't forget to uncomment the code inside the views while you implement your logic.

Now that you know the **Technological Requirements**, let us see what the **Functional Requirements** are.

2. Identity Requirements

You should scaffold Identity and use the default IdentityUser.

Remove the unnecessary code from the Login.cshtml and Register.cshtml files and leave only the needed code in order for the app to be functioning correctly.

NOTE: Don't worry about the views' style - once you scaffold Identity and remove the unnecessary code, the Login and Register pages should look like shown below. Don't add any classes to the views of those two pages!

The **password** requirements for the **IdentityUser** are the following:

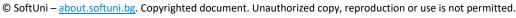
Require confirmed account: false

Require digits: **false**

• Require non-alphanumeric characters: false

Required uppercase letters: false

















3. Database Requirements

The Database of SoftUniBazar:

Δd

- Has Id a unique integer, Primary Key
- Has Name a string with min length 5 and max length 25 (required)
- Has **Description** a **string** with min length **15** and max length **250** (required)
- Has Price a decimal (required)
- Has OwnerId a string (required)
- Has Owner an IdentityUser (required)
- Has **ImageUrl** a **string** (**required**)
- Has CreatedOn a DateTime with format "yyyy-MM-dd H:mm" (required) (the DateTime format is recommended, if you are having troubles with this one, you are free to use another one)
- Has CategoryId an integer, foreign key (required)
- Has Category a Category (required)

Category

- Has Id a unique integer, Primary Key
- Has Name a string with min length 3 and max length 15 (required)
- Has Ads a collection of type Ad

AdBuyer

- BuyerId a string, Primary Key, foreign key (required)
- Buyer IdentityUser
- AdId an integer, Primary Key, foreign key (required)
- Ad Ad

Implement the entities with the correct datatypes and their relations.

Feel free to use the new syntax for realization of the many-to-many relation without a mapping table.















4. Page Requirements

Index Page (logged-out user)











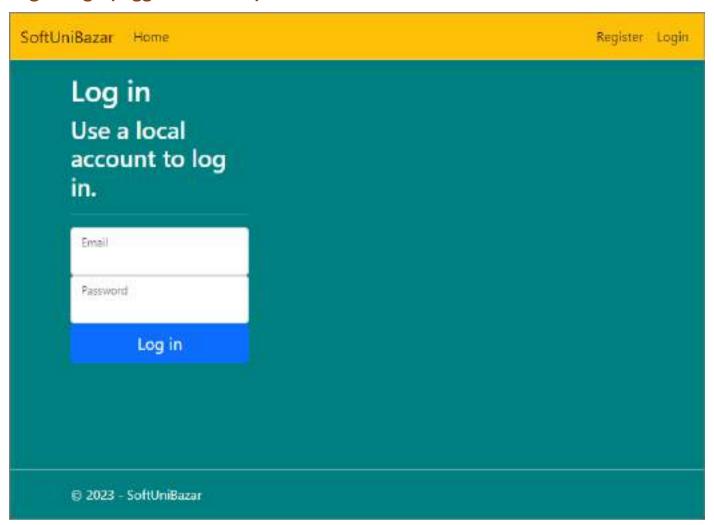








Login Page (logged-out user)









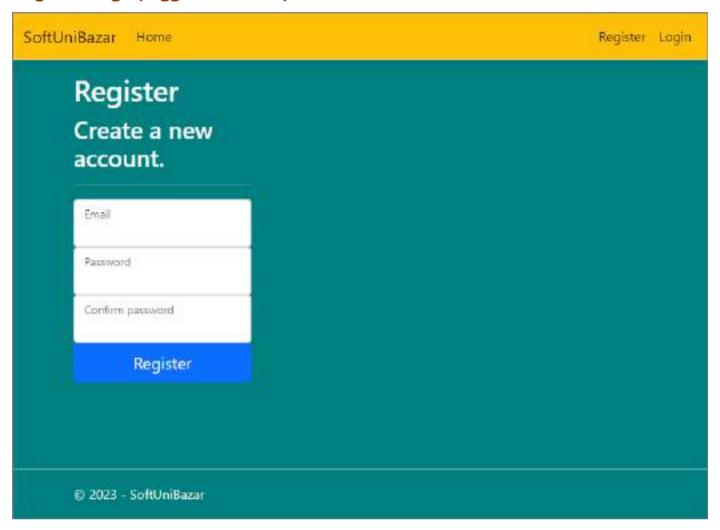








Register Page (logged-out user)











