Intermediate JavaScript Programming

LESSON 10: Building a Shopping List Web App (Vanilla JS + Bootstrap)

Learning Objectives:

By the end of this session, participants will be able to:

- Design a static web page using Bootstrap for layout and style.
- Use plain JavaScript to manage a list of interactive data.
- Build a simple shopping list app where users can add items and mark them as completed.
- Deploy their completed app using GitHub Pages.

Lesson Outline:

I. Project Overview (5 min)

Students will create a web page where users can:

- Add items to a shopping list
- Mark items as complete
- Remove items

This app will use **Bootstrap** for appearance and **vanilla JavaScript** for behavior.

II. File Structure and Setup (10 min)

Create the following files:

```
shopping-list/
  index.html
  script.js
```

Bootstrap Setup: In `index.html`, include in the `<head>`:

```
<link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.
css" rel="stylesheet">
```

III. HTML Structure (5 min)

```
<body>
<div class="container mt-4">
  <h2>Shopping List</h2>
```

```
  <div class="input-group mb-3">
    <input id="itemInput" type="text" class="form-control"
placeholder="Add item">
    <button id="addButton" class="btn btn-success">Add</button>
  </div>

  <ul id="itemList" class="list-group"></ul>
</div>

<script src="script.js"></script>
</body>
```

IV. JavaScript Logic (10 min)

In `script.js`:

```
const itemInput = document.getElementById('itemInput');
const addButton = document.getElementById('addButton');
const itemList = document.getElementById('itemList');

addButton.addEventListener('click', addItem);

function addItem() {
  const name = itemInput.value.trim();
  if (!name) return;

  const li = document.createElement('li');
  li.className = 'list-group-item d-flex justify-content-between align-
items-center';

  const span = document.createElement('span');
  span.textContent = name;
  span.style.cursor = 'pointer';
  span.addEventListener('click', () => {
    span.classList.toggle('text-decoration-line-through');
  });

  const button = document.createElement('button');
  button.textContent = '×';
  button.className = 'btn btn-sm btn-outline-danger';
  button.addEventListener('click', () => {
    itemList.removeChild(li);
  });

  li.appendChild(span);
  li.appendChild(button);
  itemList.appendChild(li);

  itemInput.value = '';
```

```
  itemInput.focus();
}
```

V. Deployment with GitHub Pages (10 min)

1. Create a new GitHub repo: `shopping-list`
2. Add your `index.html` and `script.js` to the repo
3. Commit and push to `main` branch
4. Go to Settings > Pages and publish the `main` branch root

Example URL:

```
https://your-username.github.io/shopping-list/
```

VI. Recap & Next Steps (5 min)

- Bootstrap provided layout and style
- JavaScript handled DOM interaction and list behavior
- GitHub Pages makes the app public and shareable

Final Multiple-Choice Question:

What does `addEventListener` do in JavaScript? A. Adds a visual style to an element B. Binds a function to run when a specific event occurs C. Downloads a CSS file D. Automatically submits a form

(Answer: B. Binds a function to run when a specific event occurs)