Exercises: Java EE: Servlet API

Problems for exercises and homework for the “Java Web Development Basics” course @ SoftUni. Submit your solutions on the **course page** of the **current instance**.

We will implement a simple application, only with Servlets, exploring the most interesting exploits of the Servlet API.

Chushka Application

Chishka is an application in which you create products, with several properties. It has many versions, and you will most probably see it several times.

You will have to create a simple multi-Servlet application which has several pages and 1 object entity.

1. Data

This is the data layer of the application. There is 1 data object for you to implement.

So let’s create our **Product**. The **Product** is a data object which stores data about an abstract **product**. You will see later what it will be used for.

First implement a class **Product** – which has these properties:

- **Name** – a **String**.
- **Description** – a **String**.
- **Type** – can be one of the following values – (“**Food**”, “**Domestic**”, “**Health**”, “**Cosmetic**”, “**Other**”)

Products			
Field	**1**	**2**	**3**
Name	Chushkopek	Injektoplqktor	Plumbus
Description	A universal tool for …	Dunno what this is…	A domestic tool for everything
Type	Domestic	Cosmetic	Food

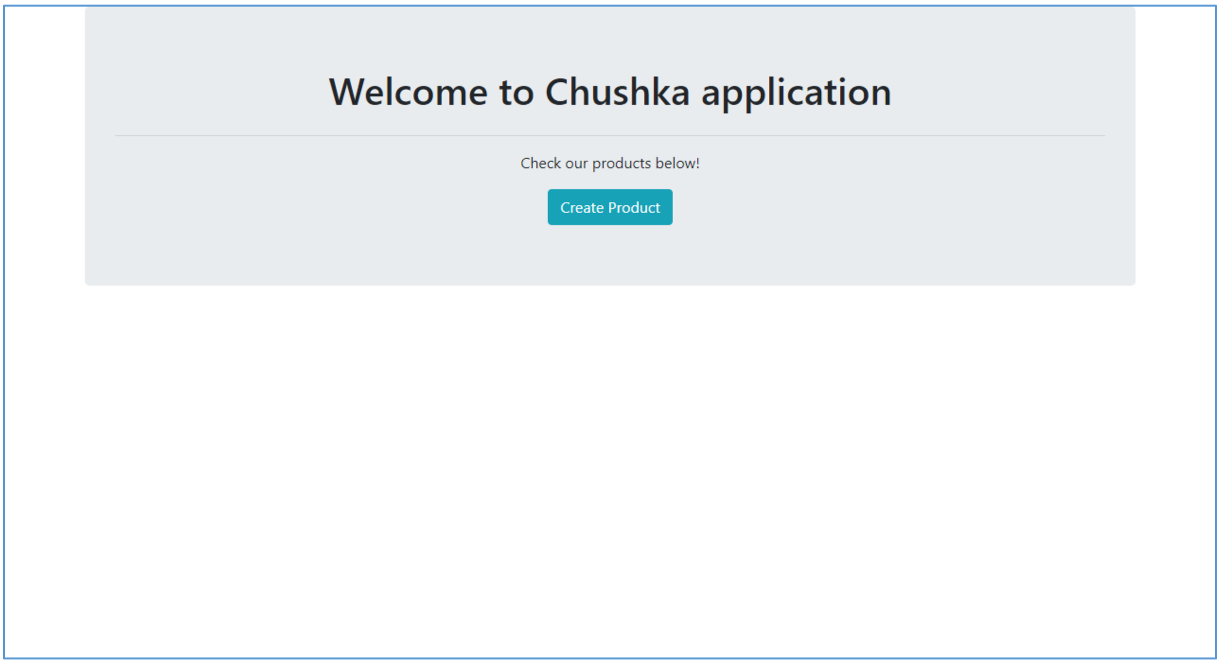
2. Views

In this application you must create several views, similar to the pictures below:

index.html

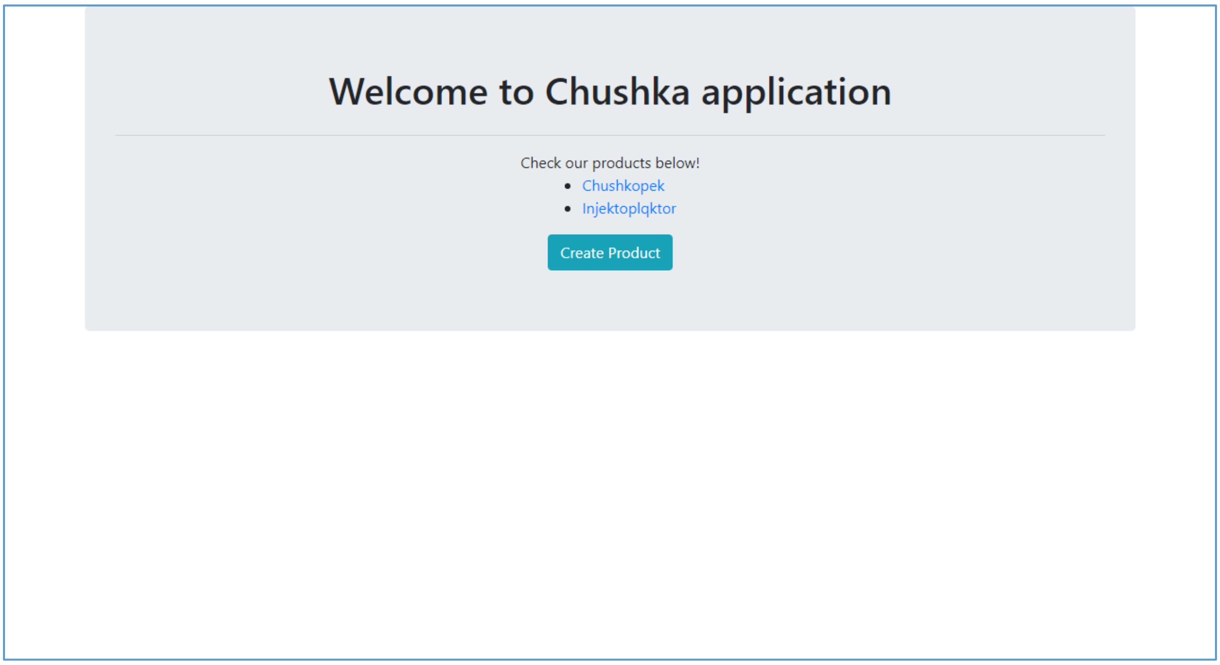
This is the home page, in other words the page that should be visualized when the application starts.

The index page must contain a button that redirect to **create-product.html** and **list all products names**. Upon clicking a **`name`** of a **`Product`**, you should be **redirected** to a **details page**, with **query parameter** – the **`name`** of the **`Product`**.



Welcome to Chushka application

Check our products below!



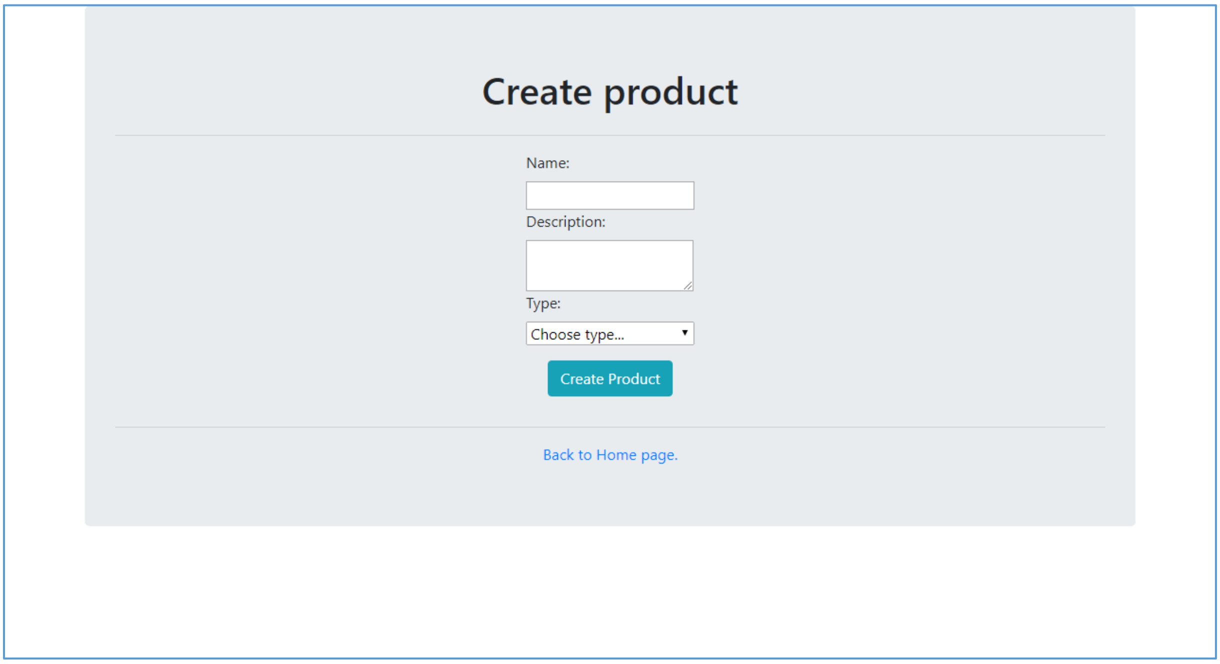
Welcome to Chushka application

Check our products below!

