# Gira

Submit your solutions on the [here](https://judge.softuni.bg/Contests/Practice/Index/3122#1).

## Database Requirements

The **Database** of the **Gira** 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 (unique)
  + Must contains '@'.
  + Cannot be null.

### Task

* 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 Progress (unique)
  + Option between (OPEN, IN\_PROGRESS, COMPLETED, OTHER)
* Has a Due Date
  + A date, that cannot be in the past
* Has a Classification
  + Classification cannot be null.
* Has a User
  + Cannot be null.

### Classification

* Has an **Id** – **UUID-string or Long**
* Has a **ClassificaionName** (unique)
  + option between (BUG, FEATURE, SUPPORT, OTHER)
* Has a Description

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

## Initialize classifications

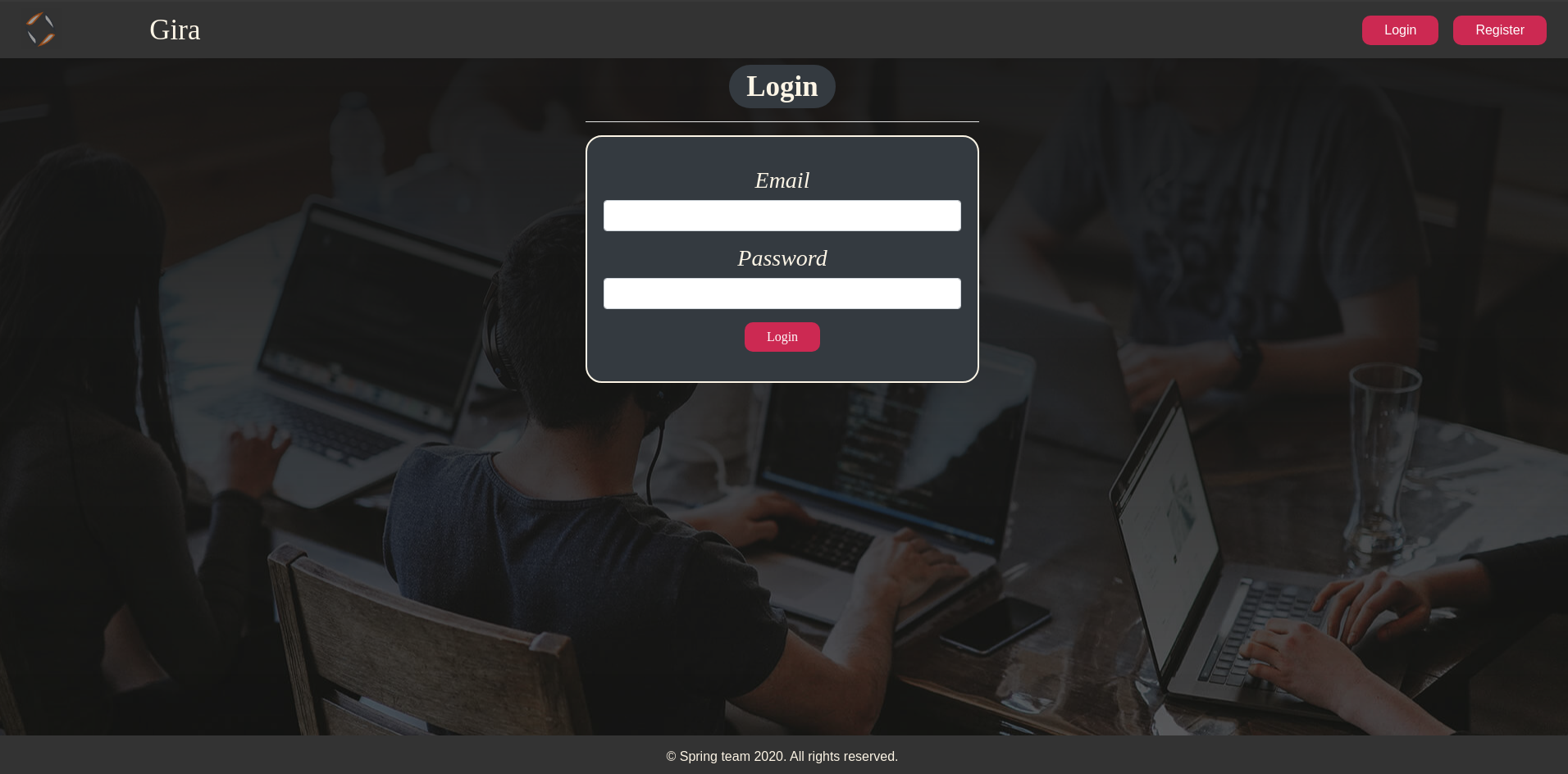
* Implement a method that checks (when the app is started) if the database does not have classifications and initialize them
  + You are free to do it in different ways.

