

Task Management Application

```
package com.example.taskmanagement.controller;

import com.example.taskmanagement.entity.Status;
import com.example.taskmanagement.entity.Task;
import com.example.taskmanagement.service.TaskService;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.*;

import java.util.List;

@Controller
@RequestMapping("/tasks")

public class TaskController {

    @Autowired
    private TaskService taskService;

    @GetMapping
    public String getAllTasks(Model model) {
        List<Task> tasks = taskService.getAllTasks();
        model.addAttribute("tasks", tasks);
        return "task-list"; // Returns task-list.html
    }

    @GetMapping("/{status}")
    public String getTasksByStatus(@PathVariable Status status, Model model) {
        List<Task> tasks = taskService.getTasksByStatus(status);
        model.addAttribute("tasks", tasks);
    }
}
```

```
        return "task-list";
    }
}
```

```
@GetMapping("/new")
public String showTaskForm(Model model) {
    model.addAttribute("task", new Task());
    return "task-form"; // Returns task-form.html
}
```

```
@PostMapping
public String createTask(@ModelAttribute Task task) {
    taskService.saveTask(task);
    return "redirect:/tasks";
}
```

```
@GetMapping("/edit/{id}")
public String showEditForm(@PathVariable Long id, Model model) {
    Task task = taskService.getTaskById(id);
    model.addAttribute("task", task);
    return "task-form";
}
```

```
@PostMapping("/{id}")
public String updateTask(@PathVariable Long id, @ModelAttribute Task task) {
    taskService.updateTask(id, task);
    return "redirect:/tasks";
}
```

```
@GetMapping("/delete/{id}")
public String deleteTask(@PathVariable Long id) {
    taskService.deleteTask(id);
}
```

```
        return "redirect:/tasks";
    }
}
```

```
package com.example.taskmanagement.entity;
```

```
public enum Status {
    PENDING, IN_PROGRESS, COMPLETED
}
```

```
package com.example.taskmanagement.entity;
```

```
import java.time.LocalDate;
```

```
import jakarta.persistence.Column;
import jakarta.persistence.Entity;
import jakarta.persistence.EnumType;
import jakarta.persistence.Enumerated;
import jakarta.persistence.GeneratedValue;
import jakarta.persistence.GenerationType;
import jakarta.persistence.Id;
```

```
@Entity
```

```
public class Task {
```

```
    @Id
```

```
    @GeneratedValue(strategy = GenerationType.IDENTITY)
```

```
    private Long id;
```

```
    private String title;
```

```
@Column(length = 500)
```

```
private String description;
```

```
@Enumerated(EnumType.STRING)
```

```
private Status status;
```

```
private LocalDate dueDate;
```

```
// Getters and Setters
```

```
public Long getId() {
```

```
    return id;
```

```
}
```

```
public void setId(Long id) {
```

```
    this.id = id;
```

```
}
```

```
public String getTitle() {
```

```
    return title;
```

```
}
```

```
public void setTitle(String title) {
```

```
    this.title = title;
```

```
}
```

```
public String getDescription() {
```

```
    return description;
```

```
}
```

```
public void setDescription(String description) {
```

```
        this.description = description;
    }
}
```

```
public Status getStatus() {
    return status;
}
```

```
public void setStatus(Status status) {
    this.status = status;
}
```

```
public LocalDate getDueDate() {
    return dueDate;
}
```

```
public void setDueDate(LocalDate dueDate) {
    this.dueDate = dueDate;
}
}
```

```
package com.example.taskmanagement.repository;
```

```
import com.example.taskmanagement.entity.Task;
import com.example.taskmanagement.entity.Status;
import org.springframework.data.jpa.repository.JpaRepository;
```

```
import java.util.List;
```

```
public interface TaskRepository extends JpaRepository<Task, Long> {
    List<Task> findByStatus(Status status);
}
```

```
package com.example.taskmanagement.service;

import com.example.taskmanagement.entity.Status;
import com.example.taskmanagement.entity.Task;
import com.example.taskmanagement.repository.TaskRepository;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import java.util.List;
import java.util.Optional;

@Service
public class TaskService {

    @Autowired
    private TaskRepository taskRepository;

    public List<Task> getAllTasks() {
        return taskRepository.findAll();
    }

    public List<Task> getTasksByStatus(Status status) {
        return taskRepository.findByStatus(status);
    }

    public Task getTaskById(Long id) {
        Optional<Task> task = taskRepository.findById(id);
        return task.orElse(null);
    }

    public Task saveTask(Task task) {
```

```
        return taskRepository.save(task);
    }
}
```

```
public Task updateTask(Long id, Task taskDetails) {
    Task task = getTaskById(id);
    if (task != null) {
        task.setTitle(taskDetails.getTitle());
        task.setDescription(taskDetails.getDescription());
        task.setStatus(taskDetails.getStatus());
        task.setDueDate(taskDetails.getDueDate());
        return taskRepository.save(task);
    }
    return null;
}
```

```
public void deleteTask(Long id) {
    taskRepository.deleteById(id);
}
}
```

