# Andrey's

**Andrey's** is a shop application. There are several requirements you must follow in the implementation.

## Technological Requirements

* Use the **SIS**
* Use **Entity Framework Core**

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

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

## Database Requirements

The **Database** of the **Andrey's** application needs to support **2 entities**:

### User

* Has an **Id – GUID-string, Primary key**
* Has a Username – a string with **min length 4** and **max length 10 (inclusive) (required)**
* Has a Password – a string with **min length** **6** and **max length 20 (inclusive)** - hashed in the database (**required**)
* Has an Email

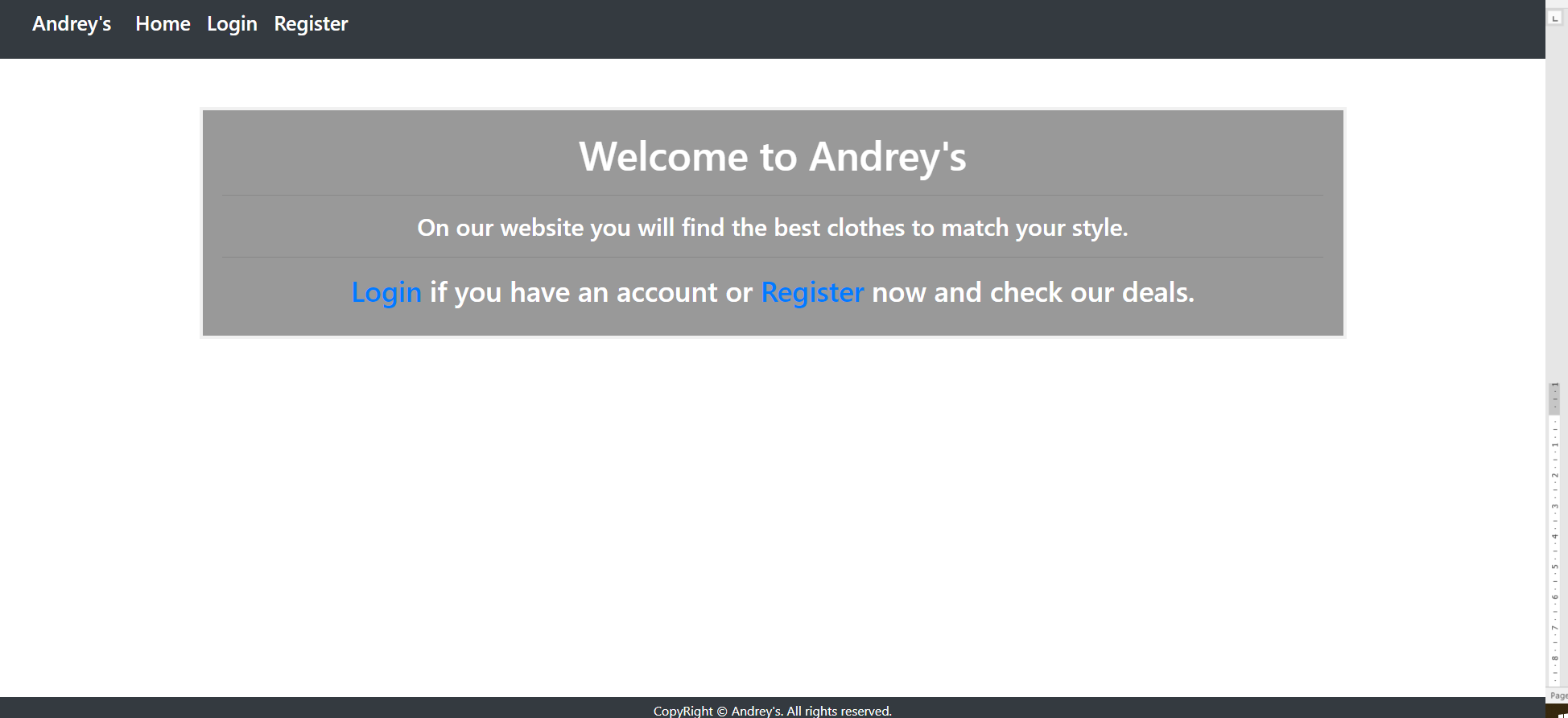
### Product

* Has an **Id – int, Primary key**
* Has a Name – a string with **min length 4** and **max length 20 (inclusive) (required)**
* Has a Description – a string with **max length 10 (inclusive)**
* Has a ImageUrl – a string
* Has a Price – a decimal **(required)**
* Has a Category – an Enum – option between (Shirt, Denim, Shorts, Jacket) **(required)**
* Has a Gender – an Enum – option between (Male and Female) **(required)**

