# Spring Fundamentals Exam

# Shopping List Application

Exam for the "Spring Fundamentals" course @ SoftUni.

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:

## Database Requirements

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

### User

* Has an **Id – UUID-string or Long**
* Has a Username (unique)
  + Username length must be between 3 and 20 characters (inclusive 3 and 20).
* Has a Password
  + Password length must be between 3 and 20 characters (inclusive 3 and 20).
* Has an Email
  + Must contains '@'.
  + Cannot be null.

### Product

* Has an **Id – UUID-string or Long**
* Has a Name (unique)
  + Name length must be between 3 and 20 characters (inclusive 3 and 20).
* Has a Description
  + Description min length must be minimum 5(inclusive) characters
* Has a Price
  + The price must be a positive number
* Has a Needed Before
  + Date and Time, that cannot be in the past
* Has a Category
  + Category cannot be null.

### Category

* Has an **Id** – **UUID-string or Long**
* Has a Name (unique)
  + an option between (Food, Drink, Household, Other)
* Has a Description

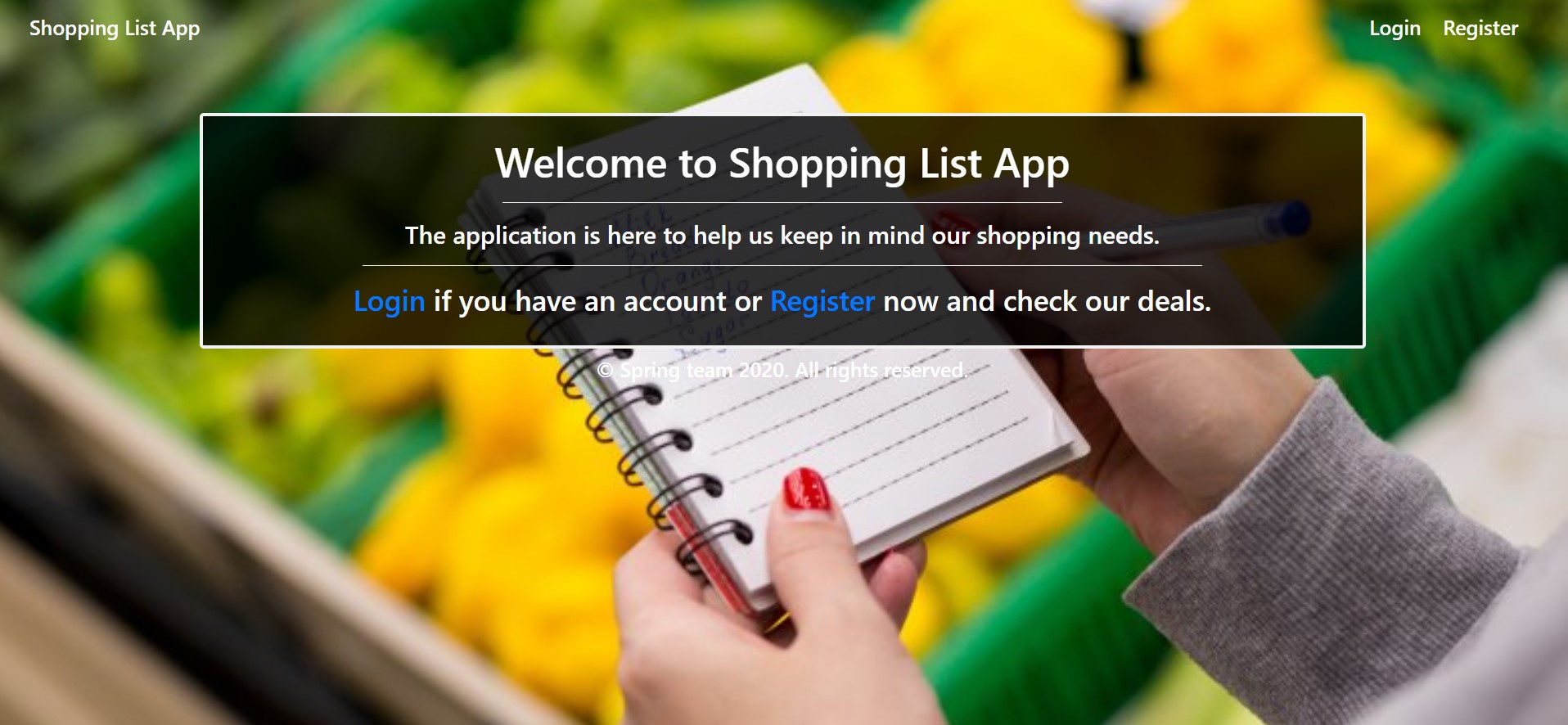
Implement the entities with the **correct data types** and implement **repositories** for them.

