**To-do application using spring boot**

Requirements:

This application that allows users to create and manage tasks. The application is built using the Spring Boot framework and uses an in-memory H2 database to store task information.

Java 11

Maven 3.x

Spring web

H2 database

Build the Application:

Build the maven project in STS

Run the Application:

The application will start on port 8080 by default. You can access it by navigating to **http://localhost:8080 in your web browser.**

The application have following API endpoints:

**The controller** handles requests and responses related to “To-Do items”.

the different methods are:

**@GetMapping("/")** - This method handles the request to the home page. It retrieves all ToDo items from the database .which is then passed to the "index" view, **(index.html).**

**@GetMapping("/createtodo")** - This method handles the request to add a new ToDo item. It creates an instance of the ToDoModel and adds it to the Model object, which is then passed to the "add" view, **(add.html).**

**@PostMapping("/createtodo")** - This method handles the submission of a new ToDo item. It saves the new item to the database and redirects back to the index page.

**@GetMapping("/update/{id}")** - This method handles the request to update an existing ToDo item. It retrieves the ToDo item with the given ID from the database and adds it to the Model object, which is then passed to the "update" view, **(update.html).**

**@PostMapping("/update/{id}")** - This method handles the submission of the updated ToDo item. It saves the updated item to the database and redirects back to the index page.

**@GetMapping("/delete/{id}")** - This method handles the request to delete a ToDo item. It deletes the item with the given ID from the database and redirects back to the index page.

**@GetMapping("/edit/{id}") -** This method handles the request to mark a ToDo item as complete or incomplete. It retrieves the ToDo item with the given ID from the database, toggles its "complete" status, saves it to the database, and redirects back to the index page.

**Create Entity Class” ToDoModel”:**

This class represents a To-Do item in a To-Do list. It contains fields for the To-Do item's,

ID

description

completion status.

**Flow diagram**

Diagram

Description automatically generated

Diagram

Description automatically generated

Index.html

Graphical user interface, application

Description automatically generated

Database

Graphical user interface, text, application

Description automatically generated