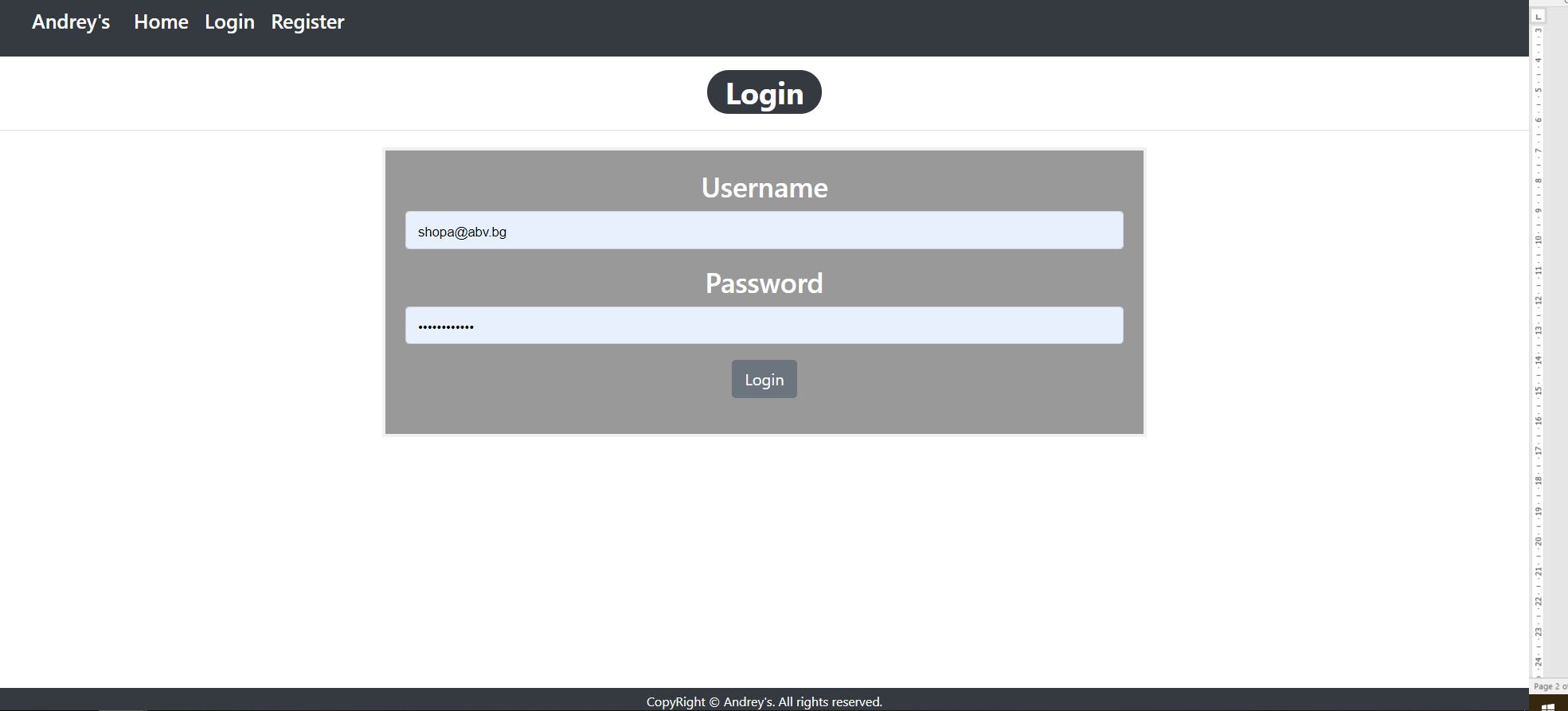
Implement the entities with the **correct datatypes** and their **relations**.