## Initialize categories

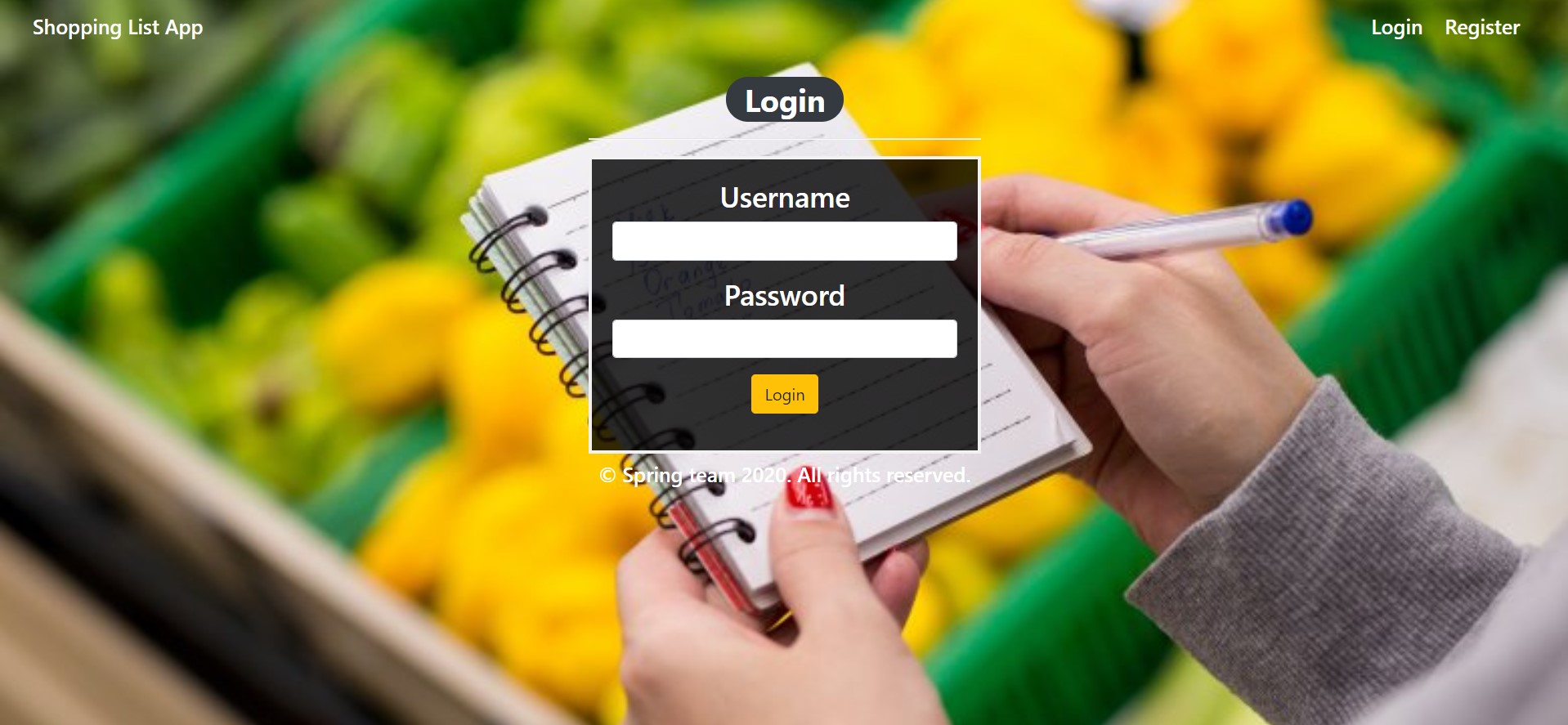
* Implement a method that checks (when the app started) if the database does not have categories and   
  initialize them
  + You are free to do it in different ways.

