



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	19/Dec/2023	Create a new Exam	Create new		

Contents

Problem: Todolist Application	4
-------------------------------------	---

		CODE: TYPE: LOC: DURATION:	FEE.Exams.Opt3 Long N/A 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

The product illustrates:

- Front-end Essentials
- Practice to use JavaScript to calculate value

Technical Requirements

- Follow the standard naming and coding convention.
- Add appropriate comments for each class, method, attribute, algorithms, ...
- Create Project named <YOUR ACCOUNT>_fee.exams.opt3 and put every file inside

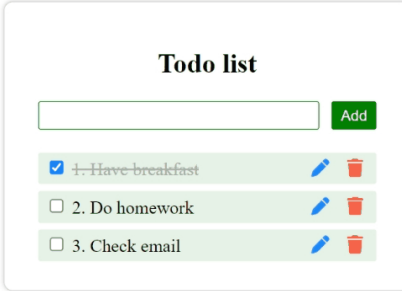
Example if your account is AnhNV then you must create a project called: **AnhNV_fee.exams.opt3**

Problem: Todolist Application

Given an array of todos:

```
var todolist = [  
  {  
    name: "Have breakfast",  
    completed: true,  
    id: 1,  
  },  
  {  
    name: "Do homework",  
    completed: false,  
    id: 2,  
  },  
  {  
    name: "Check email",  
    completed: false,  
    id: 3,  
  },  
];
```

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



The screenshot shows a web application titled "Todo list". At the top, there is a text input field and a green "Add" button. Below the input field, there is a list of three todo items, each on a light green background. The first item is "1. Have breakfast" with a checked checkbox, a blue pen icon, and a red trash icon. The second item is "2. Do homework" with an unchecked checkbox, a blue pen icon, and a red trash icon. The third item is "3. Check email" with an unchecked checkbox, a blue pen icon, and a red trash icon.

Specification:

1. Must use plain HTML, CSS and JavaScript without utilizing any frameworks or libraries. However, you are allowed to use icons and fonts.
2. The application should correctly render the information from the provided array.
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 checking or unchecking the checkbox associated with it. 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 **delete** a todo by clicking the **trash** icon. Make sure to properly render the order number of each todo after a deletion.