## Page Requirements

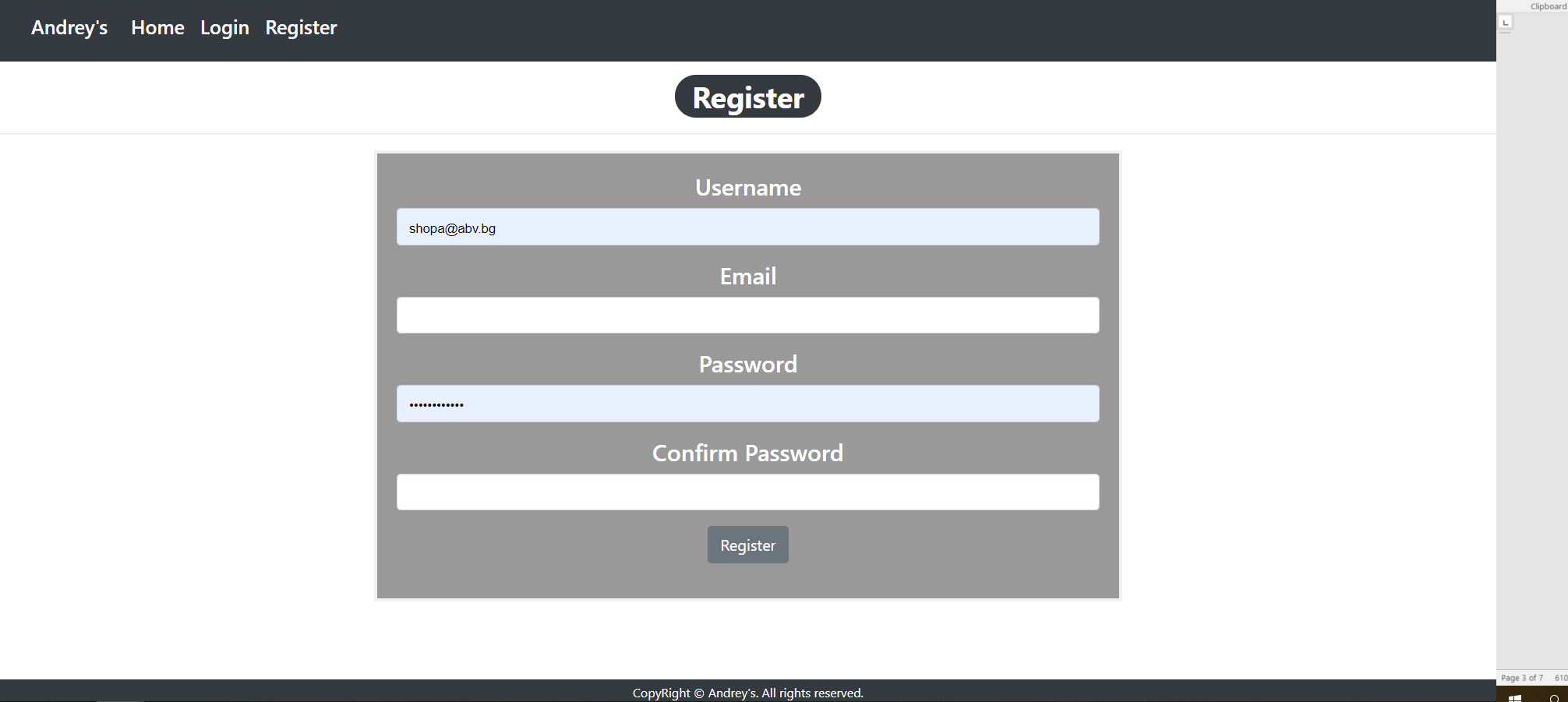
### Index Page (logged out user)