## Page Requirements

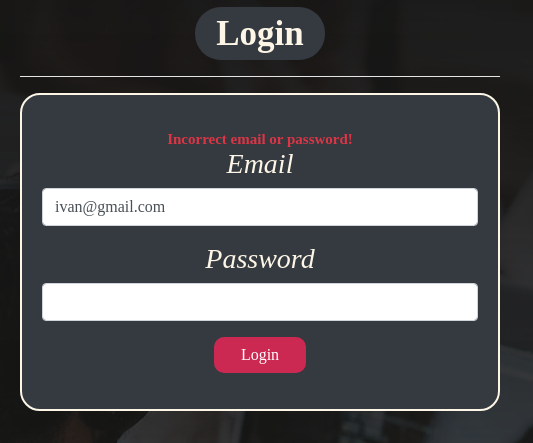
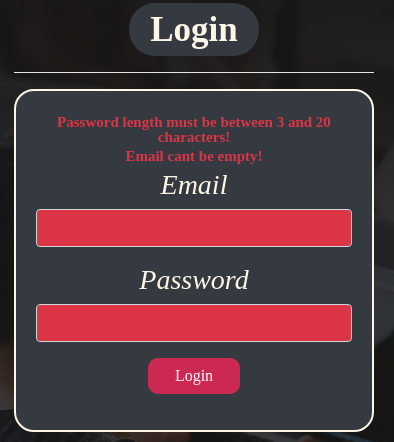
### Index Page (logged out user)



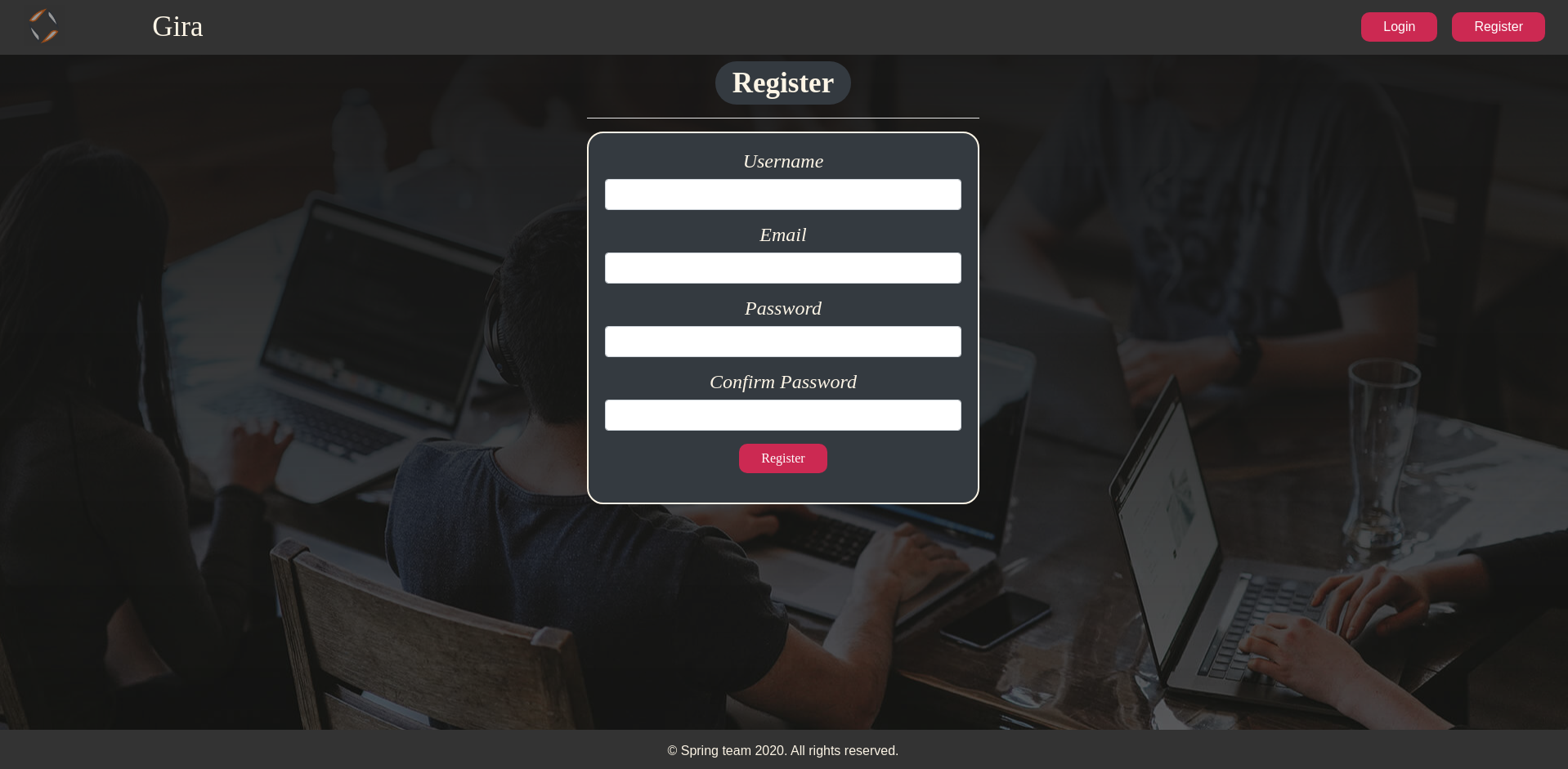
### Login Page (logged out user)



