Planner Application

Regular Exam [Spring Fundamentals]

PlannerApp is designed to help individuals and teams manage their workflow efficiently. It provides a user-friendly interface with a customizable Kanban board that allows users to create and move cards representing their tasks or work items across different stages of their workflow. The App offers features such as the ability to set due dates, add description or comments, and assign tasks. It incorporates different priorities of the tasks, such as "Urgent" "Important" and "Low". The application provides real-time collaboration, enabling team members to work together more efficiently and effectively. the newest online marketplace dedicated to buying and selling new or used items. At any given time, the site has a wide variety of listings ranging from hand tools, clothes and books to accessories, pro audio gear, bikes, smartphones, and more...

There are several requirements you must follow in the implementation:

1. Database Requirements

The **Database** of the **Reseller** application needs to support **3 entities**:

User

- **Id** – Accepts **UUID-String** or **Long** values
- **Username**
  - The **length** of the **values** should be **between 3** and **20** characters long (both numbers are **INCLUSIVE**)
  - The values should be **unique** in the database
- **Password**
  - The **length** of the **values** should be **between 3** and **20** characters long (**INCLUSIVE**)
- **Email**
  - The values should contain a **'@'** symbol)
  - The values should be **unique** in the database
- **Assigned Tasks**
  - The assignedTasks is a collection that contains users tasks that are assigned to him. One user may **have many** tasks and one task can be assigned to **only one** user.

Task

- **Id** – Accepts **UUID-String** or **Long** values
- **Description**
  - The **length** of the **values** should be **between 2** and **50** characters long (both numbers are **INCLUSIVE**)
- **Due Date** – date
  - Future date
- **Priority**
  - One task **has one** priority and one priority can have **many tasks**
- **User**
  - The user who has assigned the task to him so he can work on the task. One task **has one** user, but one user may **have many** tasks.

Priority

- **Id** – Accepts **UUID-String** or **Long** values
- **Priority name**
  - an option between (**URGENT**, **IMPORTANT** and **LOW**)

- **Description** (not null)
    - For URGENT - "An urgent problem that blocks the system use until the issue is resolved."
    - For IMPORTANT - "A core functionality that your product is explicitly supposed to perform is compromised."
    - For LOW - "Should be fixed if time permits but can be postponed."

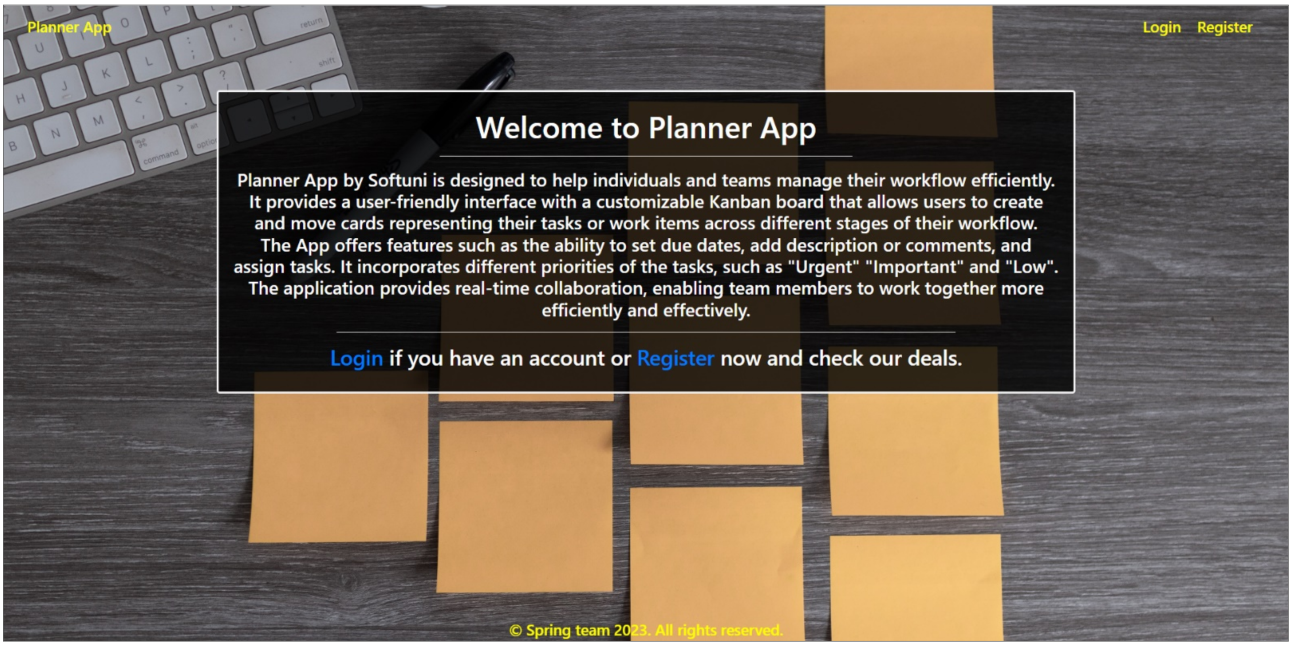
Nullable/Empty/Blank values are not allowed unless explicitly mentioned. Implement the entities with the **correct data types** and implement the **repositories** for them.