## Page Requirements

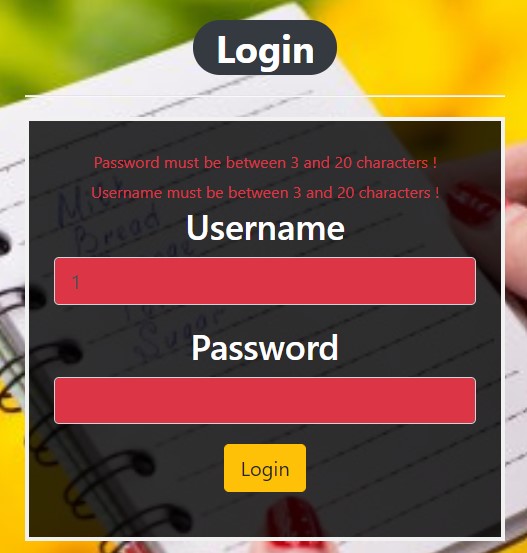
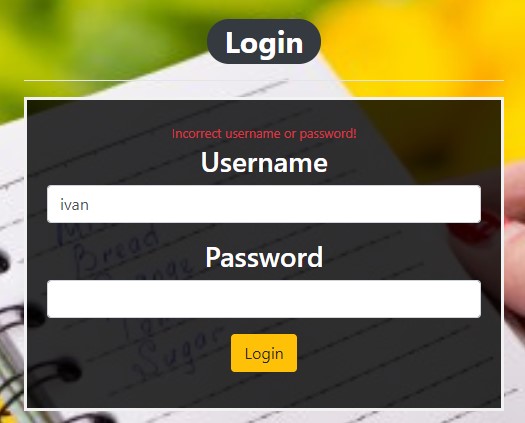
### Index Page (logged out user)



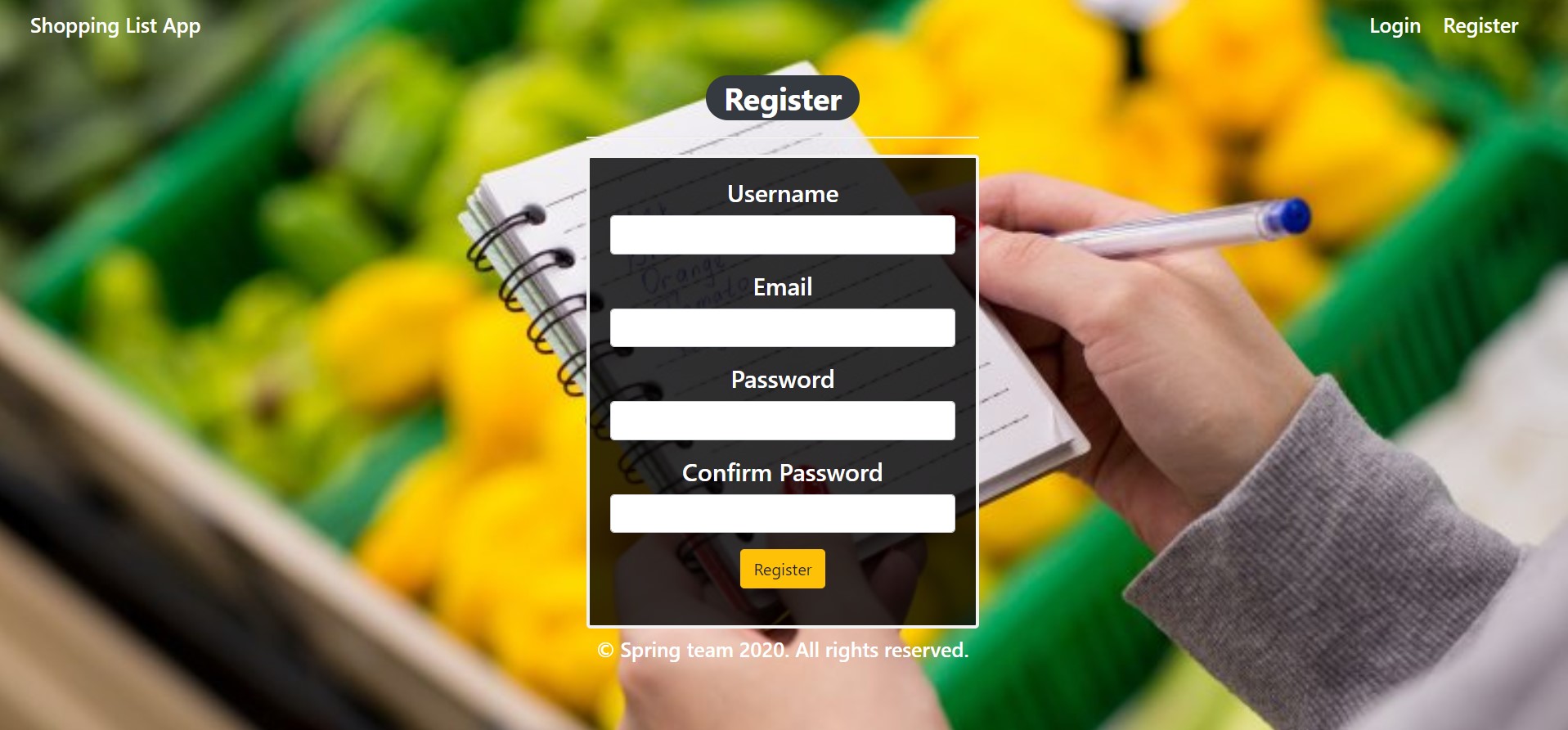
### Login Page (logged out user)



