# Java Web Development Basics

# Exam Preparation – Sboj.gb

**Sboj.gb** is an application for job offering. You have been tasked to implement this application for an unusually low price, by an unusually rich client. There are several requirements you must follow in the implementation.

## Technological Requirements

* Use the Javache Web Server
* Use the Broccolina and Toyote request handlers
* Use the Summer Framework
* Use Hibernate native (no Spring Data)

The Technological Requirements are **ABSOLUTE**. If you **do not follow** them, you will **NOT** be scored for other Requirements.

Now that you know the Technological Requirements, let us see what are the Functional Requirements.

## Database Requirements

The **Database** of the **Sboj.gb** application needs to support **2 entities**:

### User

* Has an Username
* Has a Password
* Has an Email

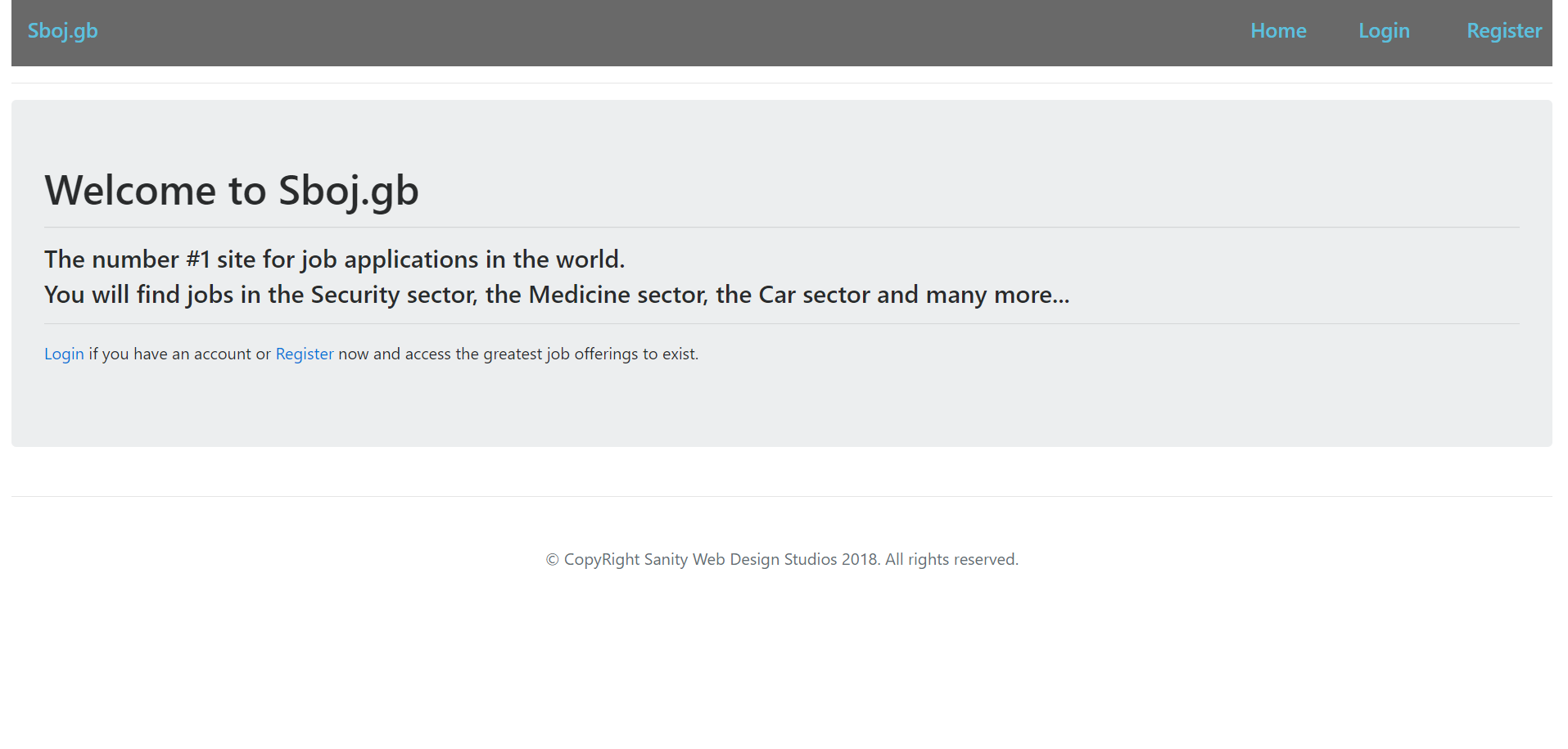
### Job Application

* Has a Sector
* Has a Profession
* Has a Salary
* Has a Description

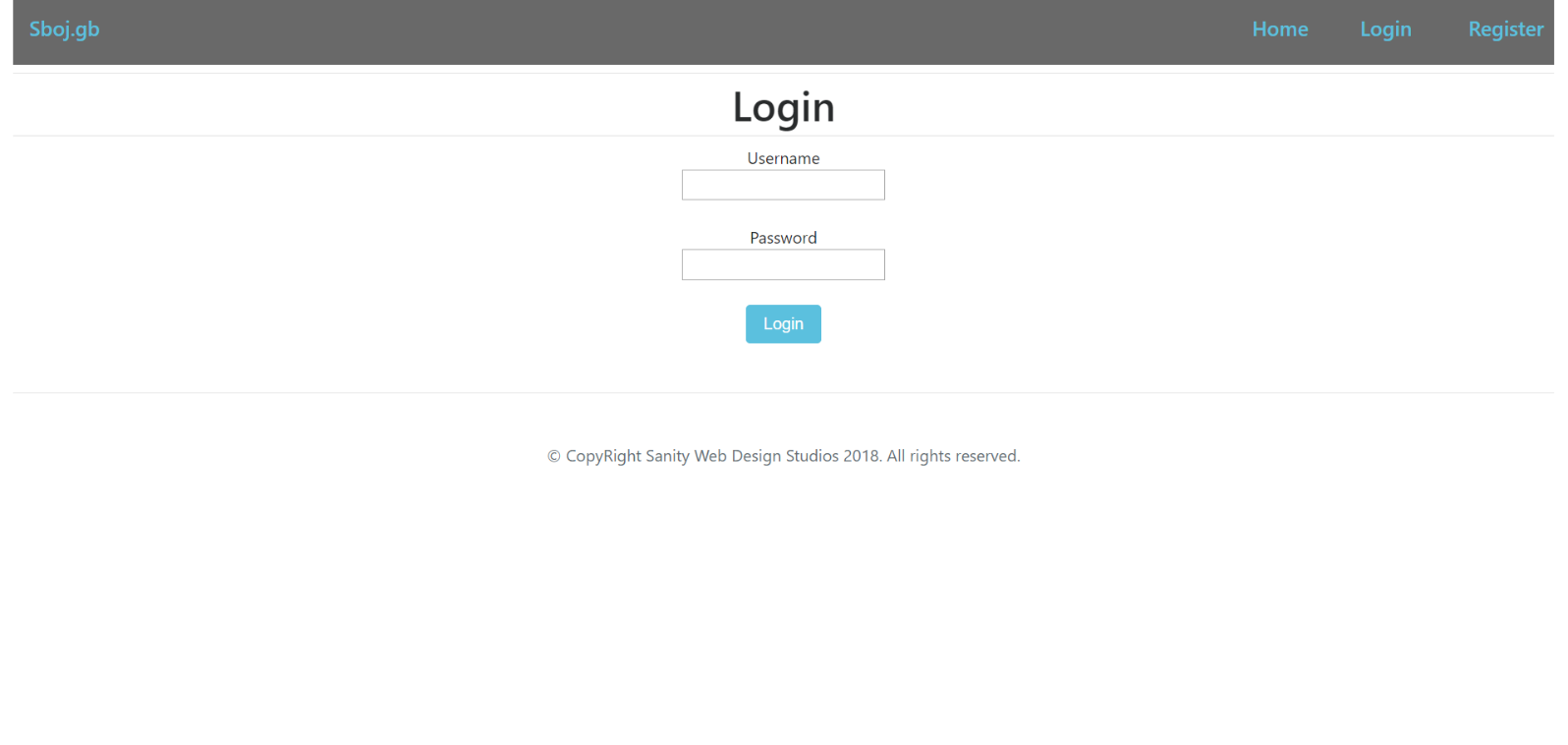
Implement the entities with the **correct datatypes**, and implement **repositories** for them.