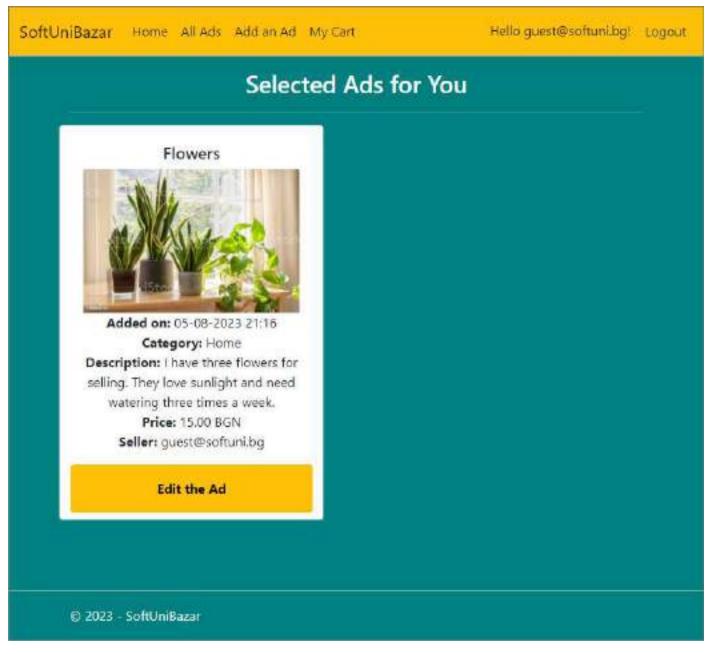








/Ad/All (logged-in user, creator of the ad)



NOTE: If the user is logged in and tries to go to the Home page, the application must redirect them to the /Ad/All.





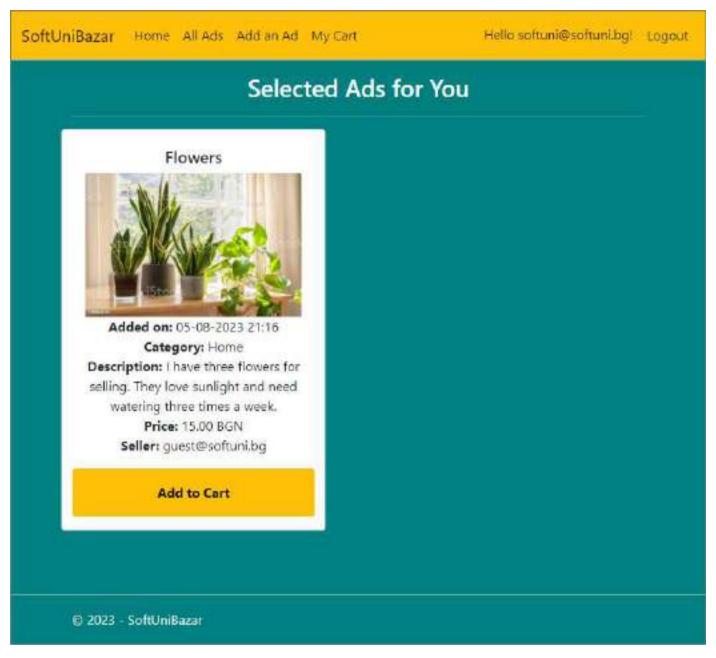








/Ad/All (logged-in user, not creator of an ad)



NOTE: If the user is logged in and tries to go to the Home page, the application must redirect them to the /Ad/All.





