

## Web Technology Lab Assignment-7

---

T1: code:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>MVC Shopping List</title>
  <style>
    body {
      font-family: Arial, sans-serif;
    }
    #shopping-list {
      margin-top: 20px;
    }
  </style>
</head>
<body>
  <h1>MVC Shopping List</h1>

  <form id="add-item-form">
    <input type="text" id="item" placeholder="Enter item">
    <button type="submit">Add Item</button>
  </form>

  <ul id="shopping-list">
  </ul>

  <script>
    const model = {
      items: [],

      addItem: function(item) {
        this.items.push(item);
        this.saveItems();
      },

      getItems: function() {
        return this.items;
      },
    };
```

```

        saveItems: function() {
            localStorage.setItem('shoppingList', JSON.stringify(this.items));
        },

        loadItems: function() {
            const savedItems = localStorage.getItem('shoppingList');
            if (savedItems) {
                this.items = JSON.parse(savedItems);
            }
        }
    };

    const view = {
        itemList: document.getElementById('shopping-list'),

        render: function() {
            this.itemList.innerHTML = '';
            const items = controller.getItems();
            items.forEach(item => {
                const li = document.createElement('li');
                li.textContent = item;
                this.itemList.appendChild(li);
            });
        }
    };

    const controller = {
        init: function() {
            model.loadItems();
            view.render();
            this.setupEventListeners();
        },

        setupEventListeners: function() {
            const form = document.getElementById('add-item-form');
            form.addEventListener('submit', function(event) {
                event.preventDefault();
                const itemInput = document.getElementById('item');
                const newItemText = itemInput.value.trim();
                if (newItemText !== '') {
                    model.addItem(newItemText);
                    view.render();
                    itemInput.value = '';
                }
            });
        },

        getItems: function() {

```

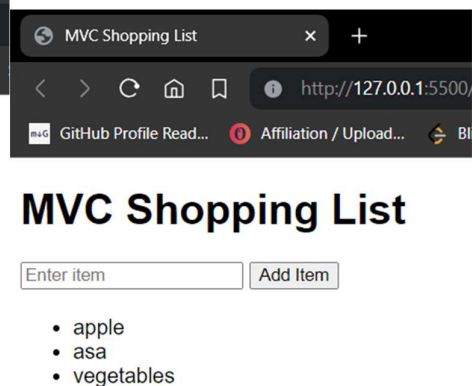
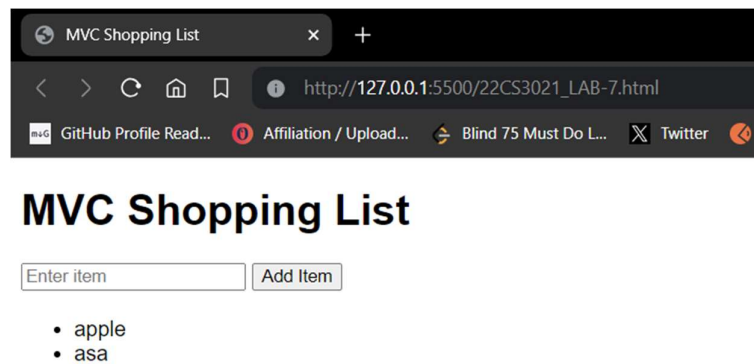
```

        return model.getItems();
    }
};

controller.init();
</script>
</body>
</html>

```

Web Page:



T2: Code:

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>MVC Shopping List</title>
  <style>
    body {
      font-family: Arial, sans-serif;
    }
    #shopping-list {
      margin-top: 20px;
    }
  </style>
</head>
<body>
  <h1>MVC Shopping List</h1>

```

```
<form id="add-item-form">
  <input type="text" id="item" placeholder="Enter item">
  <button type="submit">Add Item</button>
</form>

<ul id="shopping-list">
</ul>

<p id="visit-count">Number of times visited: <span id="count">0</span></p>

<script>
  const model = {
    items: [],

    addItem: function(item) {
      this.items.push(item);
      this.saveItems();
    },

    getItems: function() {
      return this.items;
    },

    saveItems: function() {
      localStorage.setItem('shoppingList', JSON.stringify(this.items));
    },

    loadItems: function() {
      const savedItems = localStorage.getItem('shoppingList');
      if (savedItems) {
        this.items = JSON.parse(savedItems);
      }
    },

    getVisitCount: function() {
      let count = localStorage.getItem('visitCount');
      if (count === null) {
        count = 0;
      } else {
        count = parseInt(count);
      }
      return count;
    },

    incrementVisitCount: function() {
      let count = this.getVisitCount();
      count++;
    }
  };
</script>
```

```

        localStorage.setItem('visitCount', count);
    }
};

const view = {
    itemList: document.getElementById('shopping-list'),

    visitCountSpan: document.getElementById('count'),

    render: function() {
        this.itemList.innerHTML = '';
        const items = controller.getItems();
        items.forEach(item => {
            const li = document.createElement('li');
            li.textContent = item;
            this.itemList.appendChild(li);
        });
    },

    updateVisitCount: function(count) {
        this.visitCountSpan.textContent = count;
    }
};

const controller = {
    init: function() {
        model.loadItems(); // Load items from local storage
        view.render();
        this.setupEventListeners();
        this.updateVisitCount();
    },

    setupEventListeners: function() {
        const form = document.getElementById('add-item-form');
        form.addEventListener('submit', function(event) {
            event.preventDefault();
            const itemInput = document.getElementById('item');
            const newItemText = itemInput.value.trim();
            if (newItemText !== '') {
                model.addItem(newItemText);
                view.render();
                itemInput.value = '';
            }
        });
    },

    updateVisitCount: function() {
        const count = model.getVisitCount();
    }
};

```

```

        view.updateVisitCount(count);
        model.incrementVisitCount();
    },

    getItems: function() {
        return model.getItems();
    }
};

controller.init();
</script>
</body>
</html>

```

Web Page:

MVC Shopping List

- apple
- asa
- vegetables
- vegetables
- reret

Number of times visited: 12

MVC Shopping List

- apple
- asa
- vegetables
- vegetables
- reret
- apple

Number of times visited: 14