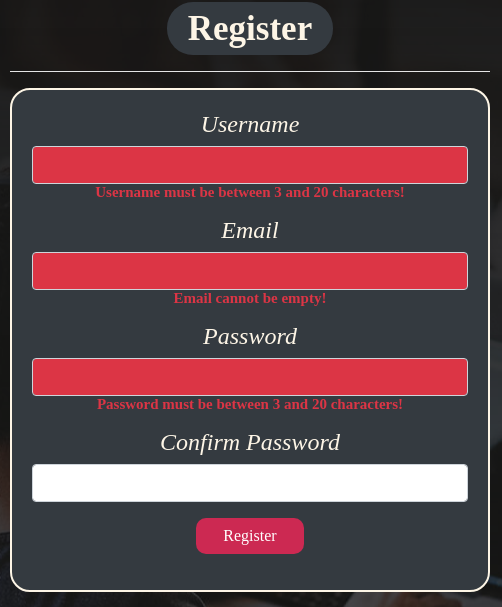
### Login Page validations



### Register Page (logged out user)

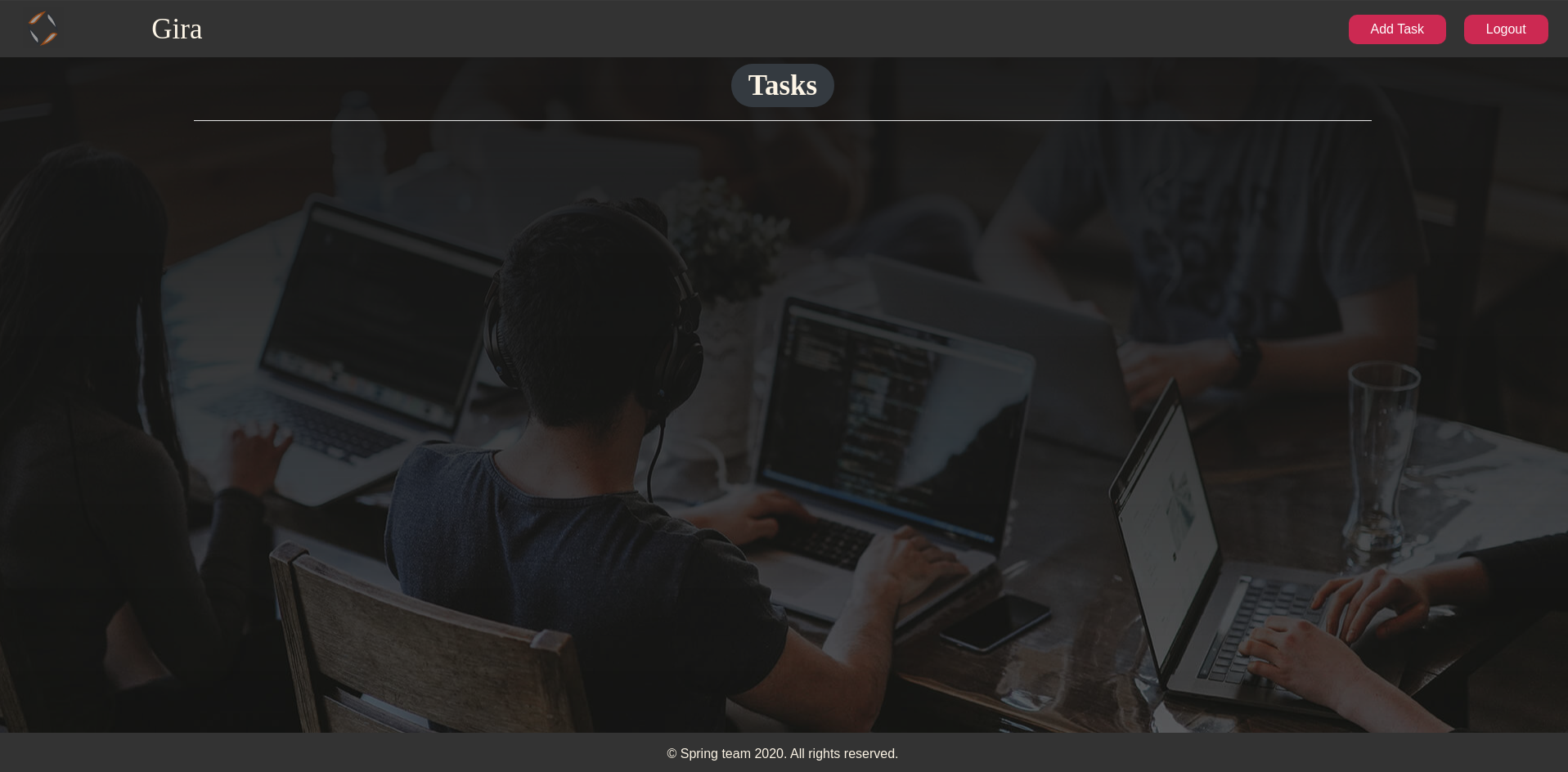


