Shipping List Application

Regular Exam [Spring Fundamentals]

The **Shopping List** Application is here to help us keep in mind our shopping needs. The functionality is simple. When a user thinks of something important, they log in and add it to existing ones. So, when a person goes to the store, they have a clear idea of exactly what to buy. So, our little app saves a lot of family scandals.

There are several requirements you must follow in the implementation:

1. Database Requirements

The **Database** of the **Shopping List** application needs to support **3 entities**:

User

- **Id** – Accepts **UUID-String** or **Long** values
- **Username**
  - The **length** of the **values** should be **between 3** and **20** characters long (both numbers are **INCLUSIVE**)
  - The values should be **unique** in the database
- **Password**
  - The **length** of the **values** should be **between 3** and **20** characters long (**INCLUSIVE**)
- **Email**
  - The values should contain a **'@'** symbol)
  - The values should be **unique** in the database

Product

- **Id** – Accepts **UUID-String** or **Long** values
- **Name**
  - The **length** of the **values** should be **between 3** and **20** characters long (both numbers are **INCLUSIVE**)
  - The values should be **unique** in the database
- **Description**
  - The **length** of the **values** should be **at least 5** characters long
- **Price**
  - The values must be a **positive** numbers
- **Needed Before**
  - **Date and Time** values, that **cannot** be in the **future**
- **Category**
  - One product **has one** category and one category **can have many** products

Category

- **Id** – Accepts **UUID-String** or **Long** values
- **Style name**
  - The values should be **unique** in the database
  - an option between (**FOOD**, **DRINK**, **HOUSEHOLD** and **OTHER**)
- **Description**
  - Fell free to add some description to every classification

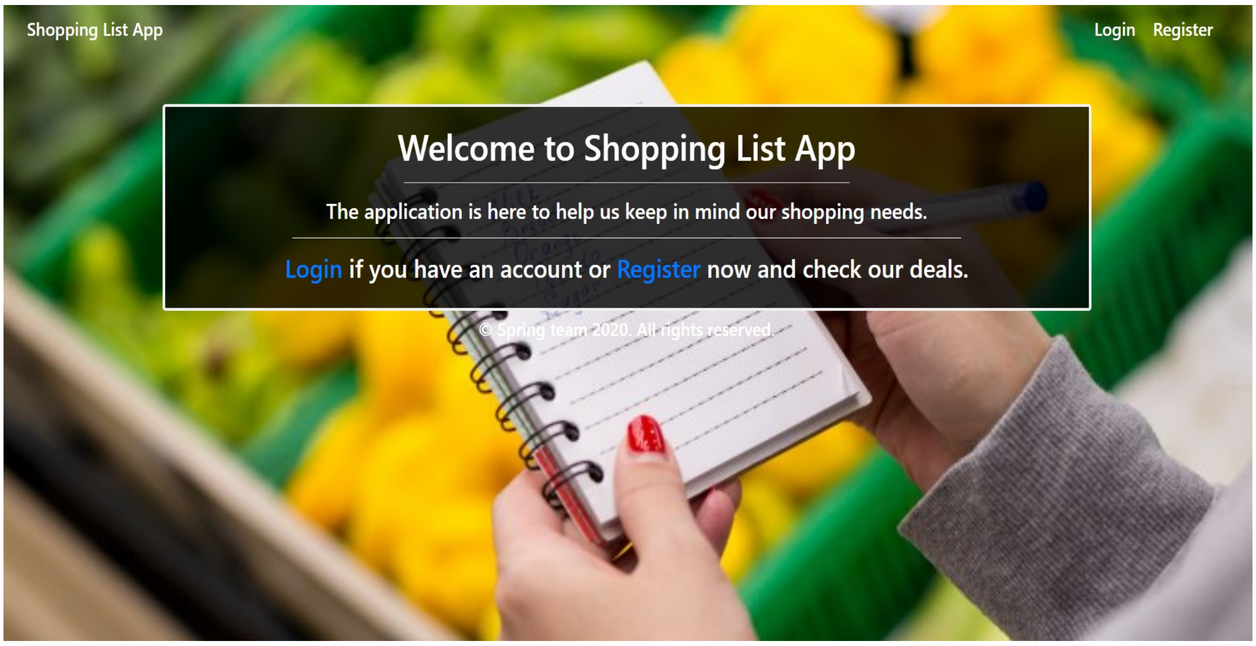
Nullable/Empty/Blank values are not allowed unless explicitly mentioned. Implement the entities with the **correct data types** and implement the **repositories** for them.