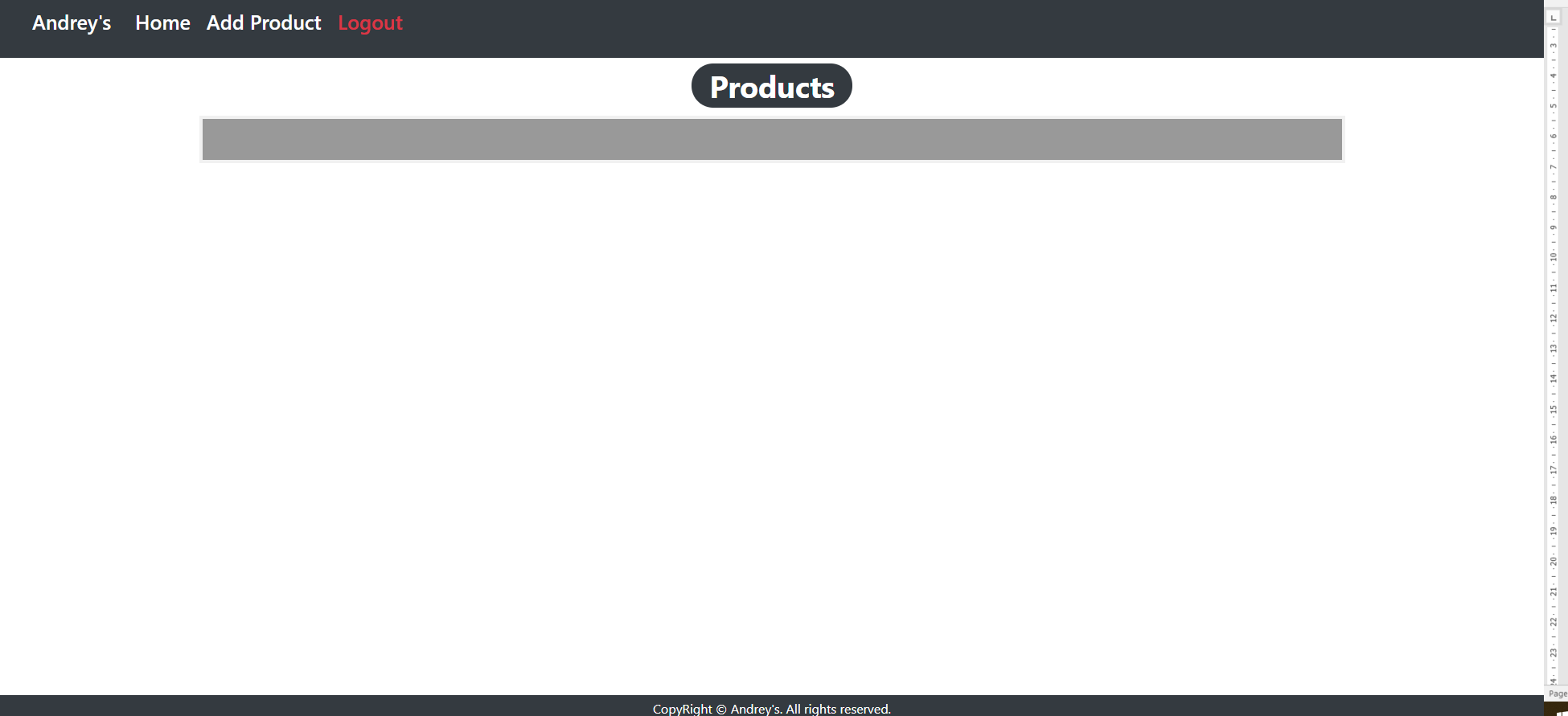
### Login Page (logged out user)