2. Initialize categories

- Implement a method that checks (when app started) if the database does not have any category and initialize them
    - You are free to do this in some different ways
    - You can skip the description if you want

3. Page Requirements

Index Page (logged out user)



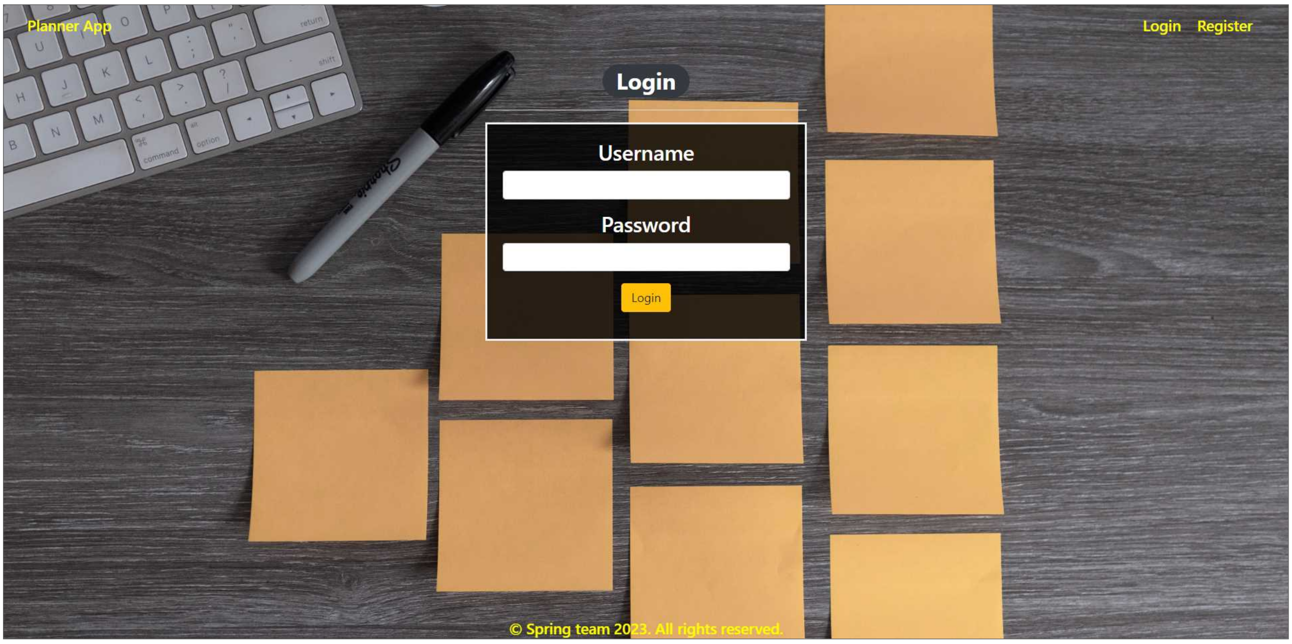
Welcome to Planner App

Planner App by Softuni is designed to help individuals and teams manage their workflow efficiently. It provides a user-friendly interface with a customizable Kanban board that allows users to create and move cards representing their tasks or work items across different stages of their workflow. The App offers features such as the ability to set due dates, add description or comments, and assign tasks. It incorporates different priorities of the tasks, such as "Urgent" "Important" and "Low". The application provides real-time collaboration, enabling team members to work together more efficiently and effectively.

Login if you have an account or Register now and check our deals.

© Spring team 2023. All rights reserved.