### Register Page validations

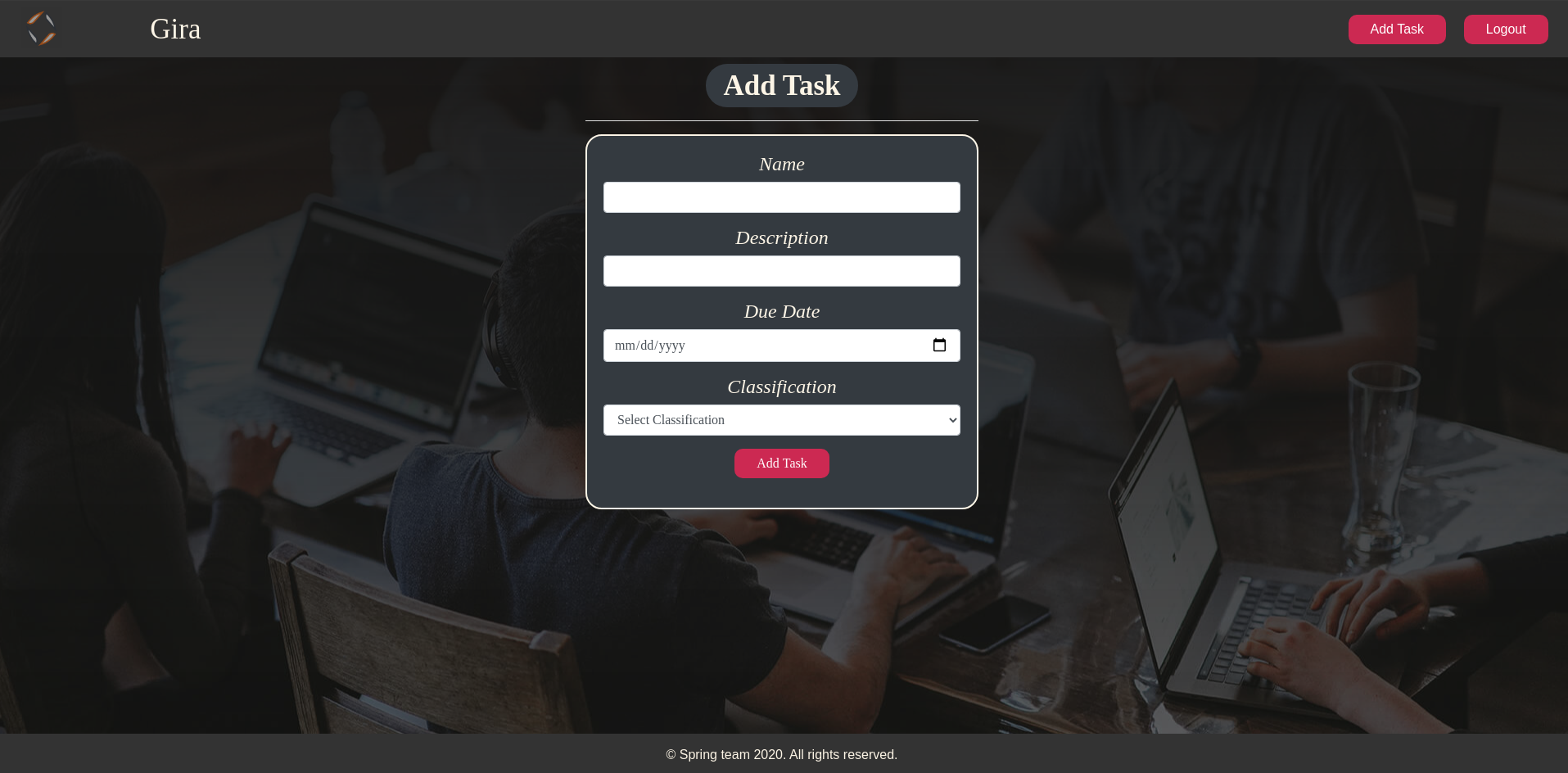


### Home Page (without having any