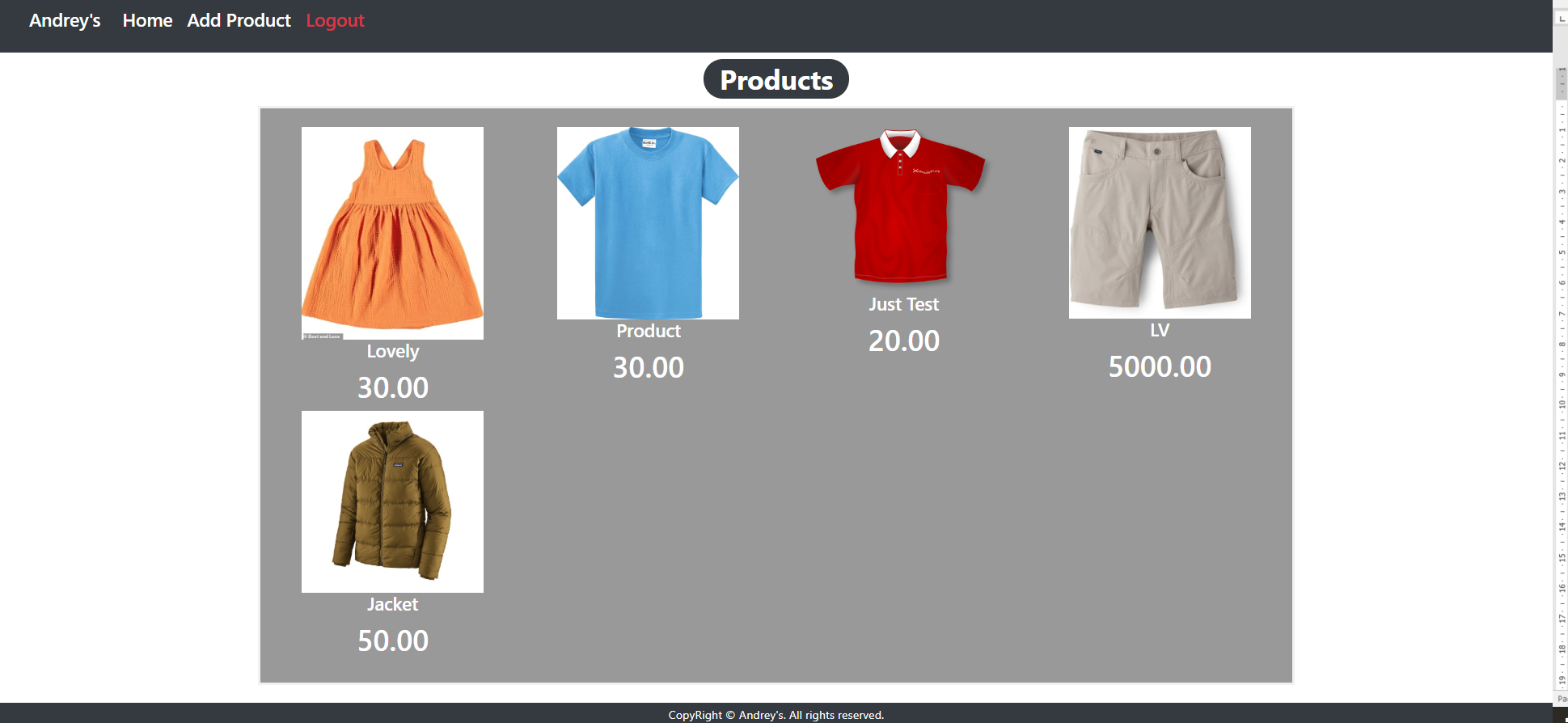


### Register Page (logged out user)

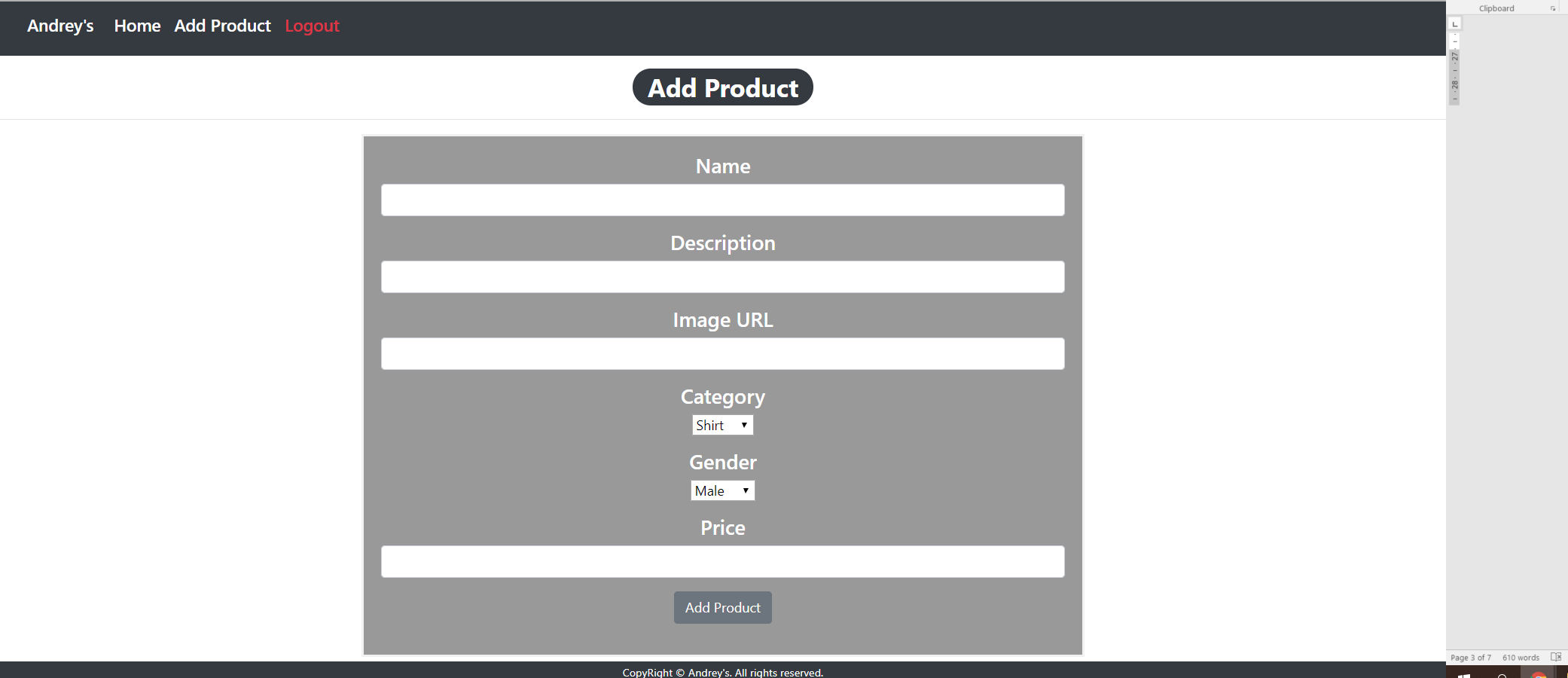


### Home Page (logged in user)

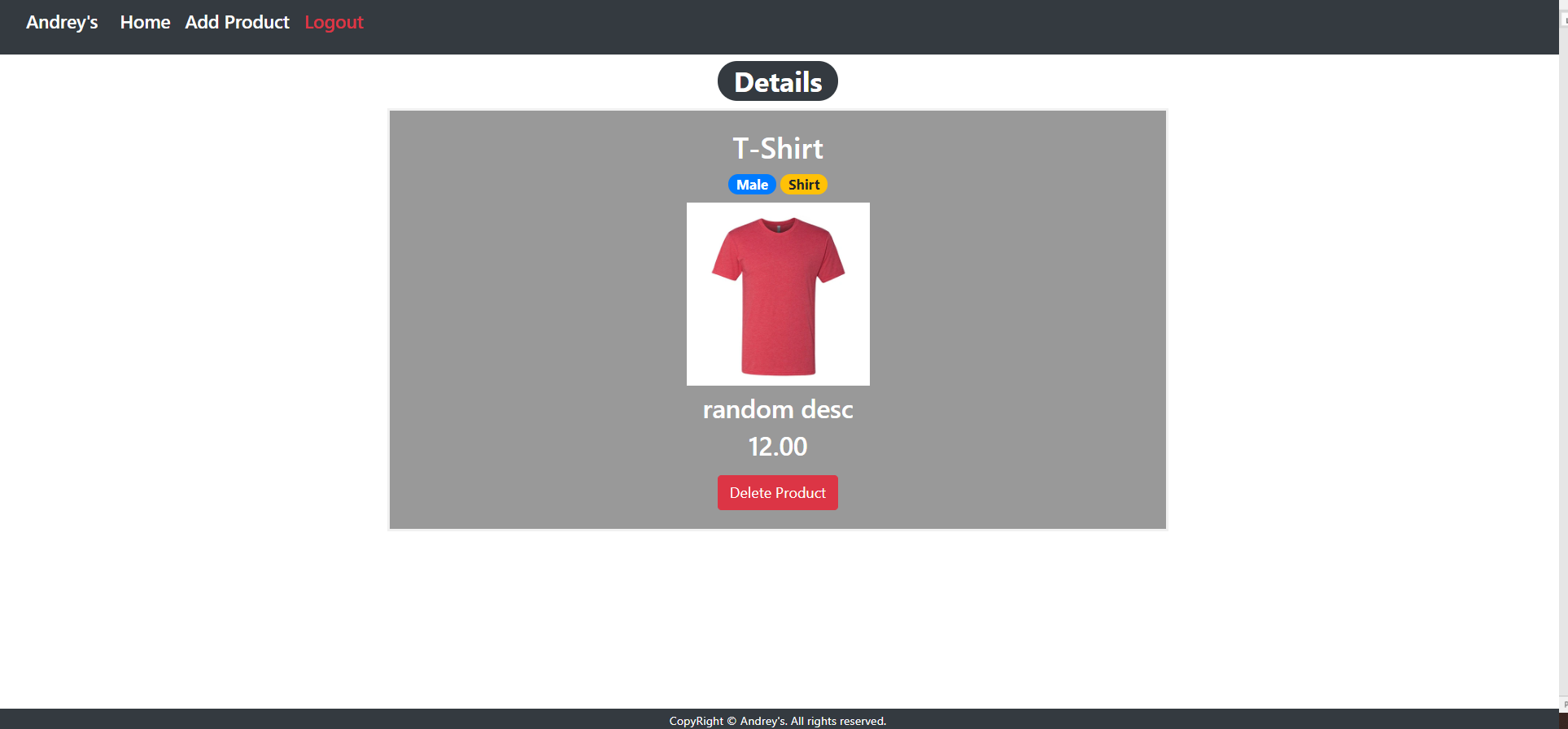




### Add Product (logged in user)



### Product Details Page (logged in user)



The templates have been given to you in the application skeleton, so make sure you implement the pages correctly.

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

**NOTE**: The templates do **NOT** **require** **additional** **CSS** for you to write. Only **bootstrap** and the **given css** are enough.

## Functional Requirements

The Functionality Requirements describe the functionality that the **Application** must support.

The **application** should provide Guest (not logged in) users with the functionality to login, register and **view** the Index page.