2. Initialize categories

- Implement a method that checks (when app started) if the database does not have any category and initialize them
  - You are free to do this in some different ways
  - You can skip the description if you want

3. Page Requirements

Index Page (logged out user)



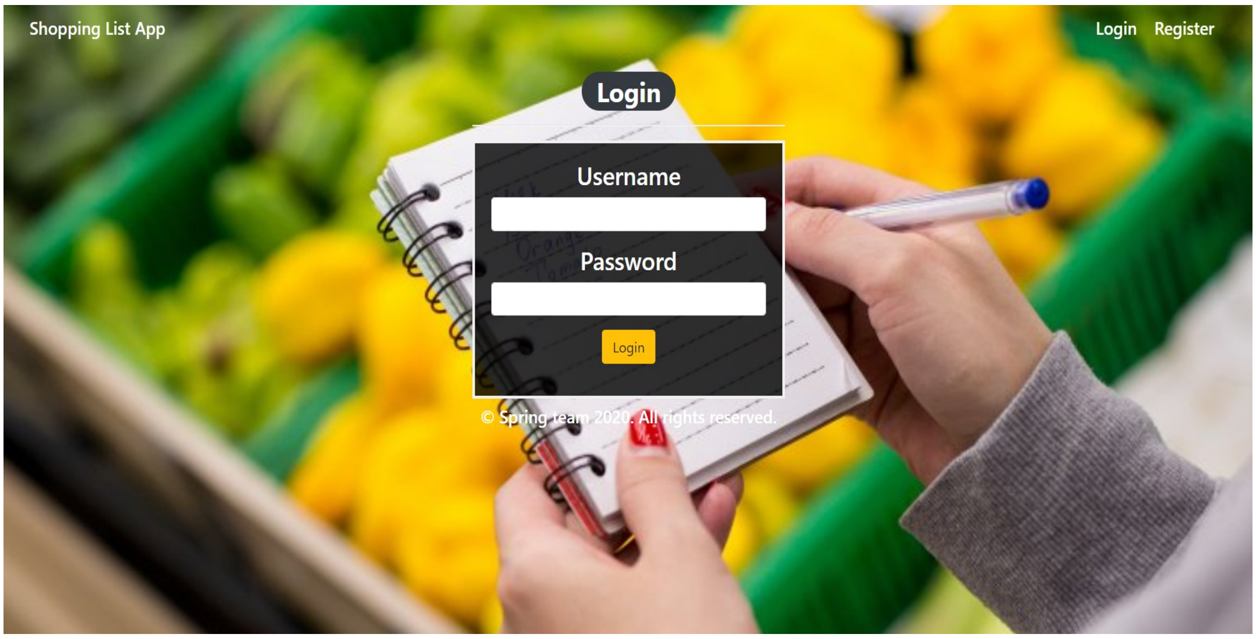
Welcome to Shopping List App

The application is here to help us keep in mind our shopping needs.

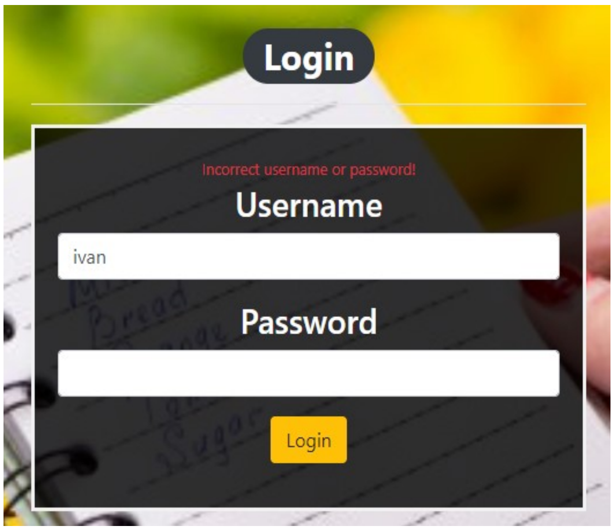
Login if you have an account or Register now and check our deals.