Login Page (logged out user)

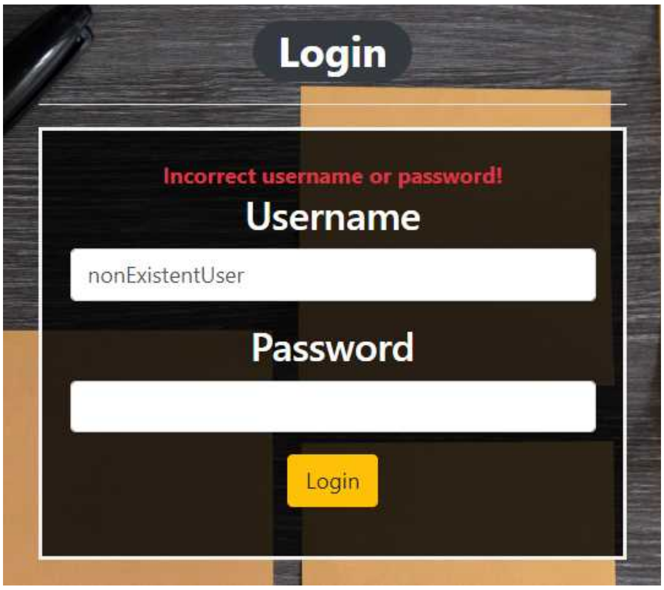


Username

Password

© Spring team 2023. All rights reserved.

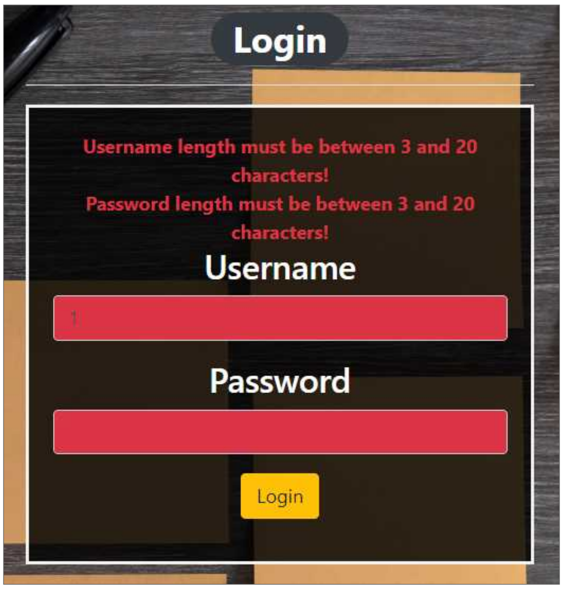
Login Page validations



Incorrect username or password!

Username: nonExistentUser

Password:



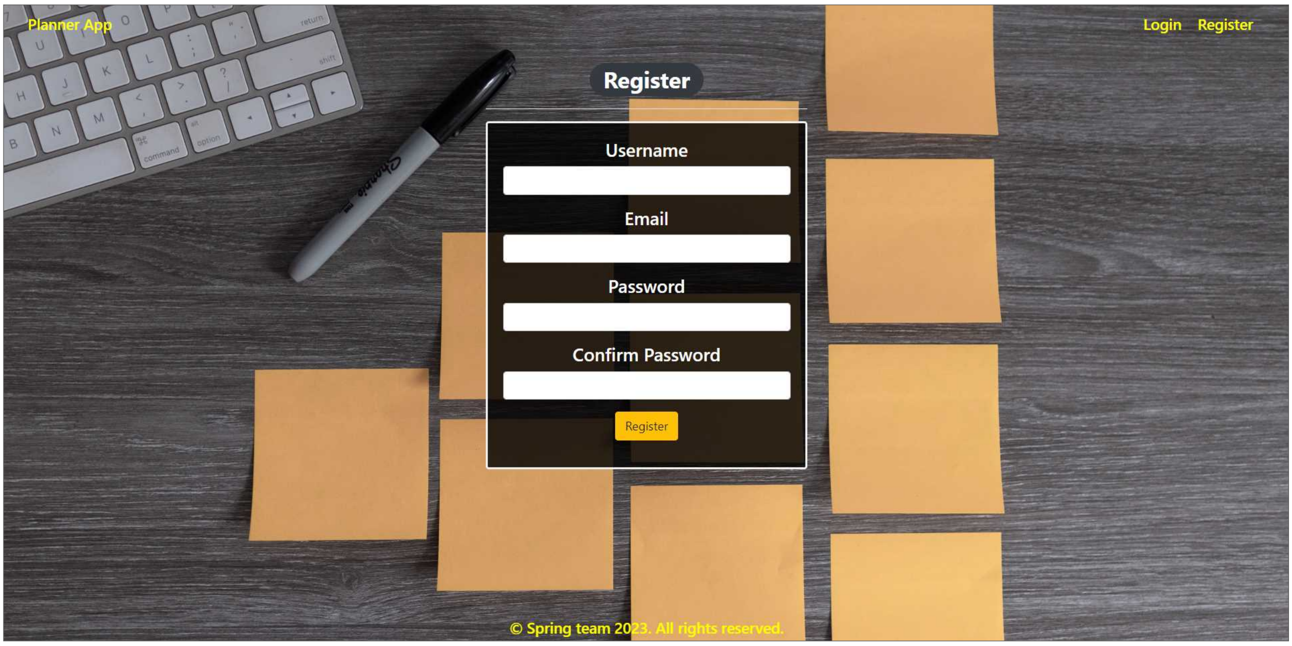
Username length must be between 3 and 20 characters!