## Template Requirements

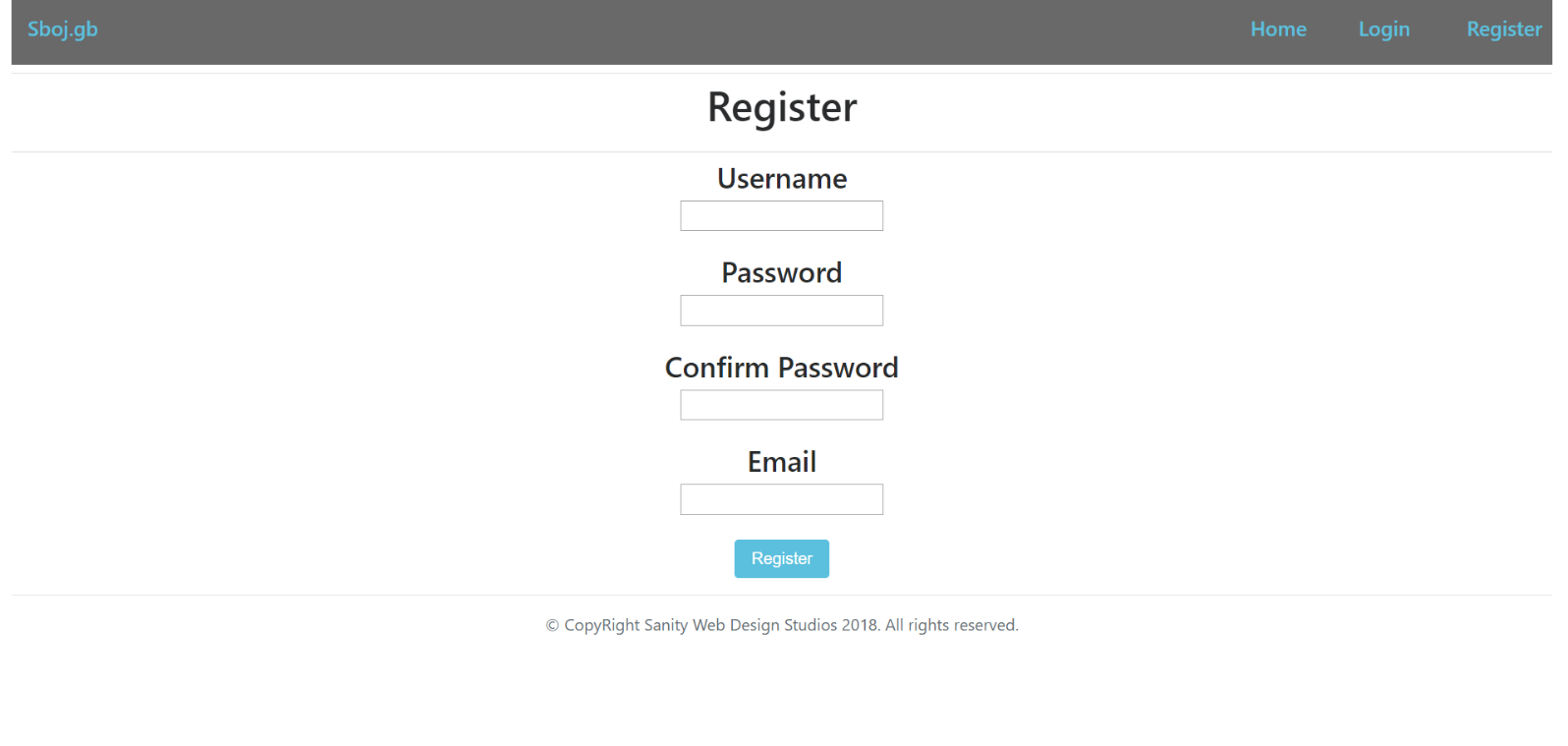
### Index Template (route = “/”) (logged out user)