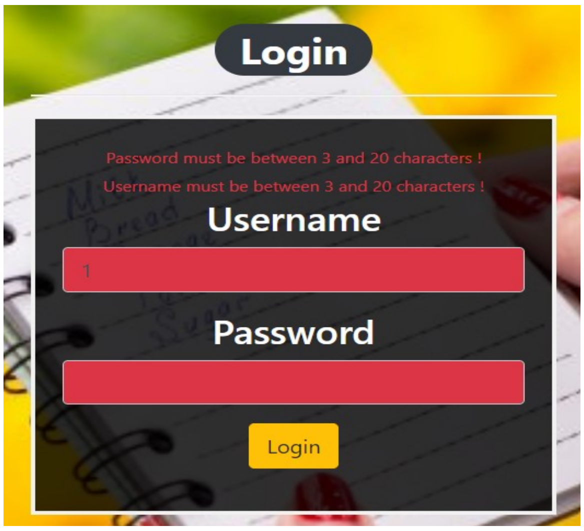
© Spring team 2020. All rights reserved.

Login Page (logged out user)

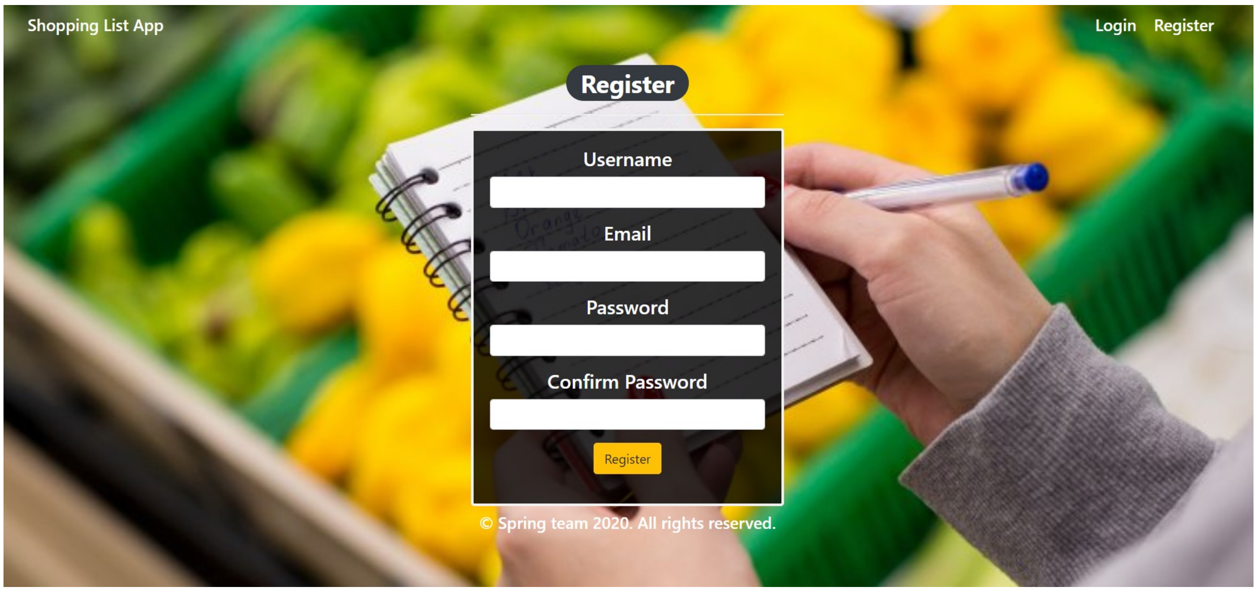


Login Page validations

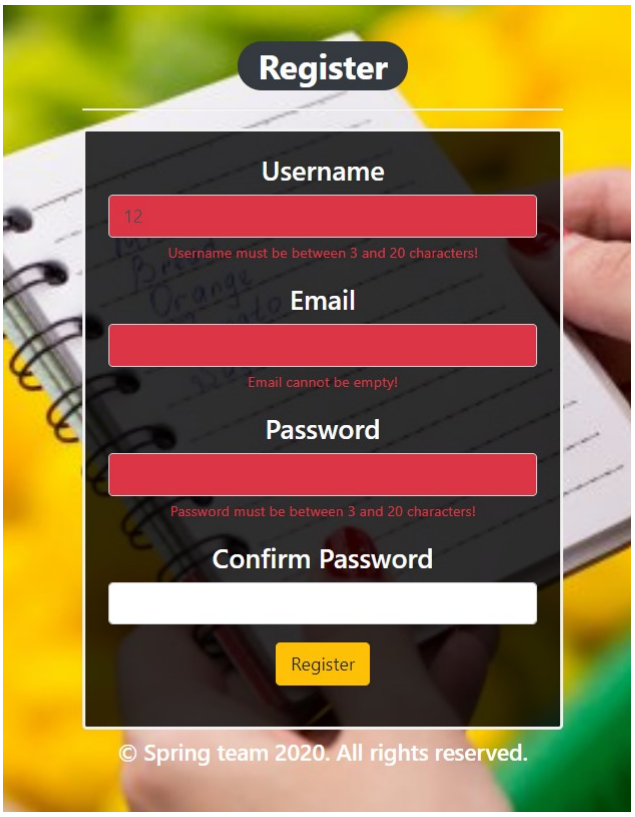



Login
Password must be between 3 and 20 characters !
Username must be between 3 and 20 characters !
Username
1
Password
Login

Registration Page (logged out user)