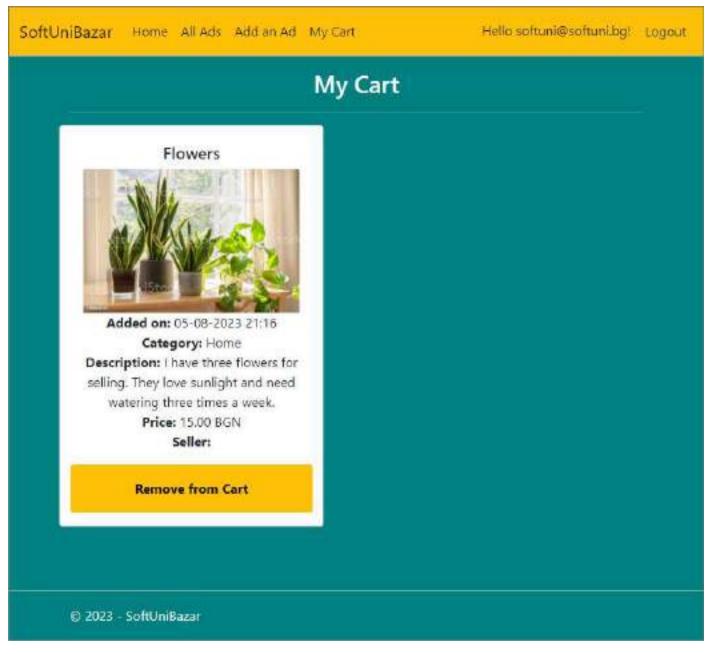








/Ad/Cart (logged-in user)



NOTE: the **DateTime** format is recommended, if you are having troubles with the one in the image, you are **free** to use **another** one.









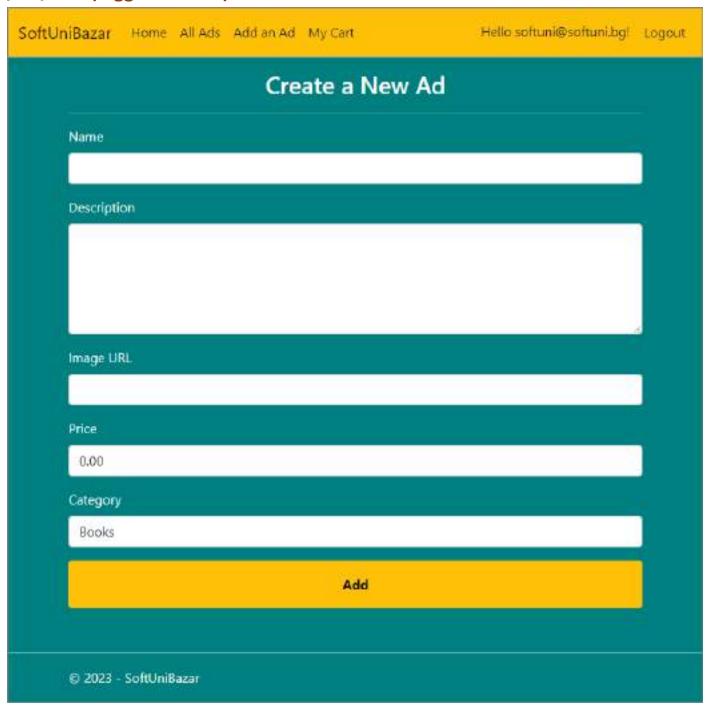








/Ad/Add (logged-in user)



NOTE: the **DateTime** format is recommended, if you are having troubles with the one in the image, you are **free** to use another one.