- Chushkopek
- Injektoplqktor

create-product.html

This is the page where you create your products. It must contain **input** field for **name**, **textarea** for **description**, field for **selecting** the product's **type**, **button** which creates a **Product** with the given values and a **link** that returns the **home** page.



Create product

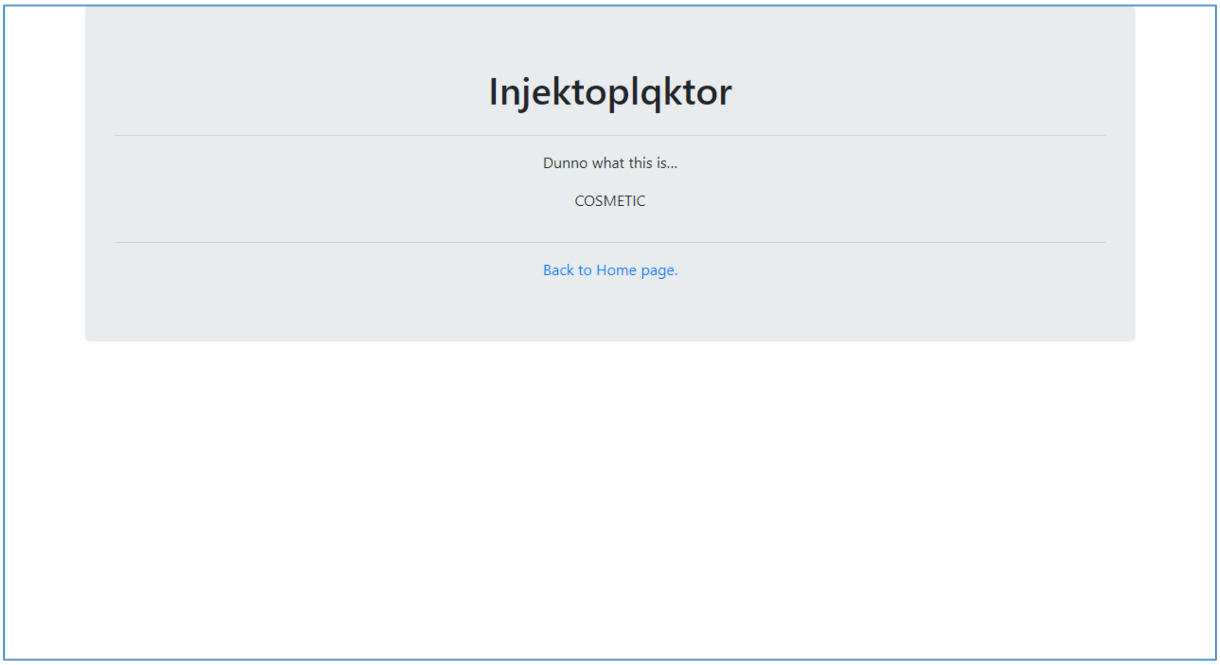
Name:

Description:

Type:

details-product.html

This is the page which **renders full data** about the **selected Product**. The selected **`Product`** should be extracted from the **`database`**, using the **name** from the **query parameters**.



Injektoplqktor

Dunno what this is...

COSMETIC

Back to Home page.

3. Servlets

Product Create

Implement a **Servlet** – **`ProductCreateServlet`**, which listens on route "**`/products/create`**".

Upon a **GET** request, it should return a form which accepts **3 inputs** – a **`name`**, a **`description`** and a **`type`**.
NOTE: You should only be able to submit "**`Food`**", "**`Domestic`**", "**`Health`**", "**`Cosmetic`**", "**`Other`**" as values for the **`type`**.

The **form** should send a **POST** request to the **`same route`**.

Upon a **POST** request, you should **redirect** to "**`/products/all`**".

Product All

Implement a **Servlet** – **`ProductAllServlet`**, which listens on route "**`/products/all`**".

Upon a **GET** request, the **Servlet** should render a page with the **names** of all created **`Products`**.
Upon clicking a **name** of a **`Product`**, you should be **redirected** to a "**`/products/details`**", with **query parameter** – the **name** of the **`Product`**.

Product Details

Implement a **Servlet** – **`ProductDetailsServlet`**, which listens on route "**`/products/details`**".

Upon a **GET** request, the **Servlet** should extract the **`Product`** with the given **name** in the **query parameters**, and render a page with **full information** about it.

4. Constraints

You must use the following things while implementing your application:

- Servlets
- Hibernate
- Repository layer
- Service layer
- Web layer