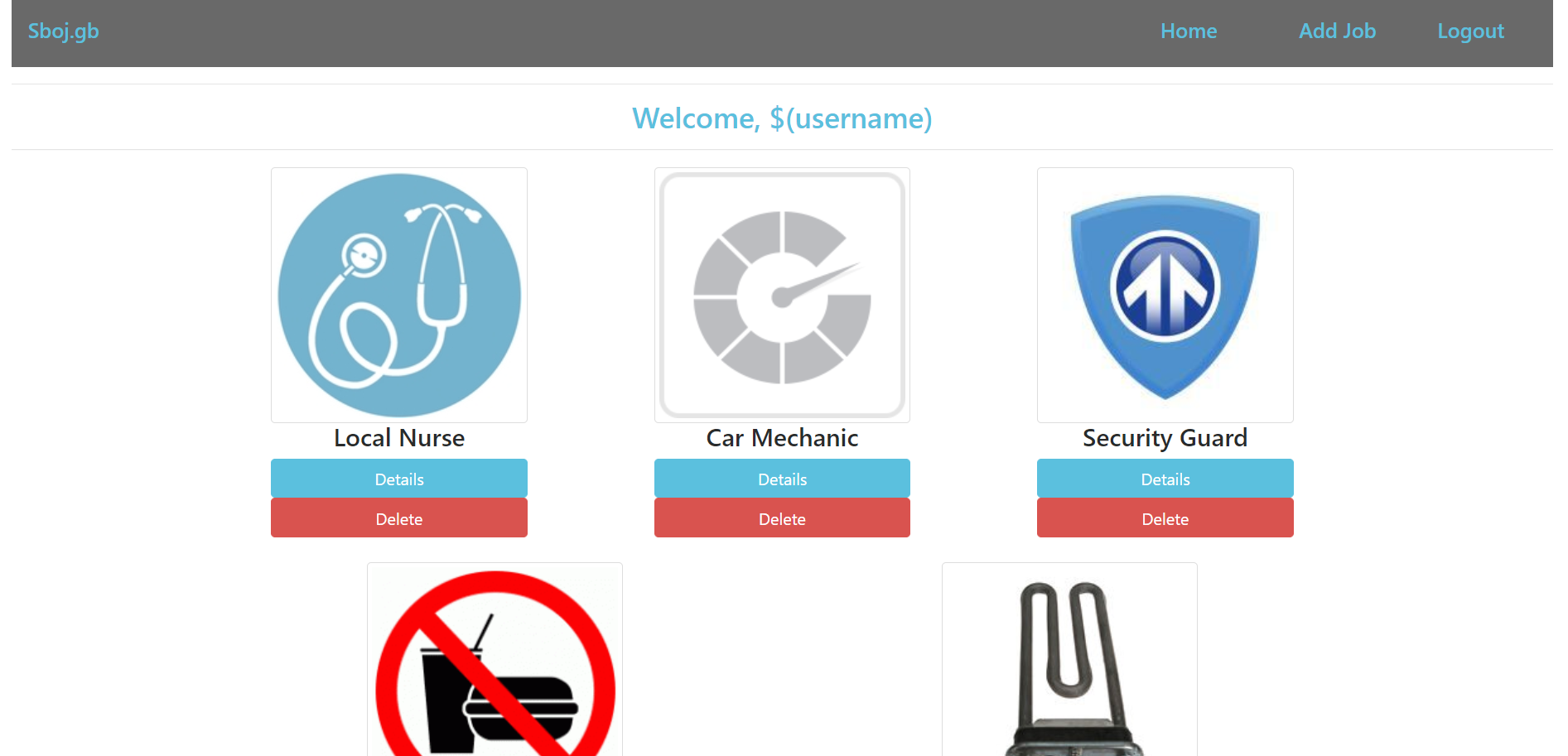
### Login Template (route = “/login”) (logged out user)



