TODO LIST

AIM:

To create a web application for managing a TO-DO list of users using Visual Studio Code, we need to use a combination of technologies, including HTML, CSS, JavaScript, language like Node.js with Express for server-side functionality and a database (such as MongoDB) for storing the TO-DO items and user data.

PROCEDURE:

Set up your development environment:

➢ Install Visual Studio Code (if you haven't already).
➢ Install Node.js and npm (Node Package Manager) to manage
dependencies. ➢ Install MongoDB to store the TO-DO items and user data.

Create a new project in Visual Studio Code:

➢ Open Visual Studio Code and create a new folder for your project.
➢ Open the project folder in Visual Studio Code.

Initialize a new Node.js project:

➢ Open the terminal in Visual Studio Code.
Run the following command to create a new package.json file:

```
npm init -y
```

Install necessary dependencies:

Install Express to create a server:

```
npm install express
```

Install Mongoose to interact with MongoDB:

```
npm install mongoose
```

Install other required dependencies like bcrypt, jsonwebtoken, etc., for user authentication and authorization.

Set up the frontend:
Create a folder in your project directory.
Inside the client folder, create files for your frontend, such as **index.html, style.css, and**

app.js. Design your frontend using HTML and CSS.

index.html - Basic HTML structure for the frontend.

```
<!DOCTYPE html>
<html lang="en">
<head>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
        <script src="app.js"></script>
        <link rel="stylesheet" href="styles.css">
        <title>To Do List</title>
</head>
<body>
```

```html
<header class="bg-success text-white p-5">
<div class="container">
<div class="row">
<div class="col-lg-12 col-md-12 col-sm-12">
<font face="Comic sans MS" size="11" color="red">
<strong>ToDo List</strong>
</font>
</div> </div> </div>
</header>
<div class="container mt-3">
<h2>Add Items</h2>
<label id="lblsuccess" class="text-success" style="display: none;">
</label>
<form id="addForm">
<div class="row">
<div class="col-lg-7 col-md-7 col-sm-7">
<input type="text" onkeyup="toggleButton(this, 'submit')"
class="form-control" id="item">
</div>
<div class="col-lg-5 col-md-5 col-sm-5">
<input type="submit" class="btn btn-dark" id="submit" value="Submit" disabled>
</div>
</div>
</form>
<h3 class="mt-4">Tasks</h3>
<form id="addForm">
<ul class="list-group" id="items"></ul>
</form>
</div>
</body>
</html>
```

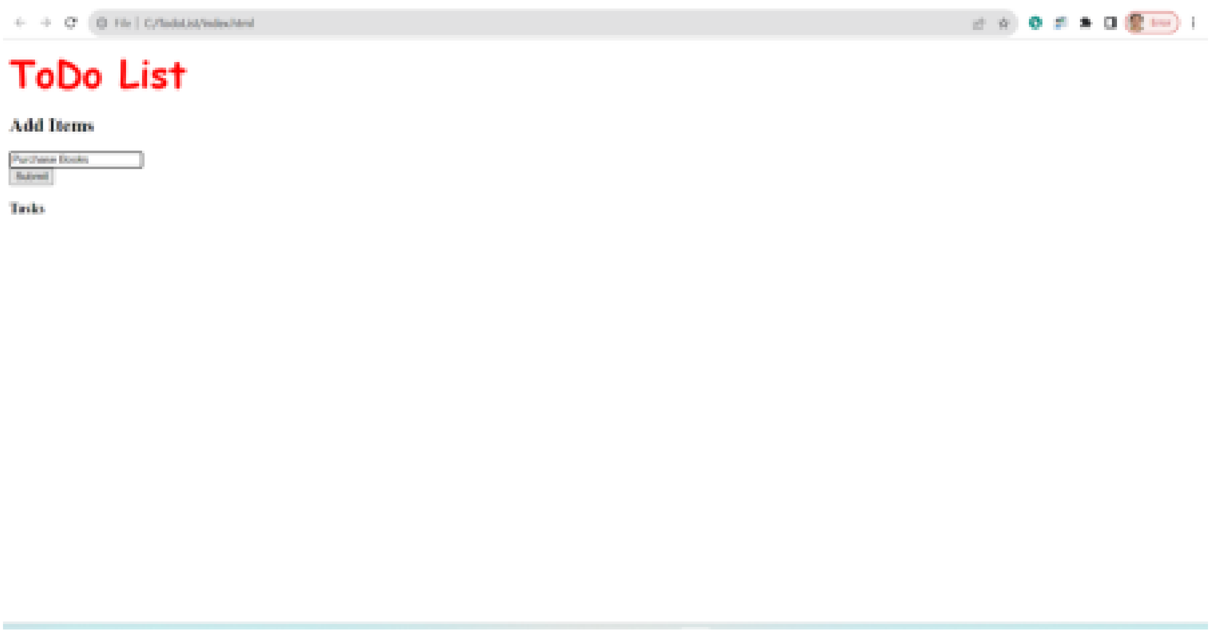
//style.css

```css
body {
 font-family: Arial, sans-serif;
 }

 button {
 background-color: #007bff;
 color: #fff;
 border: none;
 padding: 10px 20px;
 font-size: 1rem;
 cursor: pointer;
 }

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

//app.js