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
<head>
    <title>Task Form</title>
</head>
<body>
    <h1>Create / Edit Task</h1>
    <form th:action="@{/tasks}" th:object="${task}" method="post">
        <input type="hidden" th:field="*{id}" />
        <label>Title: <input type="text" th:field="*{title}" /></label><br />
        <label>Description: <input type="text" th:field="*{description}" /></label><br />
    </form>
</body>
</html>
```

```
<label>Status:
  <select th:field="*{status}">
    <option value="PENDING">Pending</option>
    <option value="IN_PROGRESS">In Progress</option>
    <option value="COMPLETED">Completed</option>
  </select>
</label><br />
<label>Due Date: <input type="date" th:field="*{dueDate}" /></label><br />
<button type="submit">Save</button>
</form>
<a href="/tasks">Back to Task List</a>
</body>
</html>
```

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
<head>
  <title>Task List</title>
</head>
<body>
  <h1>Tasks</h1>
  <table>
    <tr>
      <th>Title</th>
      <th>Description</th>
      <th>Status</th>
      <th>Due Date</th>
      <th>Actions</th>
    </tr>
    <tr th:each="task : ${tasks}">
      <td th:text="${task.title}">Title</td>
```



```

        <td th:text="${task.description}">Description</td>

        <td th:text="${task.status}">Status</td>

        <td th:text="${task.dueDate}">Due Date</td>

        <td>

            <a th:href="@{/tasks/edit/{id}(id=${task.id})}">Edit</a>

            <a th:href="@{/tasks/delete/{id}(id=${task.id})}">Delete</a>

        </td>

    </tr>
</table>

<a href="/tasks/new">Create New Task</a>
</body>
</html>

```

```
spring.application.name=task-management-system
```

```
# H2 Database Configuration
```

```
spring.datasource.url=jdbc:h2:mem:testdb
```

```
spring.datasource.driverClassName=org.h2.Driver
```

```
spring.datasource.username=sa
```

```
spring.datasource.password=password
```

```
spring.sql.init.platform=h2
```

```
# Hibernate Configuration
```

```
spring.jpa.hibernate.ddl-auto=update
```

```
spring.jpa.show-sql=true
```

```
spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.H2Dialect
```

```
# H2 Console Configuration
```

```
spring.h2.console.enabled=true
```

```
spring.h2.console.path=/h2-console
```

Logging

logging.level.org.springframework=DEBUG

logging.level.root=DEBUG

logging.file.name=logs/spring-boot-app.log

logging.level.org.apache=DEBUG

spring.thymeleaf.prefix=classpath:/templates/

spring.thymeleaf.suffix=.html

spring.thymeleaf.mode=HTML

spring.thymeleaf.cache=false

server.error.whitelabel.enabled=false

spring.datasource.hikari.maximum-pool-size=10

spring.datasource.hikari.minimum-idle=2

#server.error.whitelabel.enabled=false

#server.error.include-message=always

≤?xml version="1.0" encoding="UTF-8"?>

<project xmlns="http://maven.apache.org/POM/4.0.0"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<parent>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-parent</artifactId>

<version>3.3.4</version> <!-- Keep this updated to the latest stable version -->

<relativePath/> <!-- lookup parent from repository -->

</parent>

<groupId>com.example</groupId>

<artifactId>task-management-system</artifactId>

<version>0.0.1-SNAPSHOT</version>

<name>task-management-system</name>

<description>Demo project for Spring Boot</description>

<properties>

<java.version>17</java.version> <!-- Set to your Java version -->

</properties>

<dependencies>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-data-jpa</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-web</artifactId> <!-- This includes Tomcat as the embedded server -->

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-thymeleaf</artifactId>

</dependency>

<dependency>

<groupId>com.h2database</groupId>

<artifactId>h2</artifactId>

<scope>runtime</scope>

</dependency>

```
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-test</artifactId>
  <scope>test</scope>
</dependency>
</dependencies>

<build>
  <plugins>
    <plugin>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-maven-plugin</artifactId>
      <version>3.3.4</version><!--$NO-MVN-MAN-VER$--> <!-- Specify the same version as your
parent -->
      <executions>
        <execution>
          <goals>
            <goal>repackage</goal> <!-- Ensures executable JAR is created -->
          </goals>
        </execution>
      </executions>
    </plugin>
  </plugins>
</build>
</project>
```