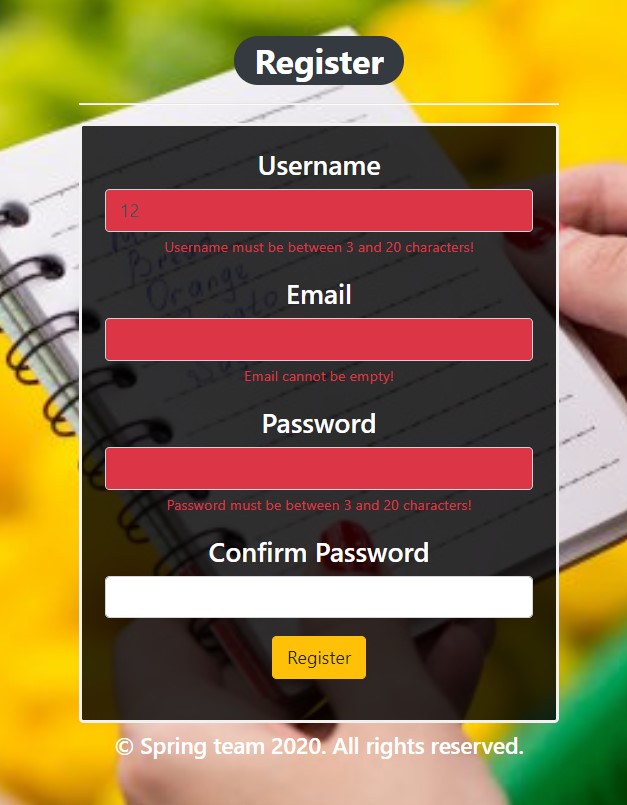
### Login Page validations



### Register Page (logged out user)

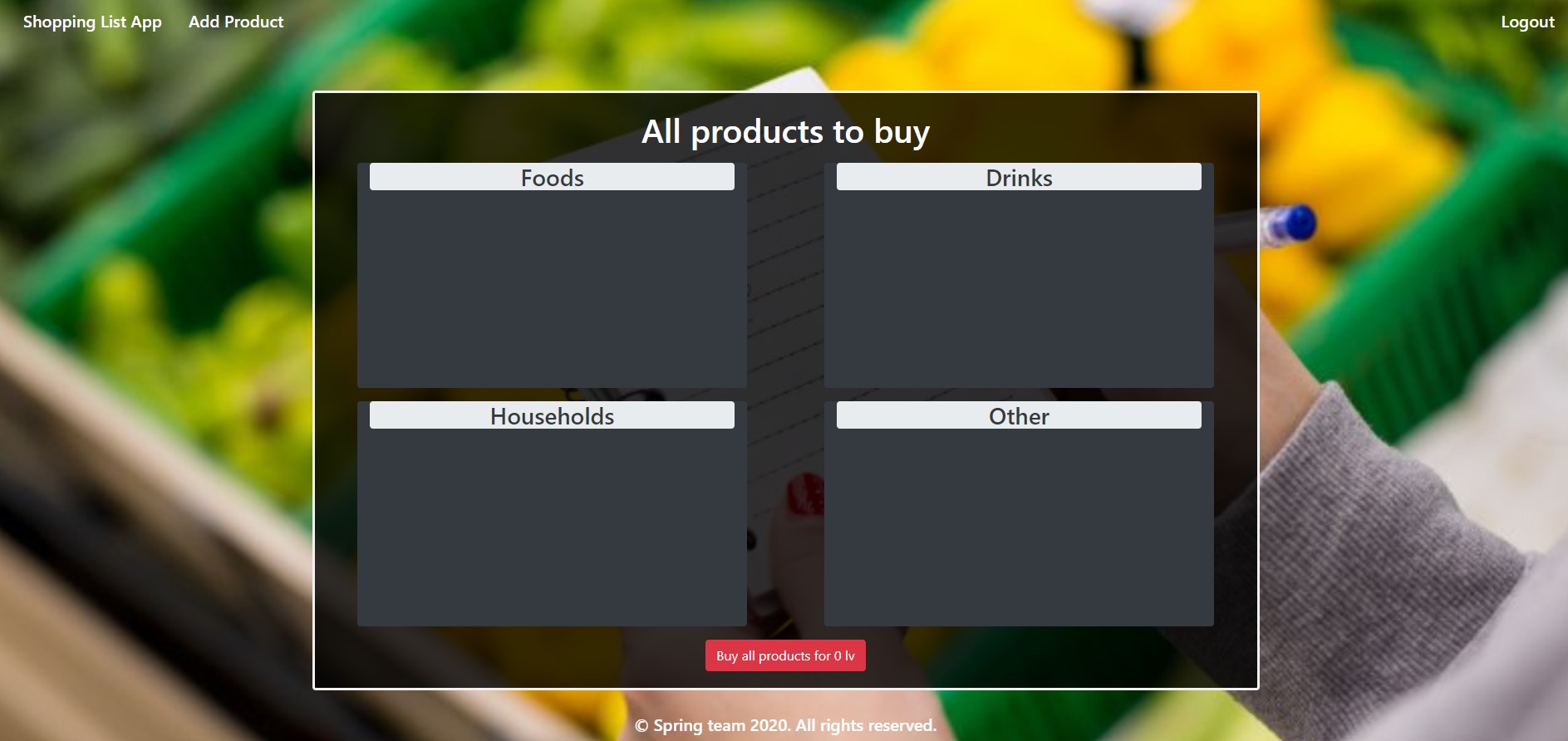


