# Exercises: Introduction to MVC

Problems for exercises and homework for the [“Java Web Basics” course @ SoftUni](https://softuni.bg/courses/java-web-development-basics).

In the following exercises, you will be required to use the Java Servlets and the MVC framework you created. Make sure that you **import the MVC Framework** as a dependency and **create the right Tomcat configuration**!

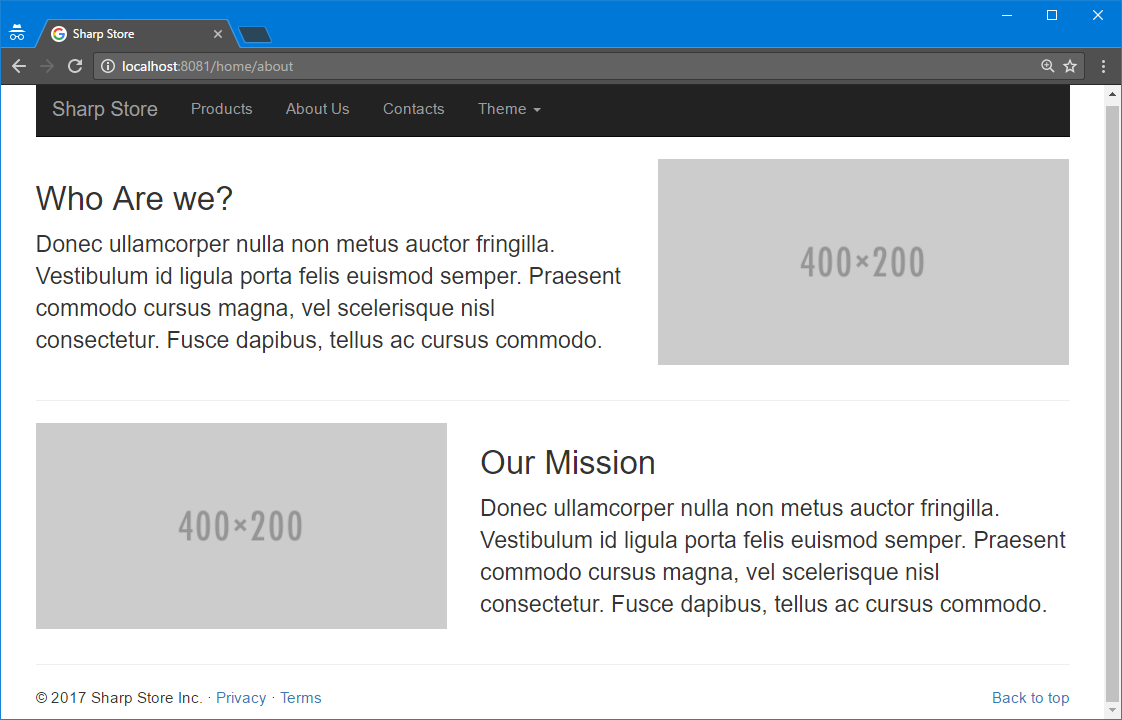
## Sharp Store – Home Page

In the following task, you will make a simple company website for company that sells knives. Check out the provided skeleton, look at the structure of the project and the files in the content folder. You are given home.jsp and about.jsp pages that you will need to use.



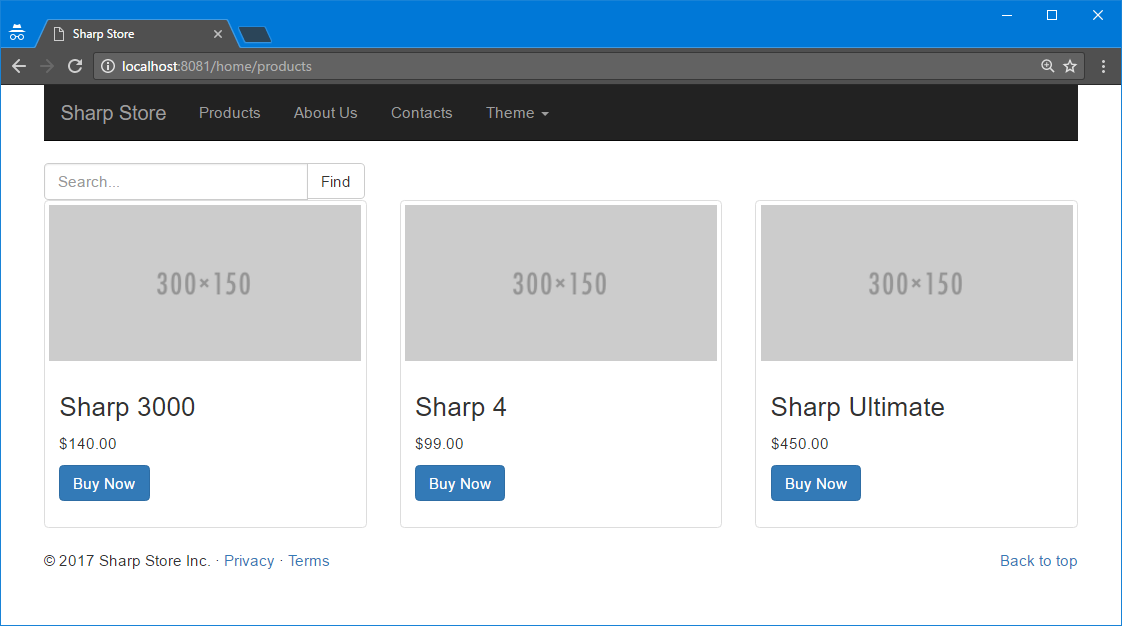
## Sharp Store – About us

Extend the website using the provided about.jsp in the resources create a page that would look like the one below. Make the links from the menu of the home page lead to that page and make the home page accessible from about us page.



