UI task

I have developed the UI in the JAVA Rest Api project IDE itself. The github repository is same for both rest api and ui Task. UI files are present inside static directory under resources. Inside static the below files are present.

Index.html

Style.css

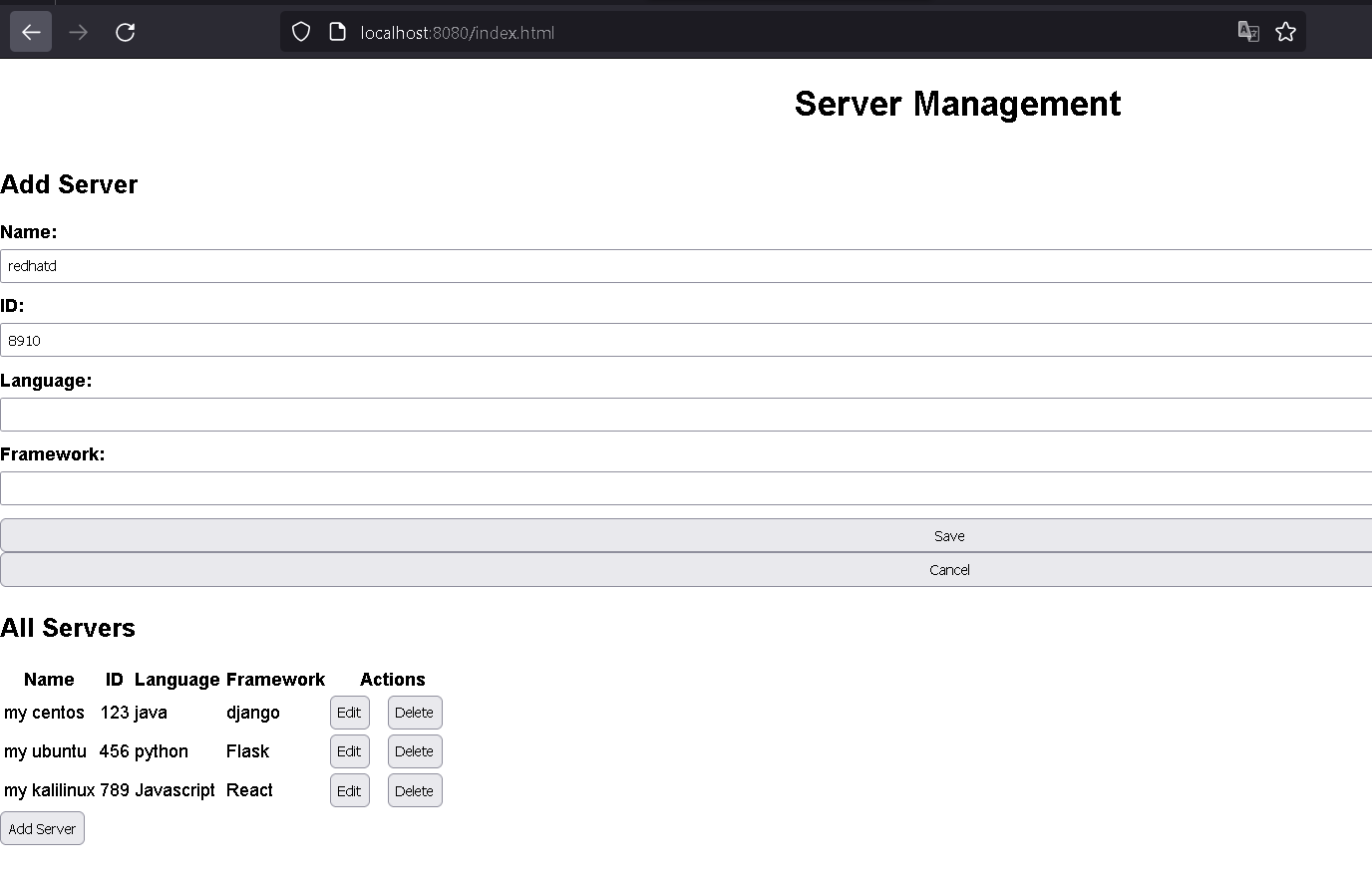
App.js

Server.js

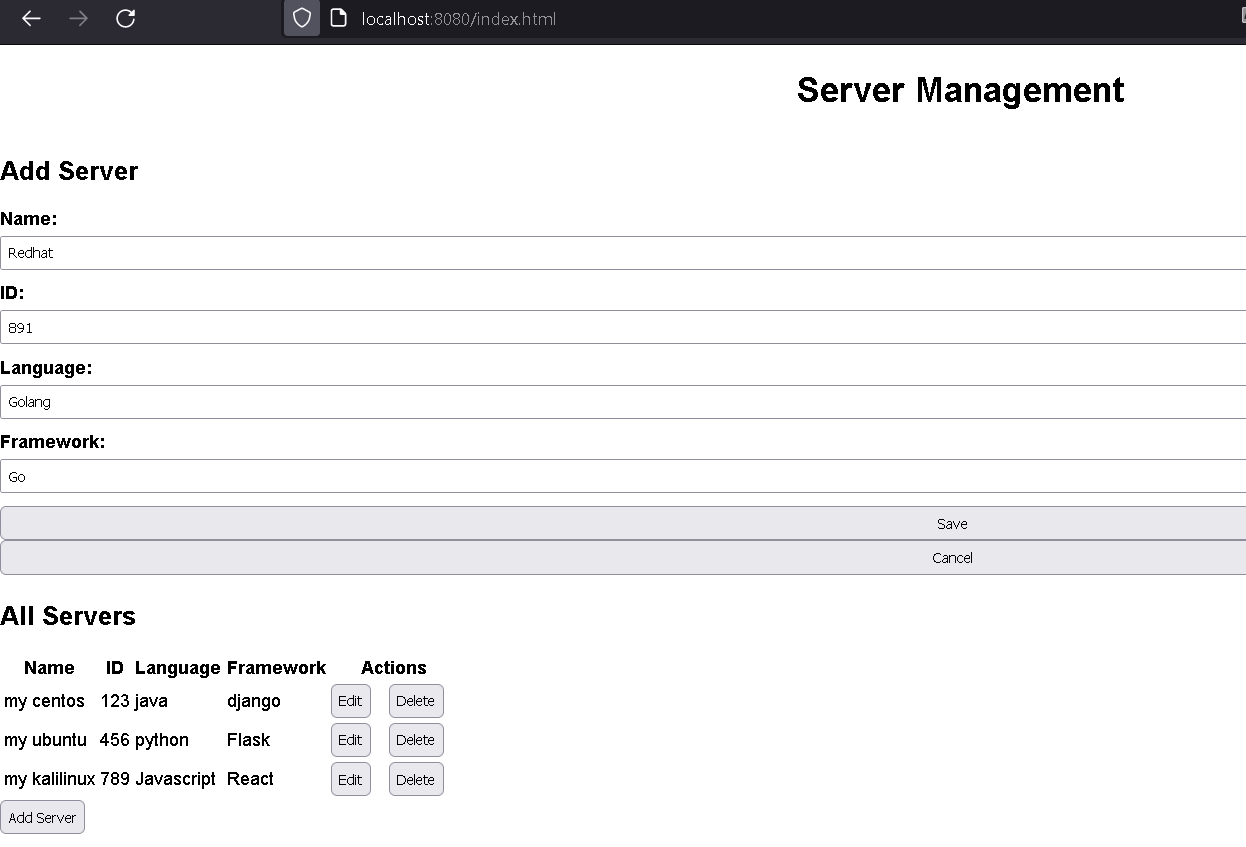
**Ui design**

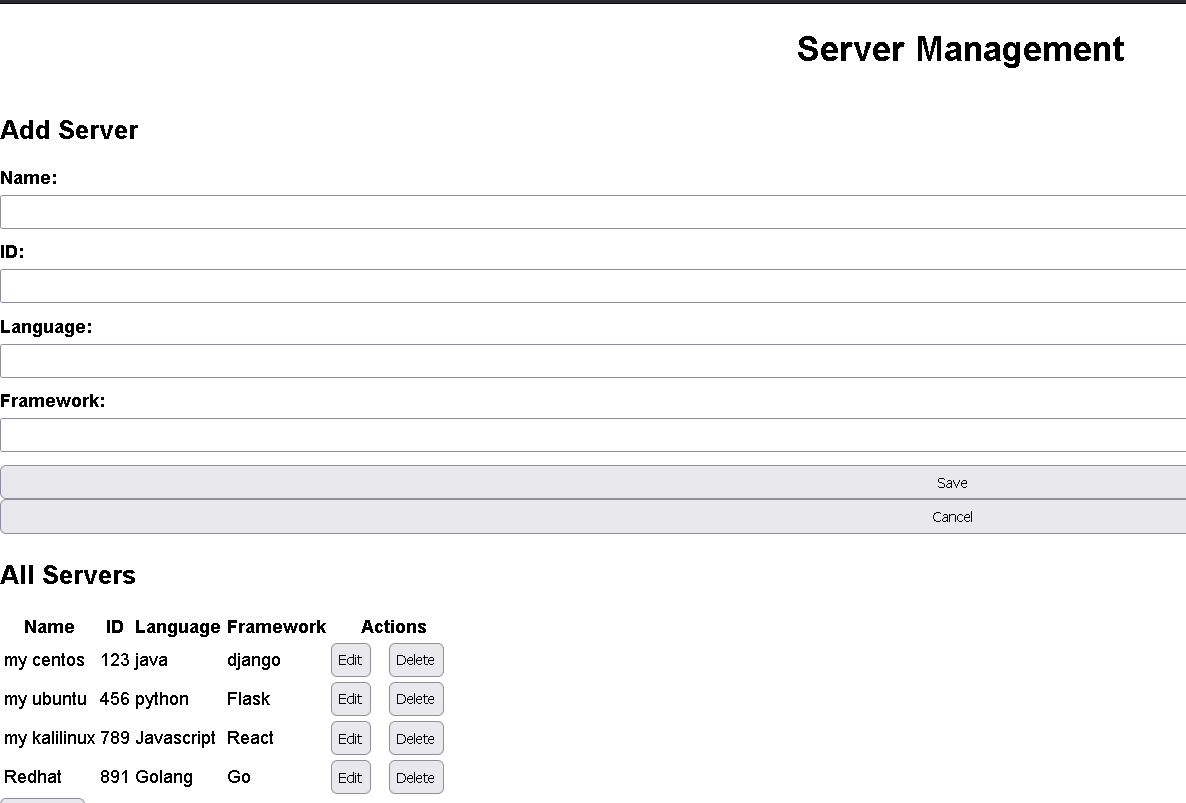
You should be able to create, show and delete records from your UI

MainUI



For example adding a new server. Type the server details and click save





We can see that the server is added.

Explanation of the main UI functioning files:

**Server.js**

The purpose of this code is to provide a Node.js server that allows for managing a collection of servers in a MongoDB database through a RESTful API.

The server is built using the Express.js framework and the Mongoose library is used to connect to the MongoDB database, define the server schema, and create the server model.