### Register Page validations

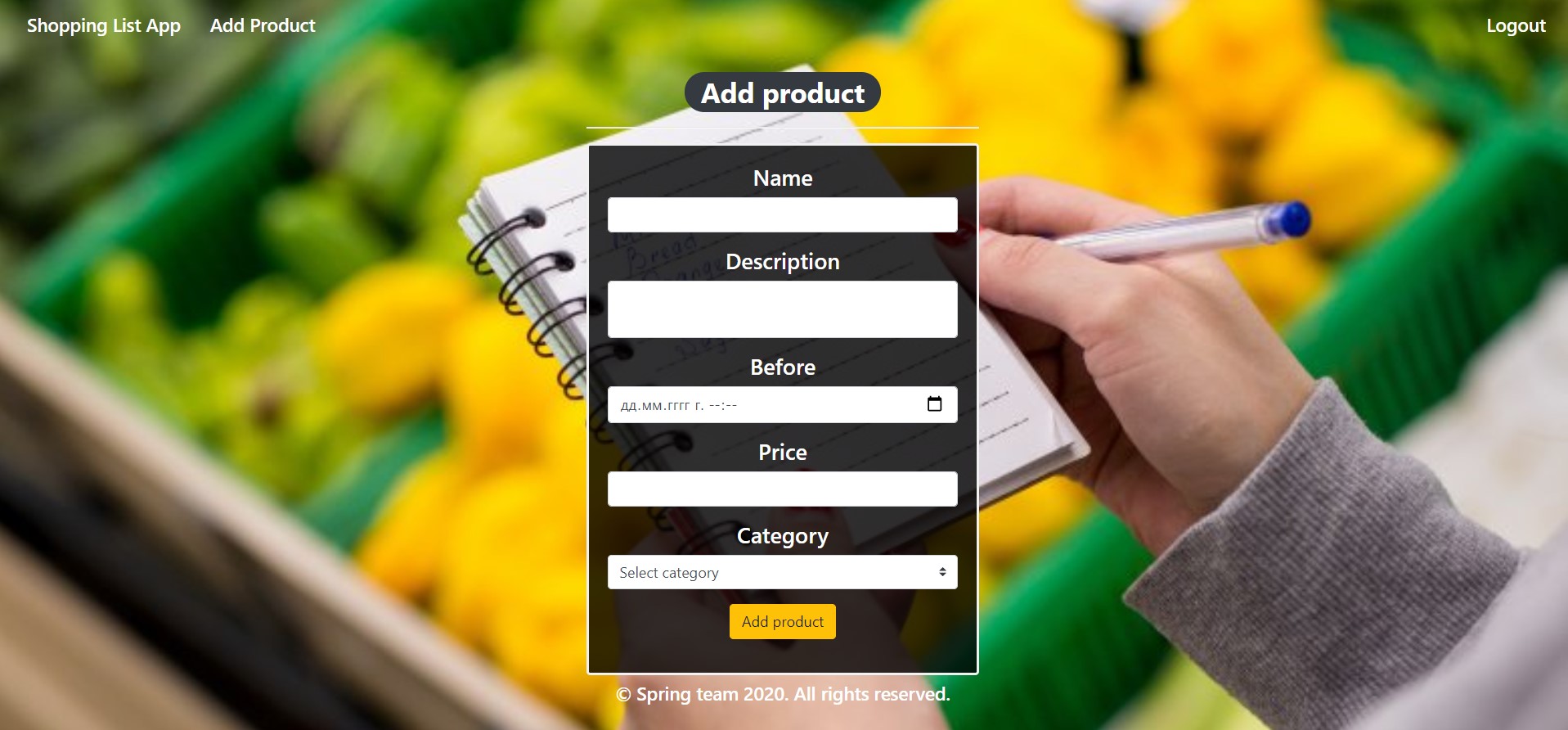


### Home Page (without having any