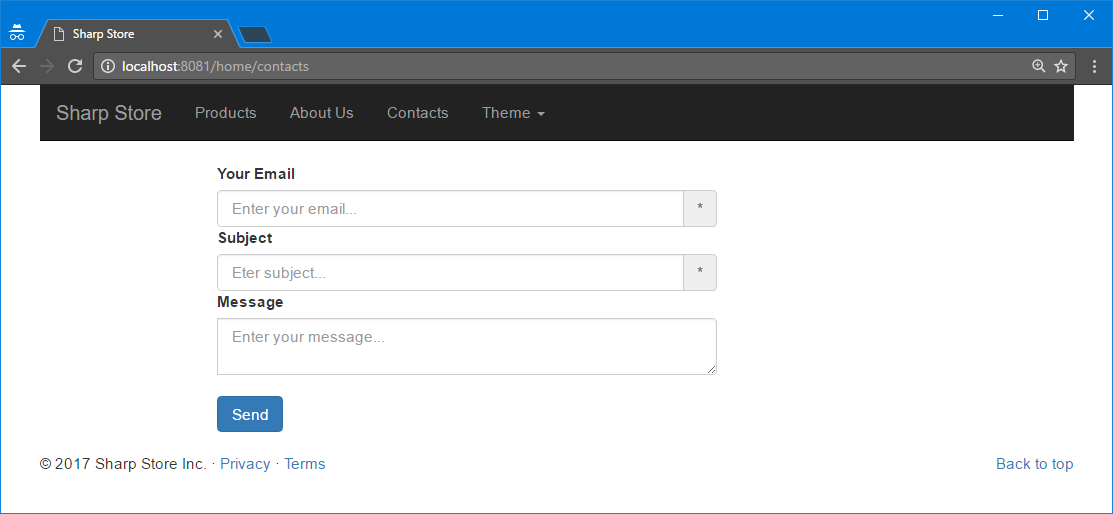
## Sharp Store – Products

Time to add new page all by yourself. You can use the initial about.jsp page as template and make page similar to that. The final result should be like the picture below and again the navigational links should work.



## Sharp Store – Contacts

Similar to the previous exercise create a contacts.jsp page like the one in the picture below.



## Sharp Store – Products from Database

Create database **SharpStore**. It should have few **singleton** **repositories**, that will keep the table objects in-memory. First, create **Knives**. Each knife has **name**, **price** and **image URL**. Populate the table with some sample records. Then **modify your application** so when /home/products page is loaded it should load thumbnails for **all knives listed in the database** instead of just the dummy placeholders you created in Problem 3.

## Sharp Store – Send Message

In the database, you created in the previous problem add new **singleton repository** **Messages**. Each message has **sender**, **subject** and **message.** Modify your application so when the form from the /home/contacts page sends a POST request the **message would be saved to the database.**

## Sharp Store – Search

Add functionality to the /home/products page so the visitors of the site can search a knife by its name (or part of it). When the **Find** button is clicked show only the knives that contains the word provided by the search form.

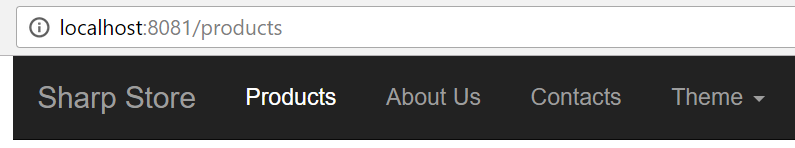
## Sharp Store – Buy Knife

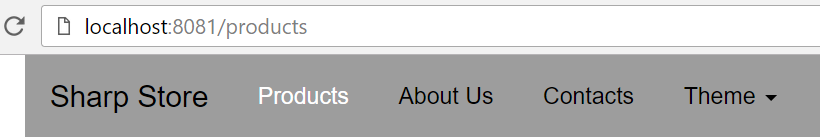
When user selects some knife, and wants to buy it when clicking on the “**Buy Now**” button he should be redirected to /home/buy page where he should fill a form and submit it. The form fields are – **name**, **phone number**, **address**, **type of delivery** (express or economic). When the form is submitted the data from the form must be **saved to the database.**

## Sharp Store – Change Site Theme

Add another functionality to the user. When Dark/Light option from the navigational bar is selected **change the color of the navigational bar** (black/light gray). Make sure it is changed when trying to visit **all pages**.

### Example





## Sharp Store – Admin Panel

The employees from the **SharpStore** should have access to the database and perform operations with it such as:

* **Add, edit or delete products**
* \*See a list of **all messages** with option to **reply** to each one by sending an email
* See a list of **all orders** with their details and has option to **change their status**. When the client places a new order it’s **default** status is **Pending,** the other options for order status are **Shipped** and **Delivered**.

The employees should **not be** **able to register from the website** they must be **added manually to the database**. Once added to the database they **can log in with the provided credentials** (username and password). Make sure the admin panel is only accessible when the user has logged in. Otherwise redirect him to the login page.