The API has five routes that allow for retrieving all servers, retrieving a single server by ID, adding a new server, updating an existing server, and deleting a server. These routes use the corresponding methods provided by the Mongoose library to interact with the MongoDB database.

The server also serves a static HTML file located in the "public" directory.

When the server starts, it listens on a specified port, which can be set through the PORT environment variable. If the PORT environment variable is not set, the server listens on port 3000.

**App.js**

This is a JavaScript code that provides functionality for managing a collection of servers through a web interface.

The code includes event listeners for various buttons and forms on the page, including the form for adding/editing a server, the cancel button, the edit button, the delete button, and the add server button.

When the user submits the server form, the code sends a POST or PUT request to the server API to add or update a server in the database, respectively. If the user clicks the cancel button, the form is cleared and any editing state is reset. If the user clicks the edit button or delete button for a specific server in the table, the code sends a GET or DELETE request to the server API to retrieve or delete the corresponding server from the database.

The code also includes a function to clear the form, a function to update the table with the latest server data from the server API, and an initial call to the updateTable function to load the table with the existing servers on page load.

The purpose of this code is to provide a simple web interface for managing a collection of servers, allowing users to add, edit, and delete servers in the database through a user-friendly interface. The code uses the Fetch API to send HTTP requests to the server API, and updates the table dynamically with the latest server data.