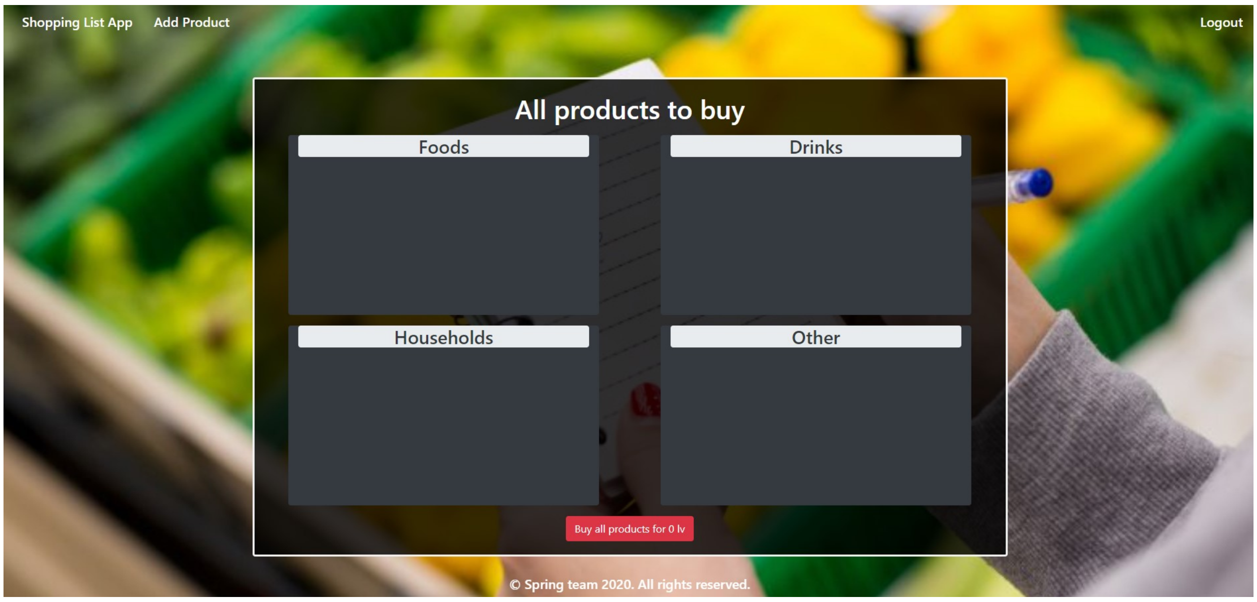


Registration Page validations

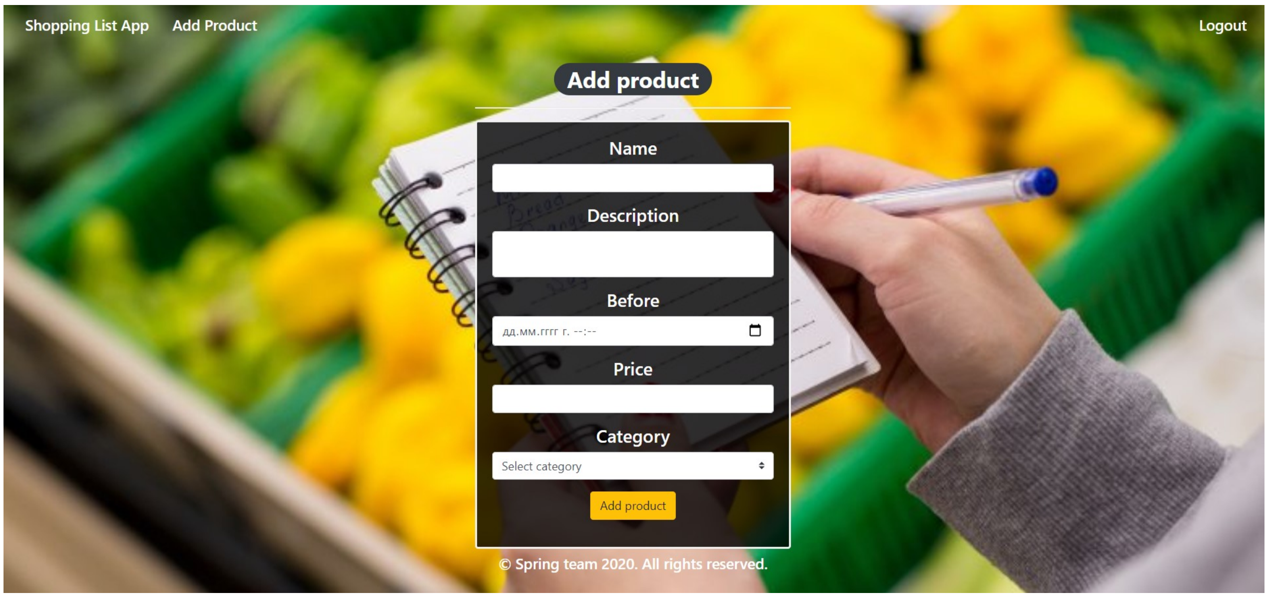

Register
Username
12
Username must be between 3 and 20 characters!
Email
Email cannot be empty!
Password
Password must be between 3 and 20 characters!
Confirm Password
Register
© Spring team 2020. All rights reserved.

Home Page (without having any products)

- The home page should visualize **all of the products** from the database.



Add products


