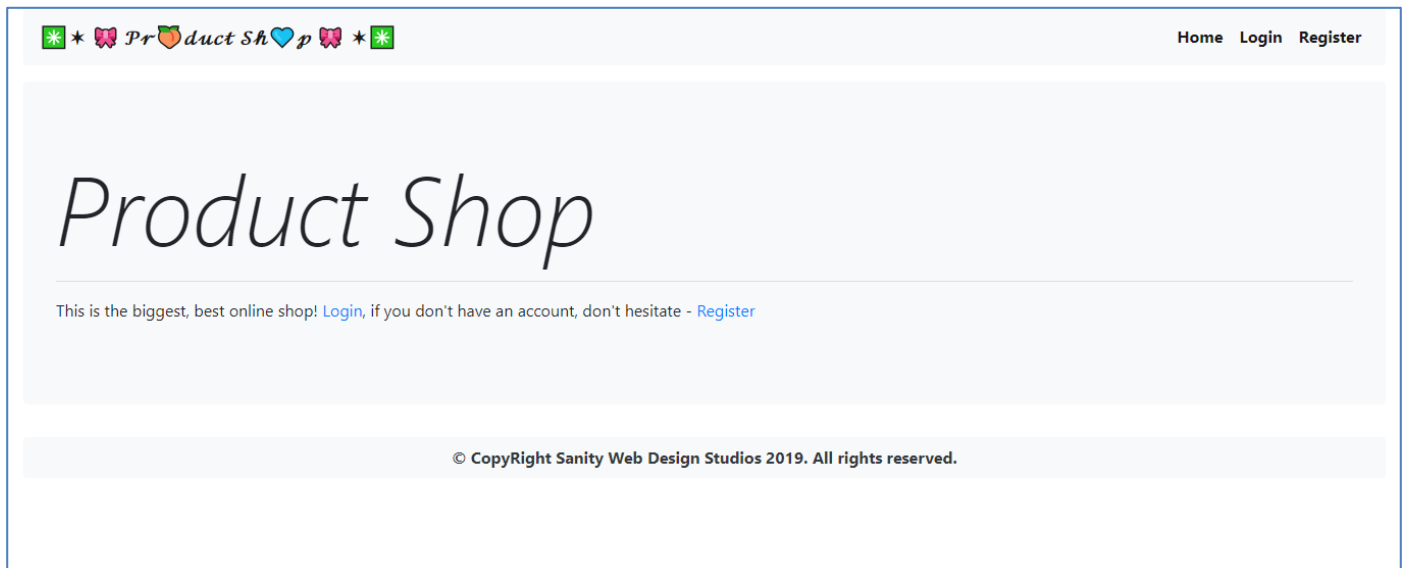


Project: Product Shop

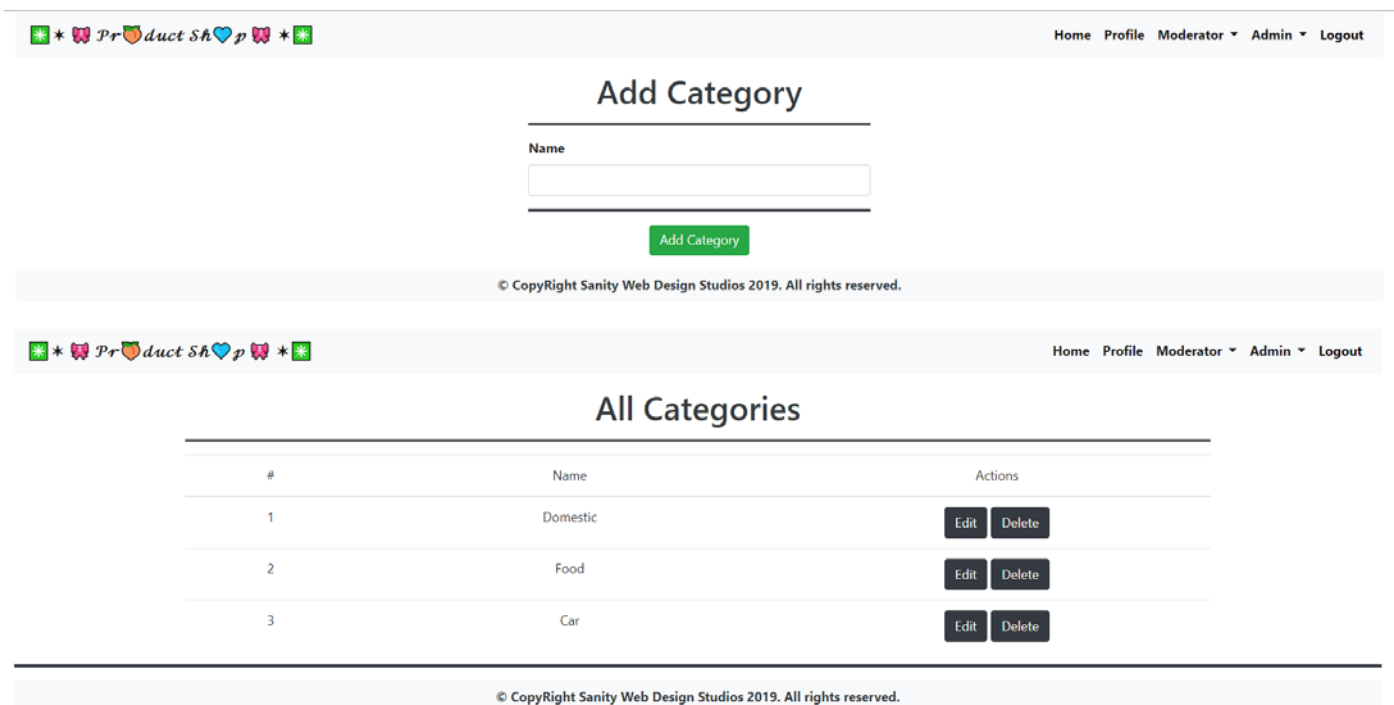
Product Shop is a system that registers users, categories, products and orders. It is a significantly big project, and as such it will have several parts. In this exercise you will land the basics of the application, in other words - the user functionality.



Workshop: Part 2 - Products

1. Views

These are the additional views you must implement.



Edit Category

Name

Domestic

Edit Category

© CopyRight Sanity Web Design Studios 2019. All rights reserved.

Delete Category

Name

Domestic

Delete Category

© CopyRight Sanity Web Design Studios 2019. All rights reserved.

Navigation

- ☐ All
- ☐ Domestic
- ☒ Food
- ☐ Car

Products



Name: Tomato
Price: 222.00



Name: Meme
Price: 123.46



Name: Cucumber
Price: 20.00



NOTE: The **Navigation** contains all of the added categories as radio buttons. Depending on your selection of a **radio** button, you should render different products. If you select the **[All]** button you should render all products.

Add Product

Name

Description

Price

Categories

--- Select Categories ---





- Domestic
- Food
- Car

Image

[Choose File](#) No file chosen

[Add Product](#)

All Products

#	Image	Name	Price	Actions
1		Tomato	222.00	Details Edit Delete
2		Meme	123.46	Details Edit Delete
3		Cucumber	20.00	Details Edit Delete
4		Onion	22222.00	Details Edit Delete

© CopyRight Sanity Web Design Studios 2019. All rights reserved.



Name: Tomato

Description: TomatoTomatoTomato

Price: 222.00

Edit Product

Name

Description

Price

Categories

Edit Product

Delete Product

Name

Description

Price

Categories

Delete Product

2. Data Entities

Create the required **entities**. Use the appropriate **data types**. In this exercise you will have Users and their Roles

- **Product**
 - **Id** – primary key, universally unique identifier

- **Name** – string
- **Description** – string
- **Price** – a floating-point number
- **ImageUrl** – a string
- **Categories** – a collection of **Category** objects.
- **Category**
 - **Id** – primary key, universally unique identifier
 - **Name** - string

3. Establish the Back-End

Create the required:

- **Entities**
- **Models**
- **Repositories**
- **Services**
- **Controllers**

4. Functionality

Categories

Implement all **CRUD** operations for the **Categories**. They should be pretty simple as the entity itself is not that complex.

Products

Implement all **CRUD** operations for the **Products**. NOTE: You cannot **edit** the **image** of the **Product**.