products)

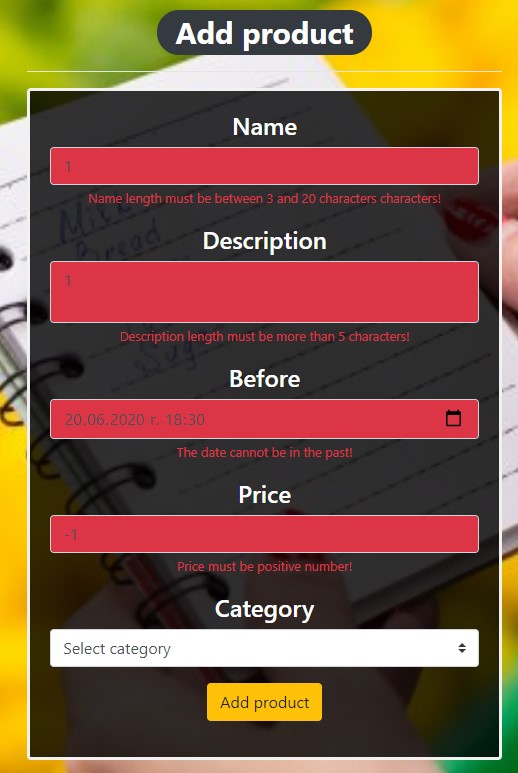
* Note: The home page should visualize **all of the products** from the database



