Assignment 1: Building a Content Management Portal with Spring, Spring MVC, JDBC Template, JSP, and Annotation-based Configurations using Maven

Objective:

The objective of this assignment is to create a Content Management Portal using Spring, Spring MVC, JDBC Template, JSP for view rendering, and incorporate annotations for configuration, all within a Maven project.

Requirements:

Project Setup:

- Create a new Maven project for your Content Management Portal.
- Configure the project to use Spring for Inversion of Control (IoC) with annotation-based configurations.
- Set up a database (e.g., MySQL or H2) and create a table named "content" with fields such as `id`, `title`, `content`, `author`, `createdDate`, and any additional fields you find relevant.

Spring IoC and JDBC Template:

- Configure Spring IoC using annotations (e.g., `@Configuration`, `@ComponentScan`, `@Bean`) within the Maven project.
- Use `JdbcTemplate` for database operations (CRUD) in a `ContentDAO` class.

Model:

- Create a `Content` class with attributes corresponding to the fields in the database table.
- Implement a `ContentDAO` class using `JdbcTemplate` to handle database operations.

Controller:

- Create a `ContentController` class with methods to handle the following actions:
    - Display a list of all content items.
    - Show a form to add a new content item.
    - Process the form submission to add a new content item with annotations for validations.
    - Display details of a specific content item.
    - Show a form to edit a content item's information.
    - Process the form submission to update a content item's information with annotations for validations.
    - Delete a content item.

View:

- Create JSP pages for displaying the list of content items, adding a new content item, editing a content item, and displaying content item details.
- Use JSTL for rendering data in the JSP pages.

Validation:

- Implement basic validation for the form submissions using annotations (e.g., `@NotBlank`, `@Size`, `@Pattern`, etc.).
- Validate number ranges and regular expressions for specific fields.

Integration:

- Integrate the `ContentController`, `ContentDAO`, and `Content` classes in the Spring application.
- Configure appropriate beans using annotations in the Spring configuration file.

Exception Handling:

- Implement exception handling for cases such as database connection issues, invalid data, etc.

Assignment 2: Enhancing a Content Management Portal with Search, Sorting, and Pagination

Objective:

The objective of this enhanced assignment is to add search, sorting, and pagination options to the existing Content Management Portal using Spring, Spring MVC, JDBC Template, JSP for view rendering, and annotations for configuration.

Requirements:

Search Functionality:

- Extend the `ContentController` to include a method for handling search requests.
- Create a search form in the JSP allowing users to input search criteria.
- Implement the search functionality in the `ContentDAO` to retrieve content items based on search criteria (e.g., title, author, content).

Sorting:

- Implement sorting options in the `ContentController` to allow users to sort content items by different criteria (e.g., title, created date).
- Enhance the UI to include sorting options in the content list view.

Pagination:

- Integrate pagination for the list of content items in the `ContentController`.
- Limit the number of items displayed per page.
- Add pagination controls in the UI to navigate through the content items.

View:

- Modify existing JSP pages to display search results along with the existing content items.
- Enhance the UI to include pagination controls and sorting options.

Integration:

- Integrate the search, sorting, and pagination functionalities into the existing `ContentController`, `ContentDAO`, and `Content` classes.
- Ensure that the UI reflects the changes and allows users to interact with sorting and pagination features.

Add ons

Implement an advanced search with multiple criteria and options.

Enhance the UI for the search form, results, sorting options, and pagination controls.

Add auto-suggestions or autocomplete for search queries.