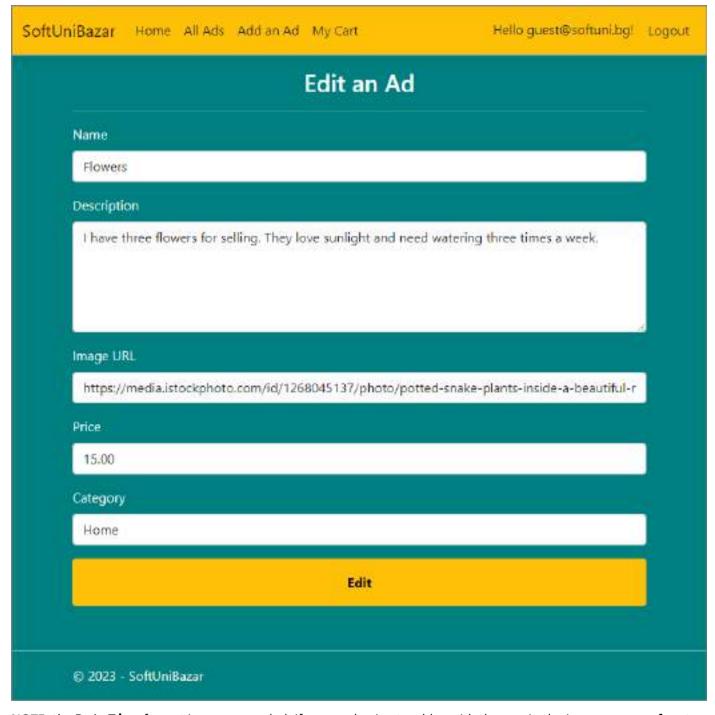








/Ad/Edit/{id} (logged-in user)



NOTE: the **DateTime** format is recommended, if you are having troubles with the one in the image, you are **free** to use another one.

NOTE: The little calendar icons in the example views are NOT required.

/Ad/AddToCart?id={id} (logged-in user)

Adds the selected ad to the user's collection of ads. If the ad is already in their collection, it shouldn't be added. If everything is successful, the user must be redirected to the their collection "/Ad/Cart" page.

/Ad/RemoveFromCart?id={id} (logged-in user)

Removes the selected ad from the user's collection of ads. If everything is successful, the user must be redirected to home "/Ad/All" page.















NOTE: The templates should look **EXACTLY** as shown above.

5. Functionality

The functionality of the **SoftUniBazar** Platform is very simple.

Users

Guests can Register, Login and view the Index Page.

Users can add ads and edit ads only they have added.

Users can see added ads by all users on the Home Page (/Ad/All).

If the user is the creator of the ad, they can see the [Edit] button. If the user is not the creator of the ad, they can add the ad to their cart.

Ads

Ads can be added by users. All created ads are visualized on the Home Page (/Ad/All).

Ads are visualized on the Home Page (/Ad/All) with some of their information.

Ads are visualized on the Home Page (/Ad/All) with one buttons:

- If the user **IS** the **creator** of the ad **[Edit]**;
- If the user IS NOT the creator of the ad [Add to Cart]

The [Edit] button displays a new page with a form, filled in with all of the info for the selected ad. Users can change this info and save it.

The [Add to Cart] button adds the ad to the user's collection of ad, unless it is already added.

Users have a **My Cart** page where only the **ads** in their collection are visualized.

• The [Remove from Cart] button removes the ad from the user's collection of ads.

Redirections

- Upon successful **Login** of an **IdentityUser**, you should be redirected to the /Ad/All.
- Upon successful **Creation** of an **Ad**, you should be redirected to the /**Ad/All**.
- Upon successful Adding an Ad to the User's collection, should be redirected to the /Ad/Cart.
- Upon successful **Editing** of an **Ad**, you should be redirected to the /Ad/All.
- Upon successful Removal of an Ad from the User's collection, should be redirected to the /Ad/All.
- If a User tries to add an already added ad to their collection, they should be redirected to /Ad/All (or just a page refresh).
- Upon successful **Logout** of a **User**, you should be redirected to the **Index Page**.
- If any of the validations in the POST forms don't pass, redirect to the same page (reload/refresh it).

6. Security

The Security section mainly describes access requirements. Configurations about which users can access specific functionalities and pages:

- **Guest** (not logged in) users can access the **Index** page.
- Guest (not logged in) users can access the Login page.
- **Guest** (not logged in) users can access the **Register** page.















- Guests (not logged in) cannot access Users-only pages.
- Users (logged in) cannot access Guest pages.
- Users (logged in) can access the Ad/Add page and functionality.
- Users (logged in) can access the Ad/Edit page and functionality.
- Users (logged in) can access the Ad/All page.
- **Users** (logged in) can access the My Cart page.
- Users (logged in) can access Logout functionality.
- Users (logged in) cannot access the Ad/Edit page of an Ad that have another user as a creator.

7. Code Quality

Make sure you provide the best architecture possible. Structure your code into different classes, follow the principles of high-quality code (SOLID). You will be scored for the Code Quality and Architecture of your project.

8. Scoring

Identity Requirements – 5 points

Database Requirements – 10 points

Template Requirements – 10 points

Functionality – 50 points

Security – 5 points

Code Quality – 10 points

Data Validation – 10 points

