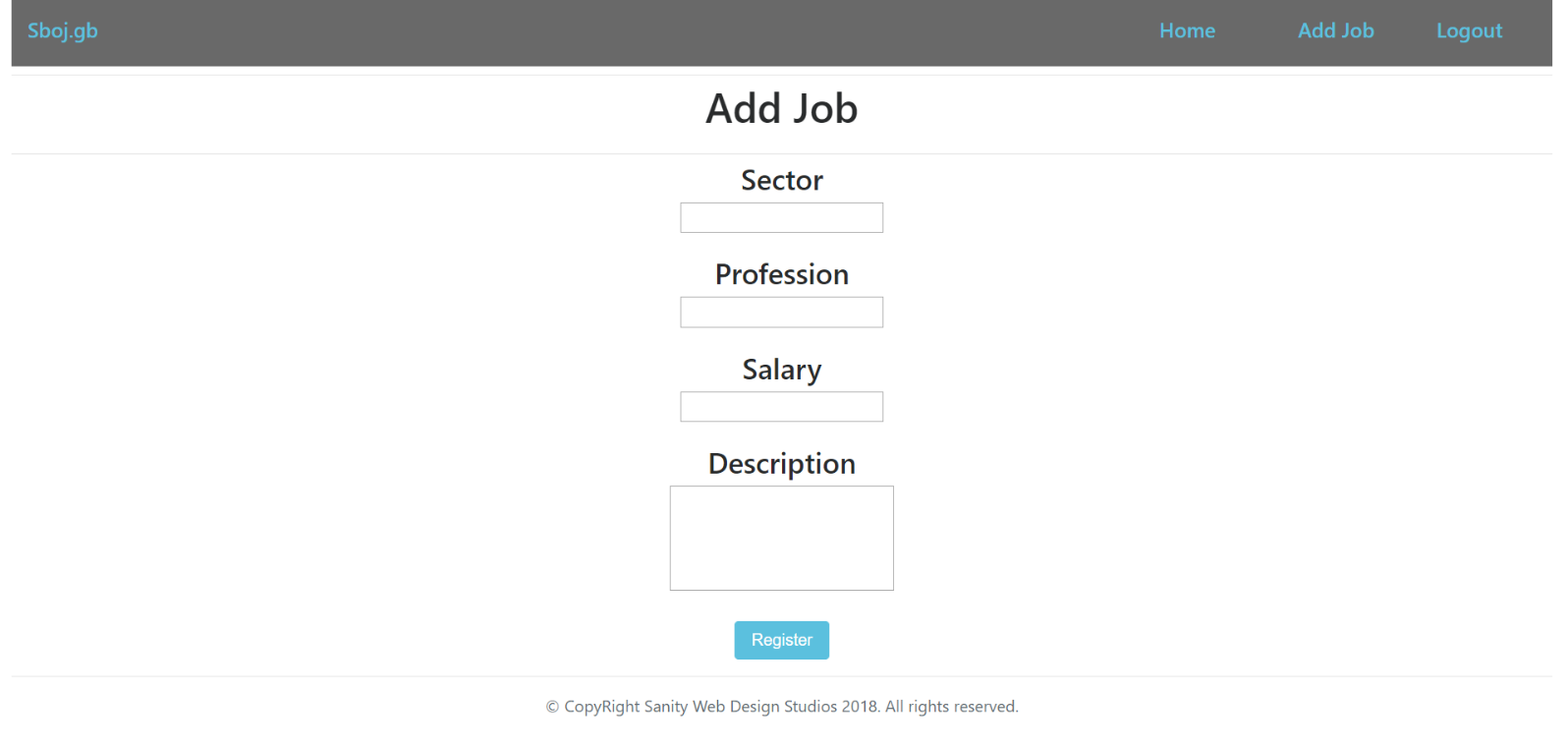
### Register Template (route = “/register”) (logged out user)



