

LAB EXERCISE 06 – NODEJS WEB SERVER

Program :-

App.js

```
document.addEventListener('DOMContentLoaded', function () {  
    const todoList = document.getElementById('todo-list');  
    const todoInput = document.getElementById('todo-input');  
    const addButton = document.getElementById('add-button');  
  
    // Function to create a new todo item and append it to the list  
    function addTodo() {  
        const taskText = todoInput.value.trim();  
        if (taskText === "") return;  
  
        const listItem = document.createElement('li');  
        listItem.innerHTML = `  
            <span>${taskText}</span>  
            <button class="delete-button">Delete</button>  
        `;  
  
        // Add an event listener to delete the task when the "Delete" button is clicked  
        const deleteButton = listItem.querySelector('.delete-button');  
        deleteButton.addEventListener('click', function () {  
            todoList.removeChild(listItem);  
        });  
  
        todoList.appendChild(listItem);  
    }  
});
```

```

        todoInput.value = "";
    }

    // Add a new task when the "Add" button is clicked
    addButton.addEventListener('click', addTodo);

    // Add a new task when the "Enter" key is pressed
    todoInput.addEventListener('keydown', function (event) {
        if (event.key === 'Enter') {
            addTodo();
        }
    });
});

```

Index.html

```

<!DOCTYPE html>
<html>
<head>
    <title>Todo App</title>
    <link rel="stylesheet" type="text/css" href="styles.css">
</head>
<body>
    <h1>Todo List</h1>
    <ul id="todo-list">
        <!-- Todo items will be added here dynamically -->
    </ul>
    <input type="text" id="todo-input" placeholder="Add a new task">

```

```
<button id="add-button">Add</button>
<script src="app.js"></script>
</body>
</html>
```

Style.css

```
body {
  font-family: Arial, sans-serif;
  text-align: center;
  background-color: #f5f5f5;
  margin: 0;
  padding: 0;
}
```

```
h1 {
  background-color: #007bff;
  color: #fff;
  padding: 20px;
  margin: 0;
}
```

```
#todo-list {
  list-style-type: none;
  padding: 0;
}
```

```
li {
```

```
background-color: #fff;
padding: 10px;
margin: 5px 0;
border: 1px solid #ddd;
border-radius: 5px;
display: flex;
justify-content: space-between;
}
```

```
#todo-input {
  width: 70%;
  padding: 5px;
}
```

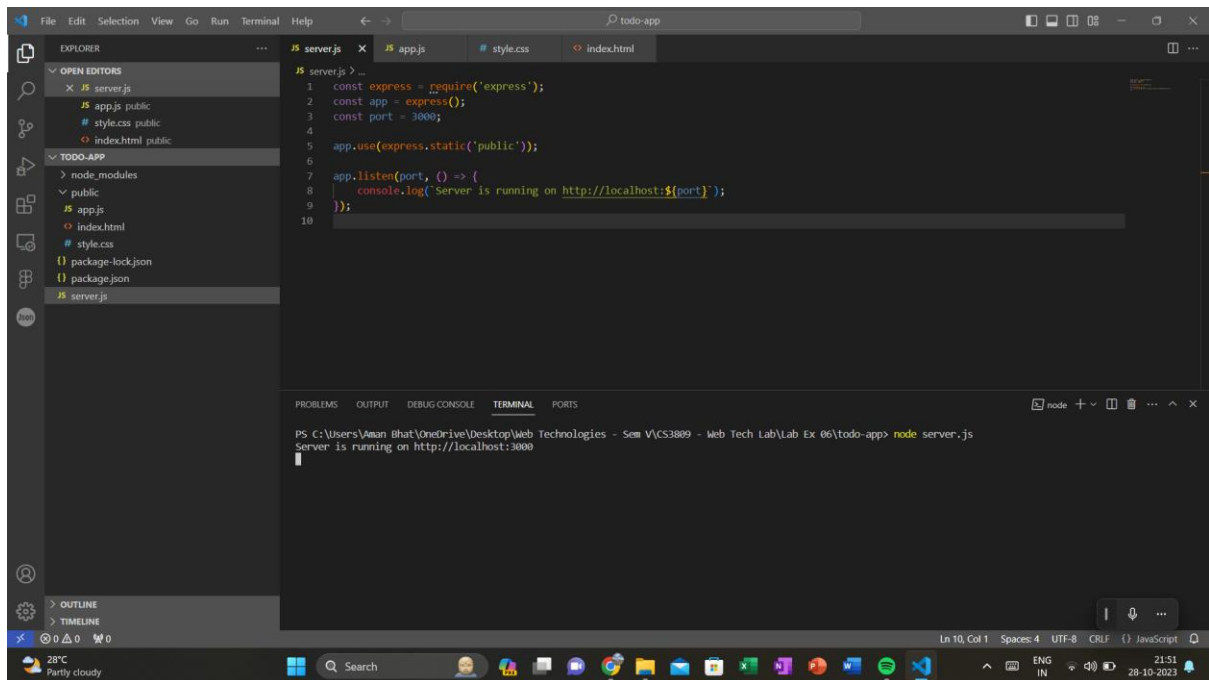
```
#add-button {
  background-color: #007bff;
  color: #fff;
  padding: 5px 10px;
  border: none;
  cursor: pointer;
}
```

```
#add-button:hover {
  background-color: #0056b3;
}
```

Server.js

```
const express = require('express');  
const app = express();  
const port = 3000;  
  
app.use(express.static('public'));  
  
app.listen(port, () => {  
  console.log(`Server is running on http://localhost:${port}`);  
});
```

Output:-



The screenshot shows the Visual Studio Code editor with a project named 'todo-app'. The Explorer sidebar on the left shows the file structure: 'server.js', 'app.js', 'style.css', and 'index.html' in the 'public' folder, and 'node_modules', 'package-lock.json', and 'package.json' in the root. The main editor window displays the 'server.js' file with the following code:

```
1 const express = require('express');
2 const app = express();
3 const port = 3000;
4
5 app.use(express.static('public'));
6
7 app.listen(port, () => {
8   console.log('Server is running on http://localhost:${port}');
9 });
10
```

The TERMINAL panel at the bottom shows the command 'node server.js' being executed, resulting in the output: 'Server is running on http://localhost:3000'.