Password length must be between 3 and 20 characters!

Username: 1

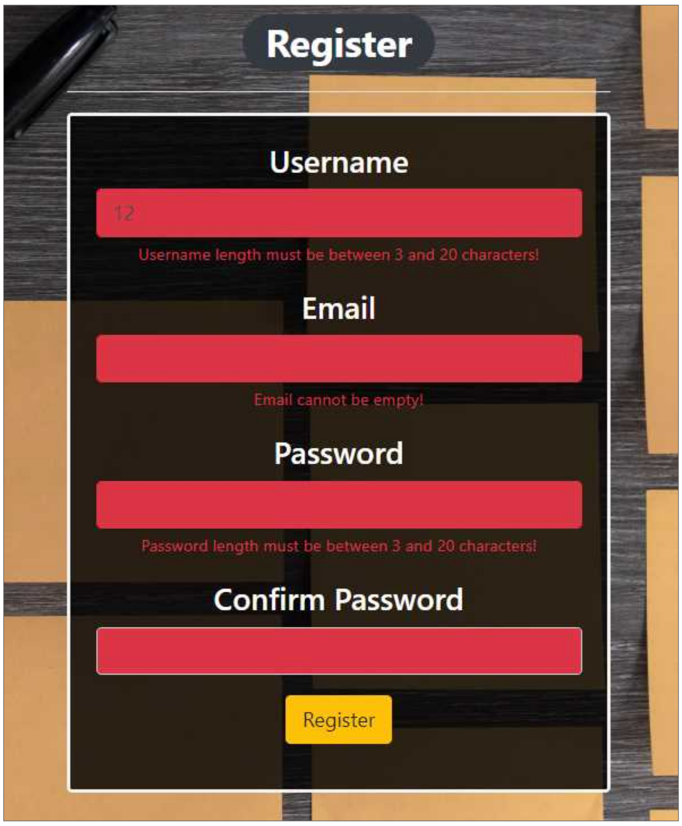
Password:

Registration Page (logged out user)



© Spring team 2023. All rights reserved.

Registration Page validations



Register

Username

12

Username length must be between 3 and 20 characters!

Email

Email cannot be empty!