Shopping List App   Add Product   Logout

Add product

Name

Description

Before

дд.мм.гггг г. --:--

Price

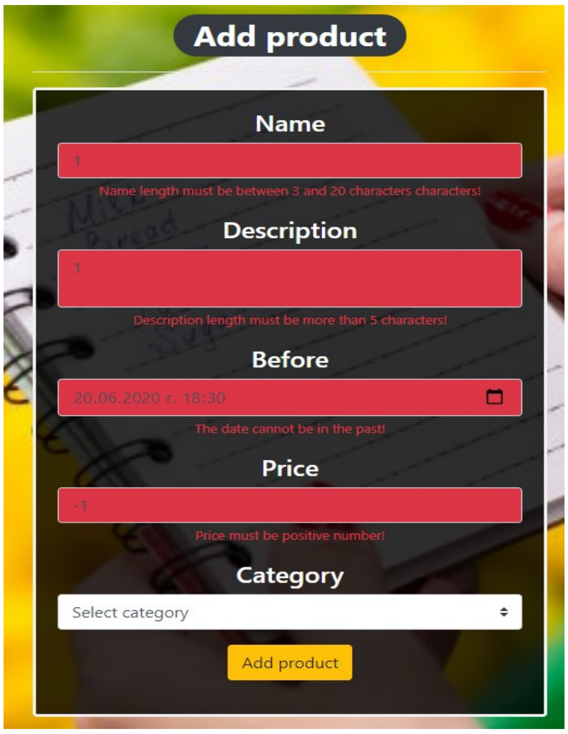
Category

Select category

Add product

© Spring team 2020. All rights reserved.

Add Offer Validation


Add product

Name

1

Name length must be between 3 and 20 characters characters!

Description

1

Description length must be more than 5 characters!

Before

20.06.2020 г. 18:30

The date cannot be in the past!