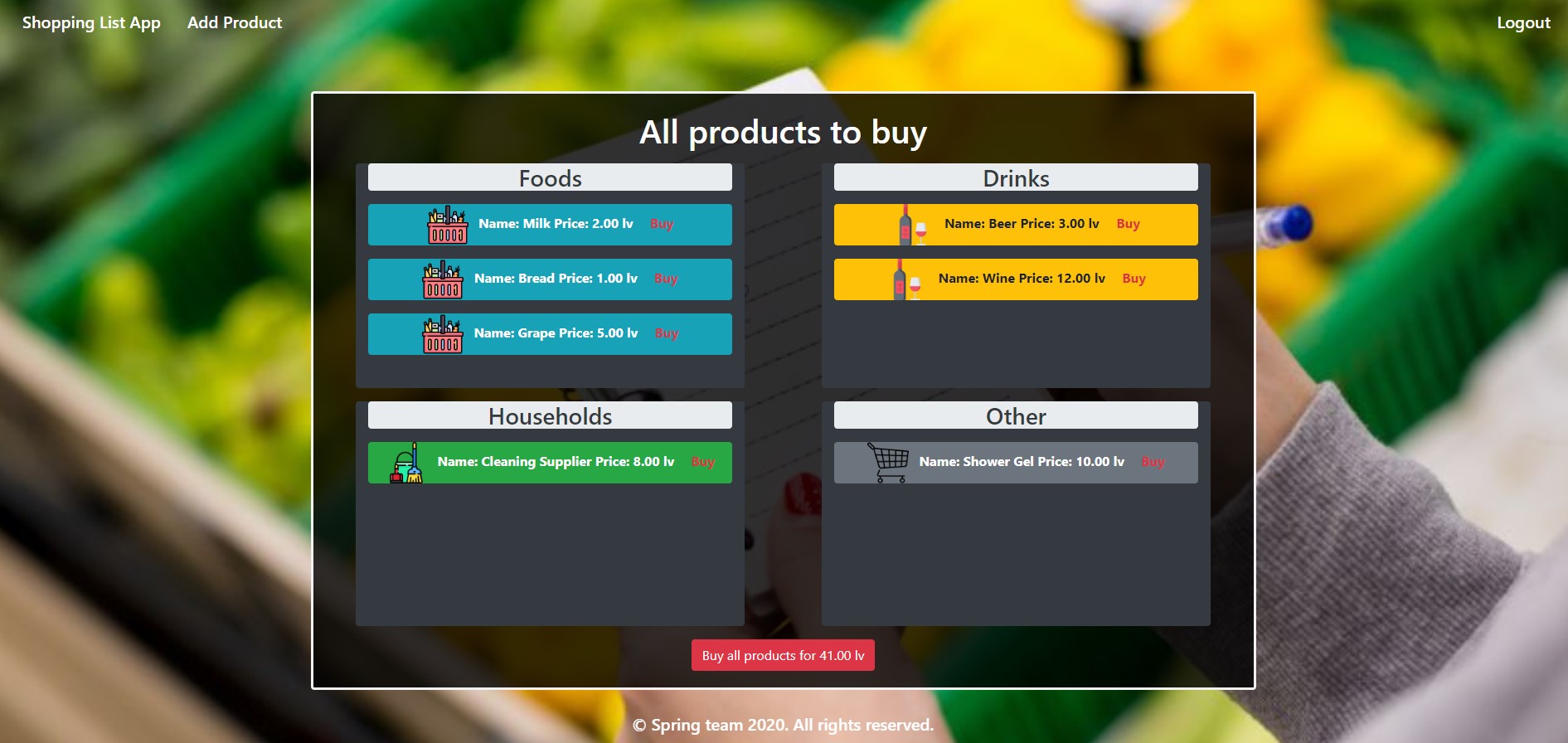
### Add products



### Add products validation



### Home Page (with products)



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 all** Products (Home page) and **Buy a single one** from products **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. / round ?

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.

## 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 Add Product page.
* Users (logged in) can access Logout functionality.

## Scoring

### Database – 10 points.

### Pages – 25 points.

### Functionality – 35 points.

### Security – 5 points.

### Validations – 15 points.

### Code Quality – 10 points.