Password

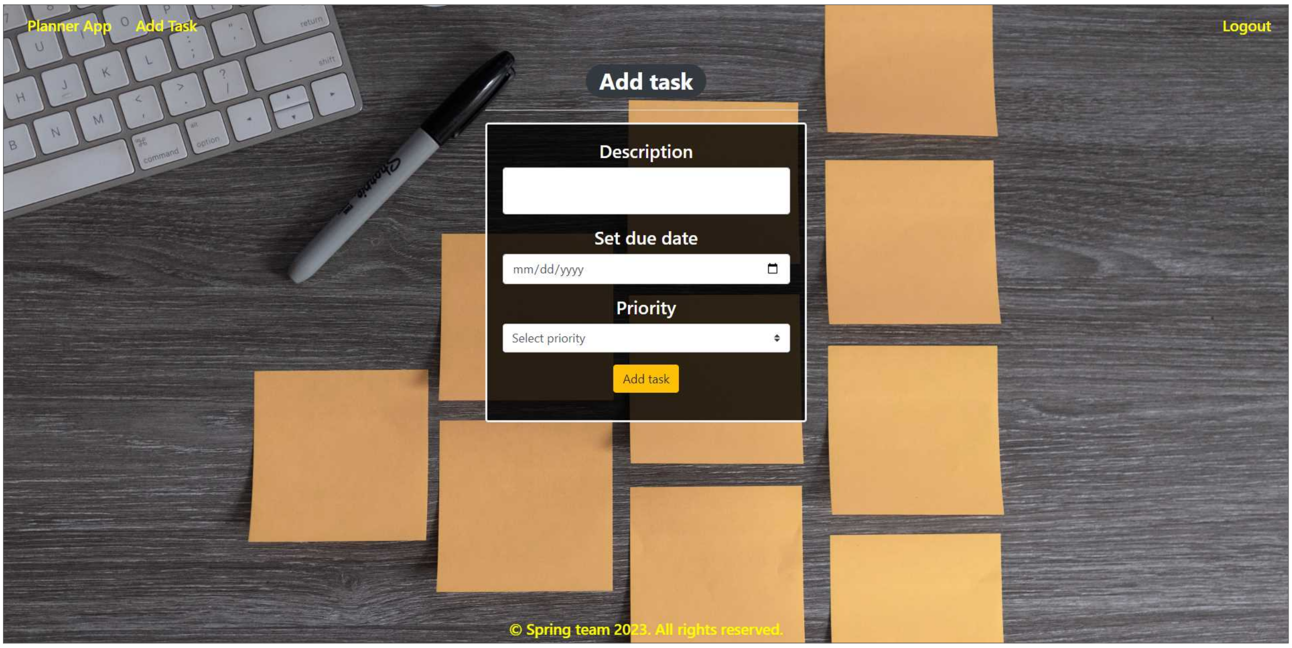
Password length must be between 3 and 20 characters!

Confirm Password

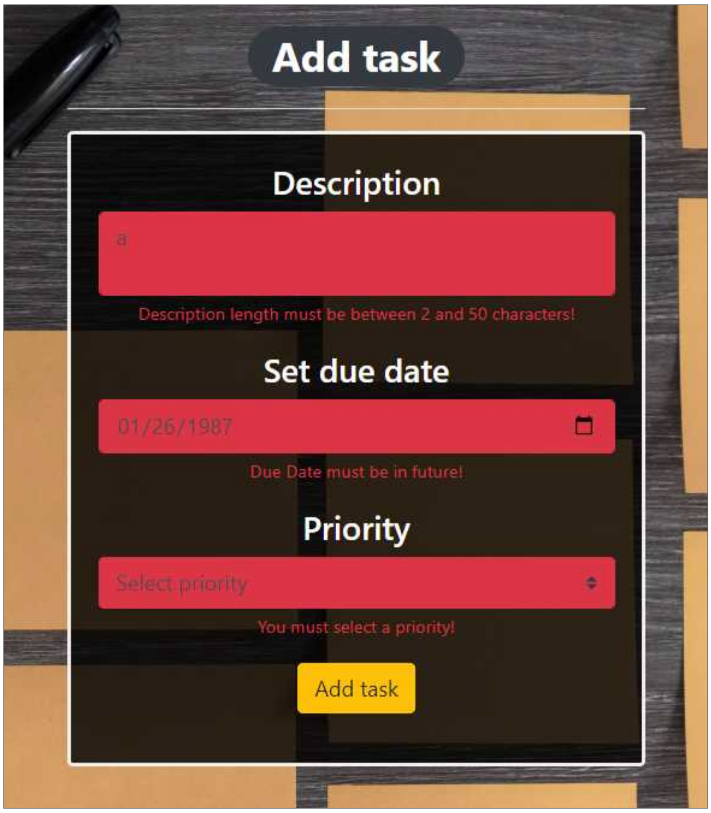
Home Page (without having any offers)

- The left section (Assigned to me) of the page should visualize **all of the current logged user tasks** from the database which he assigned to himself.
- The right section (All Available Tasks) of the page should visualize the available (not assigned) **tasks** (created tasks but not assigned).