tasks)

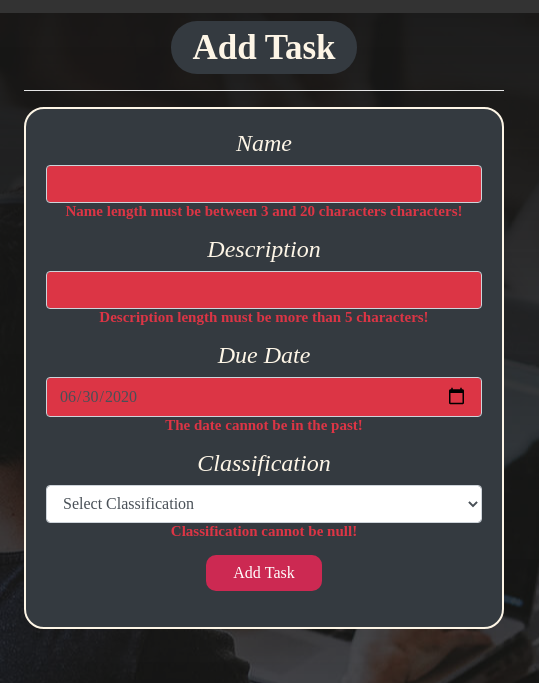
* Note: The home page should visualize **all of the tasks** from the database