The **application** should provide Users (logged in) with the functionality to logout, **add a Product**, **view** **details** about a Product, **delete** a Product, and **view all** Products (Home page).

**Andrey's button** in navbar should redirect to appropriate **URL depending** on that if the user is logged in.

The **application** should provide **functionality** for **adding products** with **category (Shirt, Denim, Shorts, Jacket) and gender (Male, Female)**

The **image** should be the one corresponding to the URL given when the Product is added. When you click on the image on **Home** **Page**, you should be redirected to **Details** **page**. On the **Details** **page** you should have **Delete** **button**.

After **deleting** or **creating** a Product, redirect to the **Home Page**.

The **application** should **store** its **data** into a MS SQL database, using Entity Framework Core.

## Security Requirements

The Security Requirements are mainly access requirements. Configurations about which users can access specific functionalities and pages.

* Guest (not logged in) users can access Index page.
* Guest (not logged in) users can access Login page.
* Guest (not logged in) users can access Register page.
* Users (logged in) can access Home page.
* Users (logged in) can access Product Add page.
* Users (logged in) can access Product Details page.
* Users (logged in) can access Logout functionality.

## 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.

## Data Validation

* In case of invalid data during the **login**, redirect to "**/Users/Login**"
* In case of invalid data during the **registration**, redirect to "**/Users/Register**"
* In case of invalid data during the **product creation**, redirect to "**/Products/Add**"

## Scoring

Database Requirements – 10 points.

Template Requirements – 10 points.

Functionality – 50 points.

Security – 10 points.

Application Structure – 10 points.

Data Validation – 10 points.