Web Technology Final Project Guideline

A. Guideline

1. **Identify the Requirements**: First, you need to identify the requirements for your Web Application. The requirements should cover the different features and functionalities that the Web Application should have.
2. **Design the Database**: Once you have identified the requirements, you need to design the database for the Web Application. You can use any tool or paper to map out the different entities in your system and their relationships.
3. **Create a Spring Boot Project**: Create a new Spring Boot project using the Spring Initializr. Make sure to include the necessary dependencies, such as Spring Data JPA, Spring Web, etc.
4. **Set Up the Database**: Set up the database by creating the necessary tables, columns, and relationships based on the ERD you created earlier.
5. **Create the Domain Model**: Create the domain model classes that will represent the different entities in your Web Application. Make sure to annotate the classes with the necessary JPA annotations, such as @Entity, @Id, @GeneratedValue, @ManyToOne, etc.
6. **Implement the Service Layer**: Implement the service layer that will handle the business logic of your Web Application. This layer will interact with the domain model classes and perform operations such as creating, updating, and deleting records in the database.
7. **Implement the Controller Layer**: Implement the controller layer that will handle incoming requests and map them to the appropriate service methods.
8. **Implement the View Layer**: Implement the view layer that will display the data from the Web Application to the user. You can use a frontend framework like Spring MVC, Angular or React to create the views.
9. **Test Your Web Application**: Test your Web Application's frontend views to make sure that they display the data correctly.
10. **Deploy Your Web Application**: Deploy your Web Application to a cloud provider like Heroku, AWS Educate, Google Cloud Platform or DigitalOcean.

B. General Requirements

In addition to the basic CRUD (Create, Read, Update, Delete) operations, consider:

1. **Authentication and Authorization**: Implementing authentication and authorization functionality is crucial for any web application that requires user login. You can use Spring Security to secure your application and provide features such as user registration, password reset, and role-based access control.
2. **Validation**: Validating user input is essential to ensure that the data entered into your application is correct and meets the required criteria. You can use Spring Validation to implement validation rules for your forms and fields.

3. **Pagination**: When dealing with large amounts of data, pagination can help improve the performance of your application by splitting the data into smaller chunks. You can use Spring Data JPA to implement pagination for your database queries.
4. **Search and Filtering**: Providing a search and filtering functionality can help users find the data they need quickly and easily. You can use Spring Data JPA to implement search and filtering queries for your database.
5. **Error Handling**: Handling errors and exceptions gracefully is essential for providing a good user experience. You can use Spring MVC's exception-handling mechanism to catch and handle exceptions and display appropriate error messages to the user.
6. **Caching**: Caching frequently accessed data can help improve the performance of your application by reducing the number of database queries. You can use Spring Cache to implement caching for your application.
7. **File Upload and Download**: Allowing users to upload and download files can be useful for many applications. You can use Spring MVC's MultipartResolver to handle file uploads and downloads.

C. Work Submission:

Use GitHub to submit the following:

i. **Project Requirements**: Provide a clear and detailed description of the project requirements, including the purpose of the project, the expected outcomes, and any specific constraints or limitations.
ii. **Project Plan**: Share a project plan that outlines the scope of the project, the timeline, and the resources used to complete the project.
iii. **Source Code**: Provide the source code for the project, including any scripts or configuration files that are required to run the application.
iv. **Database Schema**: provide the database schema, including the table definitions and any relationships between them. (diagram)
v. **User Documentation**: Provide user documentation that explains how to use the application, including any login credentials that are required, how to navigate the application, and any features or functionality that are available.
vi. **Technical Documentation**: Provide technical documentation that explains the architecture of the application, how it was implemented, any libraries or frameworks used, and any other relevant technical details.

D. Deadline:

a. **The deadline for the implementation and deployment is 08th May 2023 at 0:00 am**
b. **Final Work Submission (with documentation) is set on 11th May 2023 at 0:00 am**

NB: Share both links (App and GitHub) via Google classroom