

Name – Bhagawat Daulatrao Karhale

Roll no – 22cs3033

T1.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Shopping List</title>
  <style>
    body {
      font-family: Arial, sans-serif;
    }
    #shopping-list {
      margin: 20px;
    }
    #items {
      list-style-type: none;
      padding: 0;
    }
    #items li {
      margin-bottom: 5px;
    }
  </style>
</head>
<body>

<div id="shopping-list">
  <h1>Shopping List</h1>
  <input type="text" id="item" placeholder="Enter item">
  <button id="addItemBtn">Add Item</button>
  <ul id="items"></ul>
</div>

<script>
  class Model {
    constructor() {
      this.items = JSON.parse(localStorage.getItem('shoppingList')) || [];
    }

    addItem(item) {
      this.items.push(item);
      this.save();
    }
  }

  const model = new Model();

  document.getElementById('addItemBtn').addEventListener('click', () => {
    const item = document.getElementById('item').value;
    if (item) {
      model.addItem(item);
    }
  });
```

```

    }

    getItems() {
        return this.items;
    }

    save() {
        localStorage.setItem('shoppingList', JSON.stringify(this.items));
    }
}

class View {
    constructor() {
        this.itemsList = document.getElementById('items');
        this.itemInput = document.getElementById('item');
        this.addItemBtn = document.getElementById('addItemBtn');
    }

    getNewItem() {
        return this.itemInput.value.trim();
    }

    clearInput() {
        this.itemInput.value = '';
    }

    renderItems(items) {
        this.itemsList.innerHTML = '';
        items.forEach(item => {
            const li = document.createElement('li');
            li.textContent = item;
            this.itemsList.appendChild(li);
        });
    }
}

class Controller {
    constructor(model, view) {
        this.model = model;
        this.view = view;

        this.view.addItemBtn.addEventListener('click', () => this.addItem());

        this.initialRender();
    }

    initialRender() {
        this.view.renderItems(this.model.getItems());
    }
}

```

```

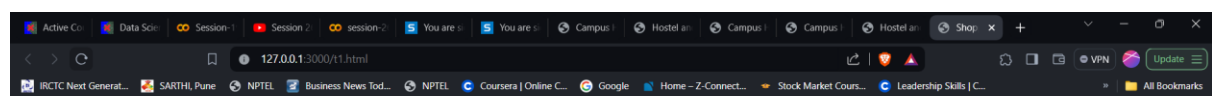
    }

    addItem() {
      const newItem = this.view.getNewItem();
      if (newItem !== '') {
        this.model.addItem(newItem);
        this.view.renderItems(this.model.getItems());
        this.view.clearInput();
      }
    }
  }
}

const model = new Model();
const view = new View();
const controller = new Controller(model, view);
</script>

</body>
</html>

```



T2.

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Shopping List</title>

```

```

<style>
  body {
    font-family: Arial, sans-serif;
  }
  #shopping-list {
    margin: 20px;
  }
  #items {
    list-style-type: none;
    padding: 0;
  }
  #items li {
    margin-bottom: 5px;
  }
</style>
</head>
<body>

<div id="shopping-list">
  <h1>Shopping List</h1>
  <input type="text" id="item" placeholder="Enter item">
  <button id="addItemBtn">Add Item</button>
  <ul id="items"></ul>
  <p>You have accessed this page <span id="pageAccessCount"></span> times.</p>
</div>

<script>
class Model {
  constructor() {
    this.items = JSON.parse(localStorage.getItem('shoppingList')) || [];
    this.accessCount = localStorage.getItem('accessCount') || 0;
  }

  addItem(item) {
    this.items.push(item);
    this.save();
  }

  getItems() {
    return this.items;
  }

  save() {
    localStorage.setItem('shoppingList', JSON.stringify(this.items));
  }

  incrementAccessCount() {
    this.accessCount++;
  }
}

```

```

        localStorage.setItem('accessCount', this.accessCount);
    }

    getAccessCount() {
        return this.accessCount;
    }
}

class View {
    constructor() {
        this.itemsList = document.getElementById('items');
        this.itemInput = document.getElementById('item');
        this.addItemBtn = document.getElementById('addItemBtn');
        this.pageAccessCount = document.getElementById('pageAccessCount');
    }

    getNewItem() {
        return this.itemInput.value.trim();
    }

    clearInput() {
        this.itemInput.value = '';
    }

    renderItems(items) {
        this.itemsList.innerHTML = '';
        items.forEach(item => {
            const li = document.createElement('li');
            li.textContent = item;
            this.itemsList.appendChild(li);
        });
    }

    renderAccessCount(count) {
        this.pageAccessCount.textContent = count;
    }
}

class Controller {
    constructor(model, view) {
        this.model = model;
        this.view = view;

        this.view.addItemBtn.addEventListener('click', () => this.addItem());

        this.initialRender();
    }
}

```

```

initialRender() {
  this.view.renderItems(this.model.getItems());
  this.view.renderAccessCount(this.model.getAccessCount());
  this.model.incrementAccessCount();
}

addItem() {
  const newItem = this.view.getNewItem();
  if (newItem !== '') {
    this.model.addItem(newItem);
    this.view.renderItems(this.model.getItems());
    this.view.clearInput();
  }
}
}

const model = new Model();
const view = new View();
const controller = new Controller(model, view);
</script>

</body>
</html>

```