Price

-1

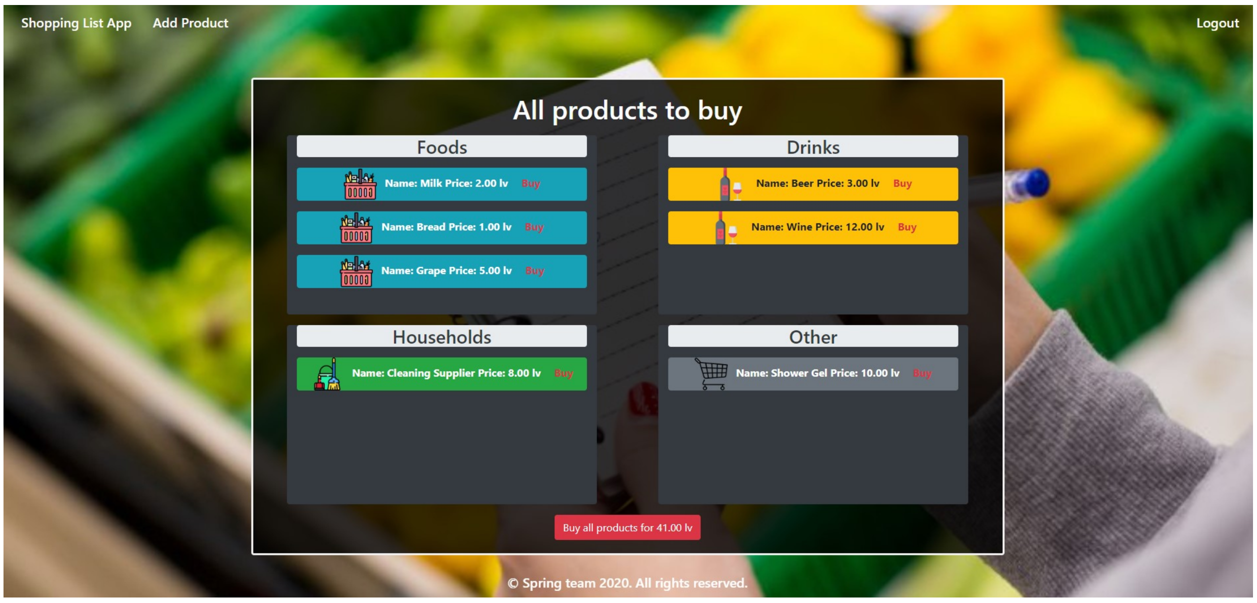
Price must be positive number!

Category

Select category

Add product

Home Page (with products)


Shopping List App
Add Product
Logout
All products to buy
Foods
Name: Milk Price: 2.00 lv Buy
Name: Bread Price: 1.00 lv Buy
Name: Grape Price: 5.00 lv Buy
Drinks
Name: Beer Price: 3.00 lv Buy
Name: Wine Price: 12.00 lv Buy
Households
Name: Cleaning Supplier Price: 8.00 lv Buy
Other
Name: Shower Gel Price: 10.00 lv Buy
Buy all products for 41.00 lv
© Spring team 2020. All rights reserved.

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

NOTES:

- The templates should look **EXACTLY** as shown above.
- The templates do **NOT require additional CSS** for you to write. Only **bootstrap** and the **given CSS** are enough.

4. 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 log in, register and view the Index page.

The application should provide Users (logged in) with the functionality to **log out, add a new product (Add product page), view all products (Home page) and Buy a single product** or **Buy all products.**

Shopping List App 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** (FOOD, DRINK, HOUSEHOLD or OTHER) and **buying** one or more of them.

The **Buy** all products button shows the **sum** of **all added products** prices. In addition to that you can display a total sum for all products grouped by a category (not required - only optional).

The **product** should be separated into different divs according to their categories. The **image** also depends on the item's category.

When the user clicks on the **Buy button** of some item, he buys it. You need to **delete** this **item** and **redirect it** to the **home** page. When he clicks on **Buy all** products, just **delete all** products in DB and again **redirect** to the **home** page.

The application should store its data in a **MySQL** database.

5. 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;
  - **Login** page;
  - **Register** page.
- **Users** (logged in) can access:
  - **Home** page;
  - **Add Product** page;
  - **Logout** functionality.