

# ASSIGNMENT - 6 JIGYASU PATEL - 22CS3031

## index.HTML

```
<!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>

  <link rel="stylesheet" href="styles.css">

</head>

<body>

  <h1>Shopping List</h1>

  <div id="items-container"></div>

  <form id="add-item-form">

    <input type="text" id="new-item" placeholder="Add new item">

    <input type="text" id="item-image-url" placeholder="Enter image URL">

    <button type="submit">Add</button>

  </form>

  <script src="shoppingList.js"></script>

  <script src="shoppingListView.js"></script>

  <script src="shoppingListController.js"></script>

  <script>

    const model = new ShoppingListModel();

    const view = new ShoppingListView();
```

```
        const controller = new ShoppingListController(model, view);

    </script>

</body>

</html>
```

## styles.css

```
body {

    font-family: Arial, sans-serif;

    margin: 0;

    padding: 0;

}
```

```
h1 {

    text-align: center;

}
```

```
#items-container {

    margin-top: 20px;

}
```

```
.item {

    display: flex;

    align-items: center;

    margin-bottom: 5px;

}
```

```
.item input[type="checkbox"] {
```

```
margin-right: 10px;
```

```
}
```

```
.completed {
```

```
text-decoration: line-through;
```

```
}
```

```
.item img {
```

```
width: 100px;
```

```
height: 100px;
```

```
object-fit: cover; /* Ensures image maintains aspect ratio and fills the specified dimensions */
```

```
}
```

## shoppingList.js

```
class ShoppingListModel {
```

```
constructor() {
```

```
  this.items = [];
```

```
}
```

```
addItem(itemName, imageUrl) {
```

```
  this.items.push({
```

```
    name: itemName,
```

```
    imageUrl: imageUrl,
```

```
    completed: false
```

```
  });
```

```
}
```

```
toggleItemCompletion(index) {  
  
    this.items[index].completed = !this.items[index].completed;  
  
}  
  
}
```

## shoppingListView.js

```
class ShoppingListView {  
  
    constructor() {  
  
        this.itemsContainer = document.getElementById('items-container');  
  
    }  
  
    addItem(item, index, toggleItemCompletion) {  
  
        const itemDiv = document.createElement('div');  
  
        itemDiv.classList.add('item');  
  
  
  
        const checkbox = document.createElement('input');  
  
        checkbox.type = 'checkbox';  
  
        checkbox.checked = item.completed;  
  
        checkbox.addEventListener('change', () => {  
  
            toggleItemCompletion(index);  
  
        });  
  
  
        const label = document.createElement('label');  
  
        label.textContent = item.name;  
  
        label.classList.toggle('completed', item.completed);  
  

```

```
const img = document.createElement('img');

img.src = item.imageUrl;

img.onerror = function() {

    img.src = 'placeholder.jpg'; // Placeholder image if provided image URL fails to load

};

itemDiv.appendChild(checkbox);

itemDiv.appendChild(label);

itemDiv.appendChild(img);

this.itemsContainer.appendChild(itemDiv);

}

}
```

## shoppingListController.js

```
class ShoppingListController {

    constructor(model, view) {

        this.model = model;

        this.view = view;

        this.view.addItem = this.view.addItem.bind(this.view);

        this.model.toggleItemCompletion = this.model.toggleItemCompletion.bind(this.model);

        this.init();

    }

}
```

```

init() {

  document.getElementById('add-item-form').addEventListener('submit', (event) => {

    event.preventDefault();

    const newItemInput = document.getElementById('new-item');

    const newItemValue = newItemInput.value.trim();

    const itemImageUrlInput = document.getElementById('item-image-url');

    const itemImageUrl = itemImageUrlInput.value.trim();

    if (newItemValue !== '') {

      this.model.addItem(newItemValue, itemImageUrl);

      newItemInput.value = '';

      itemImageUrlInput.value = '';

      this.updateView();

    }

  });

  this.updateView();

}

updateView() {

  this.view.itemsContainer.innerHTML = '';

  this.model.items.forEach((item, index) => {

    this.view.addItem(item, index, this.model.toggleItemCompletion);

  });

}

}

```

index.htmlstyles.cssshoppingList.jsshoppingListView.jsshoppingListController.js

script1.js

+

424zggqu4

NEWHTML▼RUN▶⋮⌵

1<!DOCTYPE html>

2<html lang="en">

3<head>

4<meta charset="UTF-8">

5<meta name="viewport" content="width=device-width, initial-scale=1.0">

6<title>Shopping List</title>

7<link rel="stylesheet" href="styles.css">

8</head>

9<body>

10<h1>Shopping List</h1>

11<div id="items-container"></div>

12<form id="add-item-form">

13<input type="text" id="new-item" placeholder="Add new item">

14<input type="text" id="item-image-url" placeholder="Enter image URL">

15<button type="submit">Add</button>

16</form>

17

18<script src="shoppingList.js"></script>

19<script src="shoppingListView.js"></script>

20<script src="shoppingListController.js"></script>

21<script>

22const model = new ShoppingListModel();

23const view = new ShoppingListView();

24const controller = new ShoppingListController(model, view);

25</script>


26</body>

27</html>


28

Shopping List

☐ Milk



☐ Ata



Add new item

Enter image URL

Add