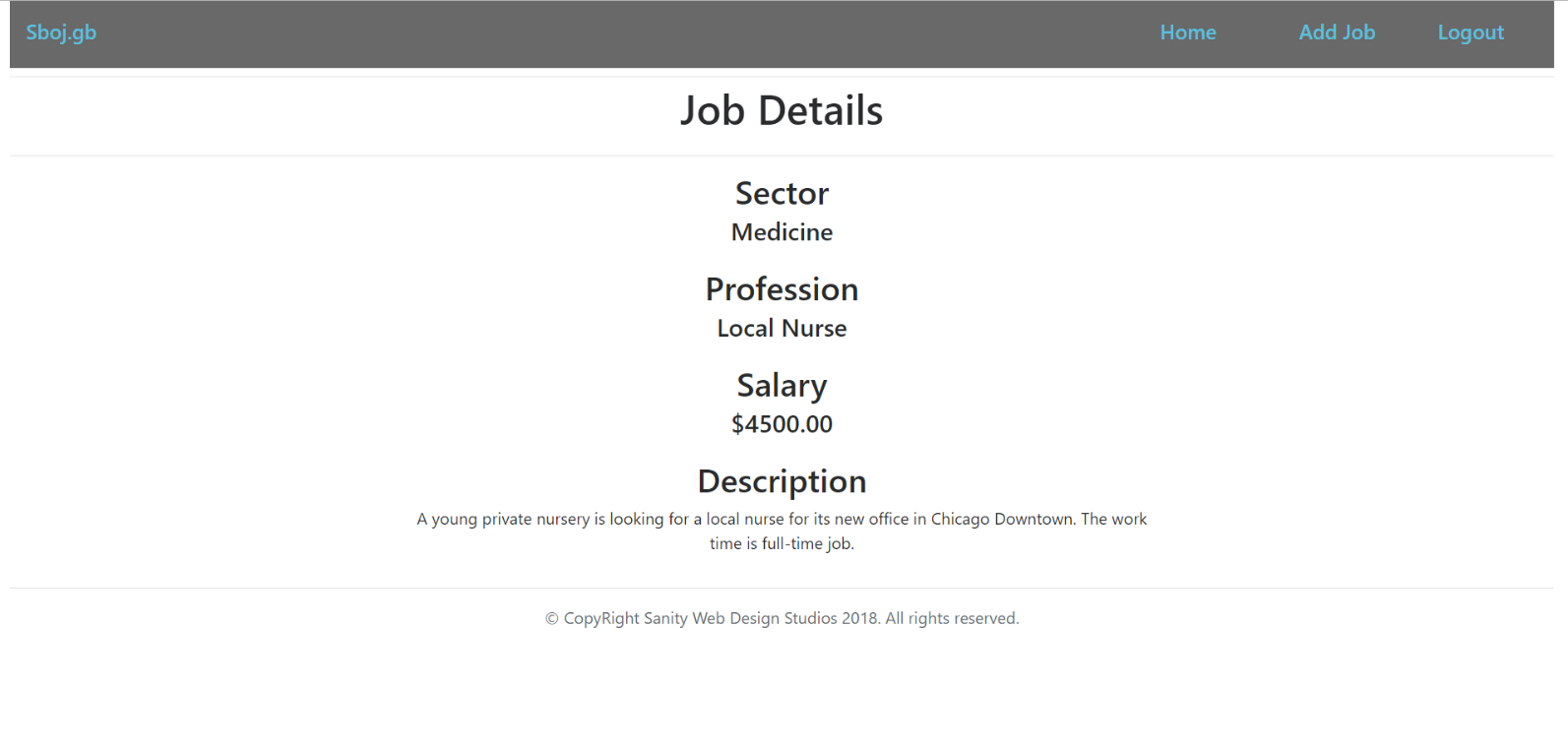
### Home Template (route=”/home”) (logged in user)