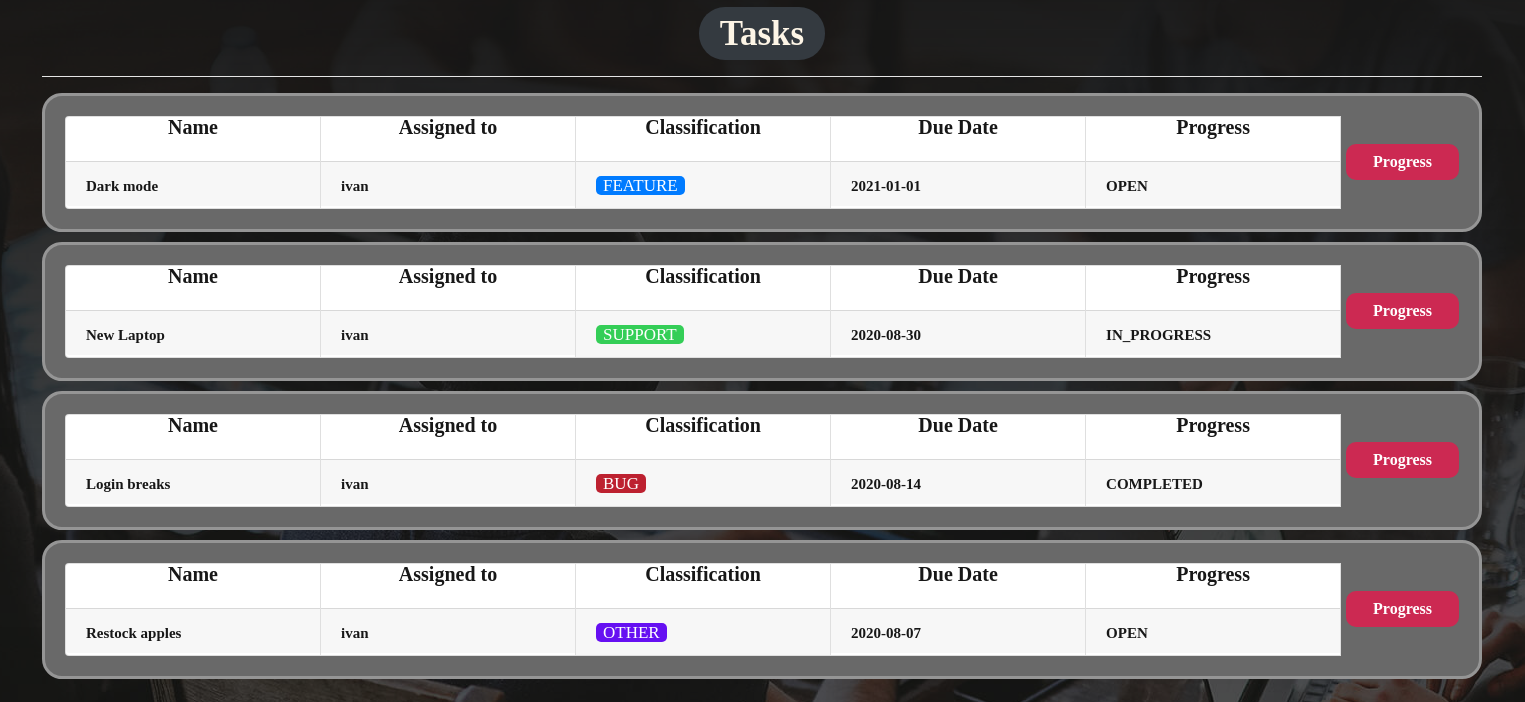
### Add task



### Add task validation



### Home Page (with tasks)



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. The **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 Task**, **view all** Tasks (Home page). When creating a task, the **"Assigned to"** field should have **the person's name who created the task** and the initial progress should be **"OPEN".**

**The progress button** changes the progress column of the task:

* If the progress is **"OPEN"**, the progress becomes **"IN\_PROGRESS"**.
* If the progress is **"IN\_PROGRESS"**, the progress becomes **"COMPLETED"**.
* If the progress is **"COMPLETED"**, **delete** the task **from the database.**

**Gira** in navbar should redirect to appropriate **URL depending** on that if the user is logged in.

The **application** should provide **functionality** for **adding tasks** with **classification (**BUG, FEATURE, SUPPORT, OTHER).

The task's classification should have different colours. (The colours and the classes are in the end of the style.css file)

The **appropriate task's classification** should have the **appropriate style class.** The names are the samefor **the css classes** and **the classification names.** For example:



The **application** should **store** its **data** into 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 Task page.
* Users (logged in) can access Logout functionality.