

## Assignment 2 – Web-based To-Do App

### Objectives:

- Gain experience in full-stack web development
- Develop the skills and understanding for frontend and backend integration

Your task is to build a **To-Do List** web application. The application has a Vue2 client for the UI and saves data through a Spring-based REST server to an H2 database. The requirements are as follows:

UI reference: <https://dribbble.com/shots/5325611-Daily-UI-42-ToDo-List> (for reference only!)

### Basic Requirements:

1. Users should be able to create, edit and delete to-do items (text content only).
2. Users should be able to browse a list of to-do items.
3. Users should be able to toggle the state of items between **pending** and **completed**.
4. Users should be able to hide or show completed items.
5. To-do items should be saved to the database automatically, assuming connectivity is always available.
6. The app should be visually appealing using plain CSS or the [Bootstrap framework](#).
7. The app should be written using Vue and Spring frameworks only. No other programming languages or libraries are allowed.
8. The app should run in a web browser and be accessible to all users, regardless of the device or screen size.
9. The app should provide **schema.sql** and **data.sql** as the default schema and sample data to initialize the database for testing.
10. The programs should have proper comments to explain the code.

### Advanced Requirements:

1. Limit each page to display 5 items and provide pagination for more items.
2. Support offline mode. Save to-do items in the client's local storage and synchronise with the server's database when there is connectivity. The client needs to check the server's status constantly and indicate the current working mode in the UI.
3. Implement server-side session to authenticate users with username and password.

### Submission and Evaluation:

1. The app will be evaluated based on its functionality, user experience, and the quality of the code and documentation.
2. The app should be submitted to Canvas as a zip archive of your Maven project, along with a brief explanation (readme.txt) of how to start the app (e.g. username/password for a test user).