Add Offer



Add Offer Validation

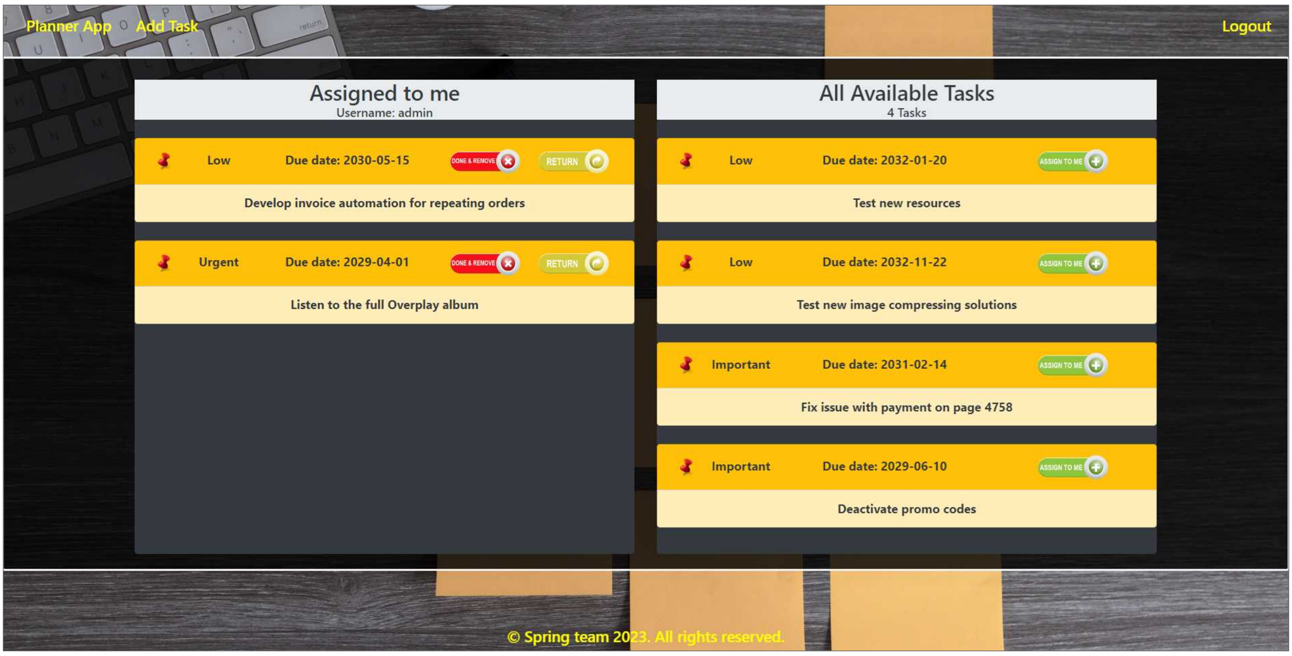


Description length must be between 2 and 50 characters!

Due Date must be in future!

You must select a priority!

Home Page (with tasks)



Assigned to me
Username: admin

Low Due date: 2030-05-15
Develop invoice automation for repeating orders

Urgent Due date: 2029-04-01
Listen to the full Overplay album

All Available Tasks
4 Tasks

Low Due date: 2032-01-20
Test new resources

Low Due date: 2032-11-22
Test new image compressing solutions

Important Due date: 2031-02-14
Fix issue with payment on page 4758

Important Due date: 2029-06-10
Deactivate promo codes

© Spring team 2023. All rights reserved.

The templates have been given to you in the application skeleton, so make sure you implement the pages correctly.

NOTES:

- The templates should look **EXACTLY** as shown above
- The templates do **NOT require additional CSS** for you to write. Only **bootstrap** and the **given CSS** are enough.

4. 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 log in, register and view the Index page.

The application should provide **Users** (logged in) with the functionality to **log out, add a new task (Add task page), view all tasks (Home page) and assign a task from "All Available Tasks" to "Assigned to me"**.

Planner Application in navbar should **redirect** to the appropriate URL depending on that if the **user is logged in**. The application should provide functionality for **adding tasks** with priorities of **Urgent, Important or Low**. The offers should be separated into different sections according to their ownership.

When the user clicks on the **ASSIGN TO ME button of some task**, he is assigning the task to **his collection (Assigned to me)**. You should **not delete** this task from DB. The task should be removed from the collection with **All Available Tasks** and should be added to the collection with **assignedTasks** of the user.

The **Done & Remove** button in "Assigned to me" section **should remove the task** from DB.

By using the **Return** button in "Assigned to me" section the user is abandoning the task and is returning it to the **All Available Tasks**. The user should unassign this task from his collection of **assignedTasks** and the **task** should no longer have user.

Bellow the "All Available Tasks" banner is located an info bar that shows the **sum of the unassigned tasks in this field**.

Bellow the "Assigned to me" banner is located an info bar that shows the **username of the current logged in user**.

The application should store its data in a MySQL database.

5. 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;
    - **Login** page;
    - **Register** page.
- **Users** (logged in) can access:
    - **Home** page;
    - **Add Task** page;
    - **Logout** functionality.