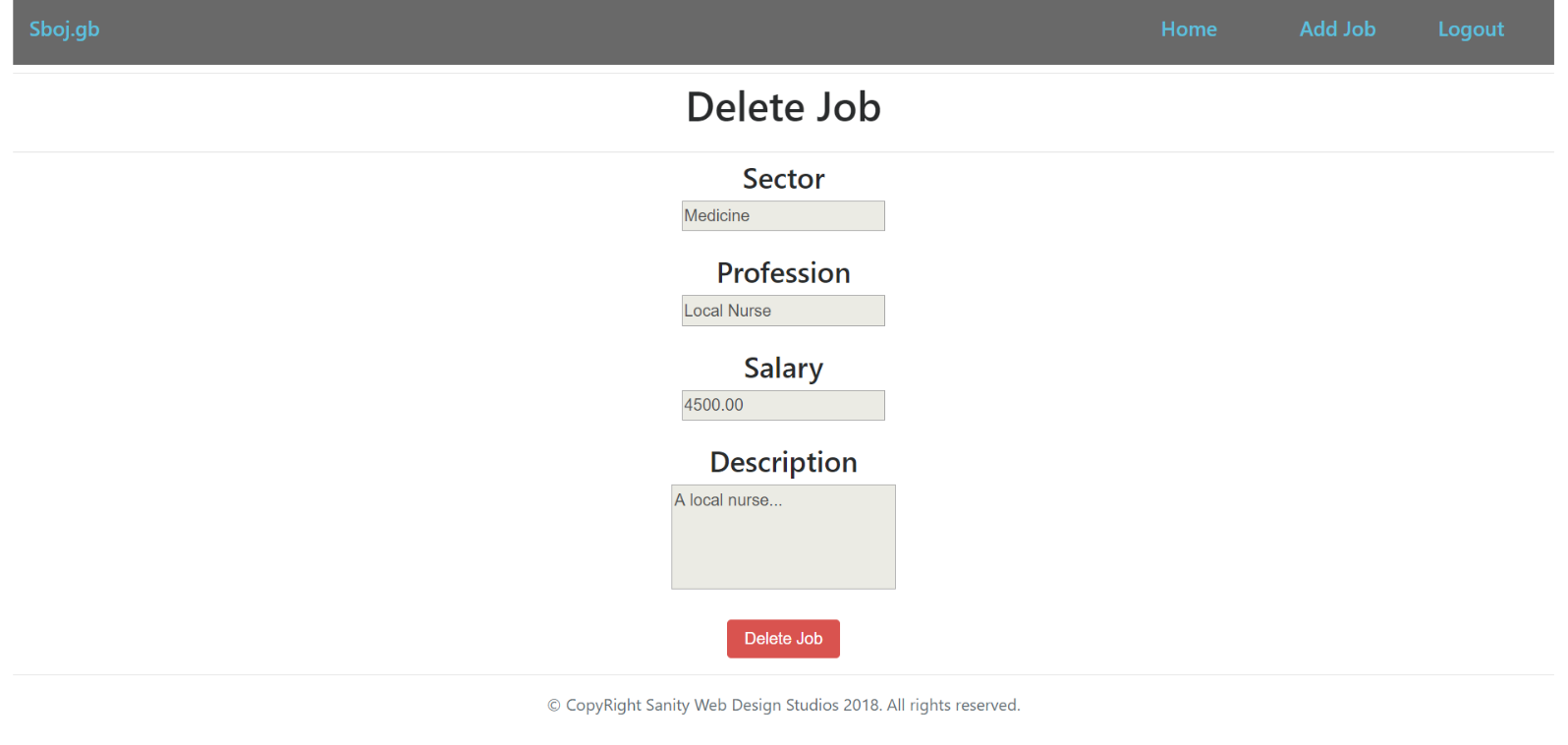
### Add Job Template (route=”/jobs/add”) (logged in user)



### Job Details Template (route=”/jobs/details/{id}”)



### Job Delete Template (route=”/jobs/delete/{id}”)



Some of the templates have been given to you in the application skeleton, but the others will be for you to implement, so make sure you implement them correctly. You can use the given ones as helpers.

**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 job, view details about a job, delete a job, and view all jobs (Home page).

The **application** should provide **functionality** for **adding jobs** with **only** **5 sectors** – “Medicine”, “Car”, “Food”, “Domestic”, “Security”.

The **application** should **store** its **data** into a MySQL database, using Hibernate native.

## 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 Job Details page.
* Users (logged in) can access Job Delete page.
* Users (logged in) can access Add Job page.
* Users (logged in) can access Logout functionality.

## Framework Requirements

Extend the given skeleton (Javache Web Server, Broccolina / Toyote Request Handlers, Summer Framework) by **adding** **notifications**.

For example, when the **user logs** in with a **wrong password**, you should **notify him** of that, with a **simple element** on the **front-page**.

## Scoring

This section describes how the scoring of the Exam Preparation will be made.

### Database Requirements – 20 points.

### Template Requirements – 15 points.

### Functional Requirements – 35 points.

### Security Requirements – 10 points.

### Framework Requirements – 20 points.