Front-End Essentials

Exam

Document Code	25e-BM/HR/HDCV/FSOFT
Version	1.1
Effective Date	20/11/2012

RECORD OF CHANGES

No	Effective Date	Change Description	Reason	Reviewer	Approver
1	26/Jan/2024	Create a new Exam	Create new		

Contents

Problem: Todolist Application 4

CODE:	FEA.Practice.T102
TYPE:	Long
LOC:	N/A
DURATION:	150 MINUTES

Working tools and delivery requirements

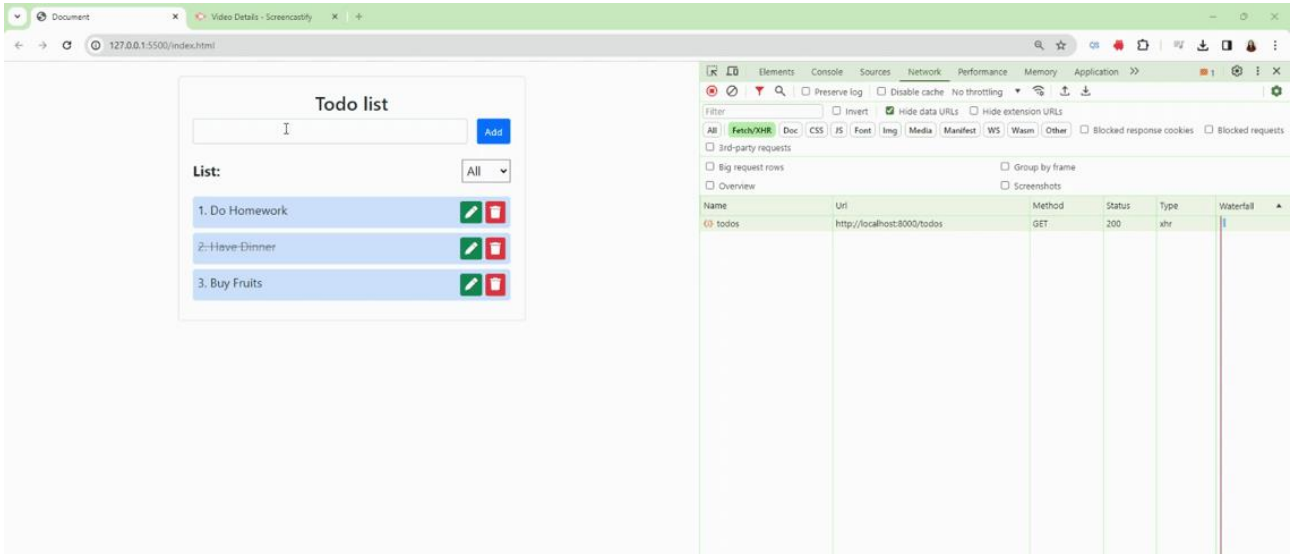
- **Working tools**: Visual Studio Code
- **Delivery**: Source code in a compressed archive (zip extension is a must).

Technologies

- Interact with API server to CRUD data.
- Use Bootstrap for styling.
- Follow the standard naming and coding convention.

Problem: Todolist Application

Your task is to build a Todo List Application that looks like below (correct spacing and icons are not required)



Specification:

1. Must use Bootstrap and JavaScript to interact with API server.
2. The application should correctly render the information from data API's response.
3. Users should be able to add a new todo by clicking the "Add" button or press "Enter" after entering its name in the input text box. The todo name should be cleared after each addition, and a new todo should not be added when the input text box is empty.
4. Users should be able to toggle the status of a todo by clicking the todo's name. A completed todo should have a line-through style, with the text color set to #aaa.
5. Users should be able to edit a todo by clicking the pen icon. Upon clicking the pen icon, the todo's name will be displayed in an input text field for editing. The "Add" button will toggle to an "Save" button. After clicking the "Save" button, the corresponding todo will be updated, and the "Save" button will toggle back to the "Add" button.

6. Users should be able to filter todos by selecting “All”, “Todo” (completed: false) or “Done” (completed: true) in the select element.
7. Users should be able to delete a todo by clicking the trash icon. Make sure to properly render the order number of each todo after a deletion.

APIs:

Backend Server are provided in a zip file named "mock-data.zip"
Unzip, then start the Server by run `npm install`, then run `npm start` on cmd. You should see the output:

BASE URL: http://localhost:8888

Endpoints:

Get todoList:

```
GET /todos
```

Add a new todo:

```
POST /todos
Example: {
“name”: “string”,
“completed”: boolean
}
```

Update a single todo by its id

```
PATCH /todos/id
Example: {“completed”: boolean}  //update status
Example: {“name”: “string”}  //update name
```

Filter todos by status

```
GET /todos?completed=true
```

Delete a single todo by its id

```
DELETE /todos/id
```