```javascript
window.onload = () => { const form1 = document.querySelector("#addForm");
let items = document.getElementById("items");
let submit = document.getElementById("submit");
let editItem = null;
form1.addEventListener("submit", addItem);
items.addEventListener("click", removeItem);
};
function addItem(e) {
e.preventDefault();
if (submit.value != "Submit") {
console.log("Hello");
editItem.target.parentNode.childNodes[0].data = document.getElementById("item").value;
submit.value = "Submit";
document.getElementById("item").value = "";
document.getElementById("lblsuccess").innerHTML = "Text edited successfully";
document.getElementById("lblsuccess").style.display = "block";
setTimeout(function() {
document.getElementById("lblsuccess").style.display = "none";}, 3000);
return false;
}
let newItem = document.getElementById("item").value;
if (newItem.trim() == "" || newItem.trim() == null)
return false;
else
document.getElementById("item").value = "";
let li = document.createElement("li");
li.className = "list-group-item";
let deleteButton = document.createElement("button");
deleteButton.className = "btn-danger btn btn-sm float-right delete";
deleteButton.appendChild(document.createTextNode("Delete"));
let editButton = document.createElement("button");
editButton.className = "btn-success btn btn-sm float-right edit";
editButton.appendChild(document.createTextNode("Edit"));
li.appendChild(document.createTextNode(newItem));
li.appendChild(deleteButton);
li.appendChild(editButton);
items.appendChild(li);
}
function removeItem(e) {
e.preventDefault();
if (e.target.classList.contains("delete")) {
if (confirm("Are you Sure?")) {
let li = e.target.parentNode;
items.removeChild(li);
document.getElementById("lblsuccess").innerHTML = "Text deleted successfully";
document.getElementById("lblsuccess").style.display = "block";
setTimeout(function() {
document.getElementById("lblsuccess").style.display = "none";}, 3000);
} }
if (e.target.classList.contains("edit")) {
document.getElementById("item").value = e.target.parentNode.childNodes[0].data;
submit.value = "EDIT";
editItem = e;
} }
function toggleButton(ref, btnID) {
document.getElementById(btnID).disabled = false; }
```

OUTPUT:
//open index.html in browser (Front-End)


File | C:/TodoList/index.html

ToDo List

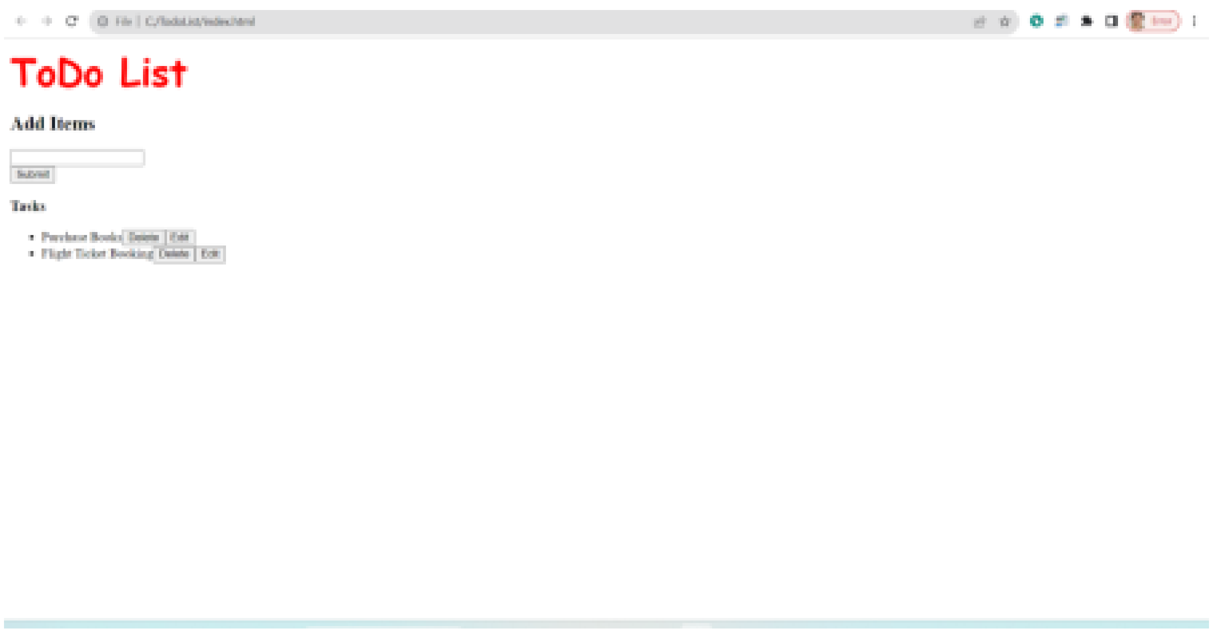
Add Items

Purchase Books

Submit

Tasks

//Add tasks in the list


File | C:/TodoList/index.html

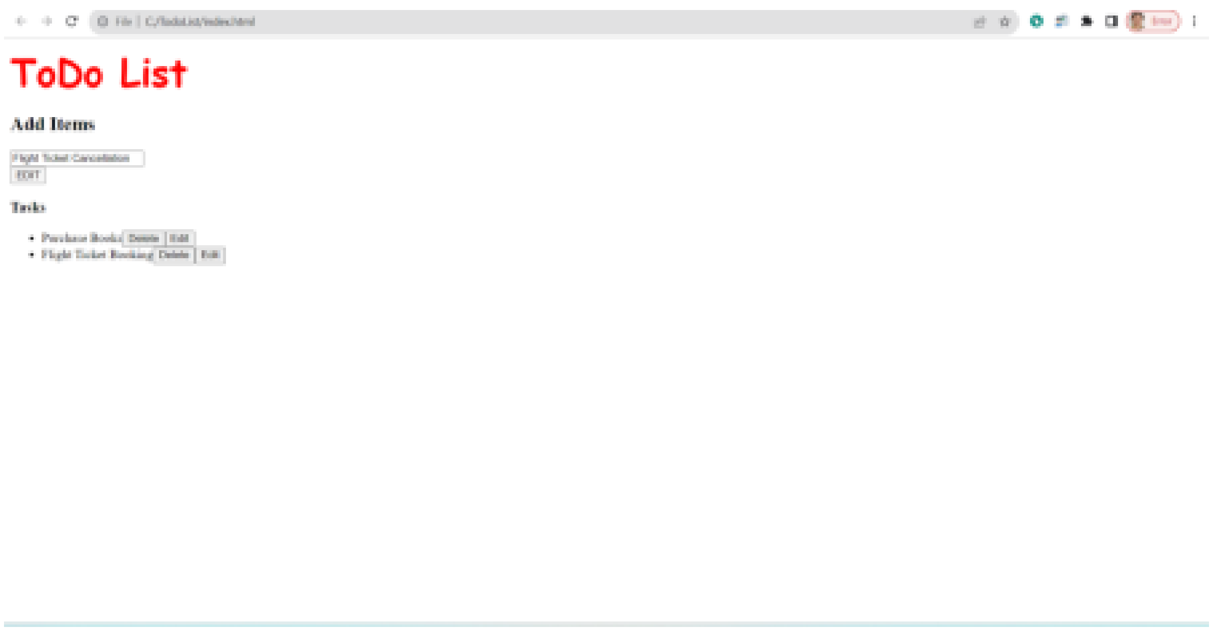
ToDo List

Add Items

Submit

Tasks

- Purchase Books Delete Edit
- Flight Ticket Booking Delete Edit


File | C:/TodoList/index.html

ToDo List

Add Items

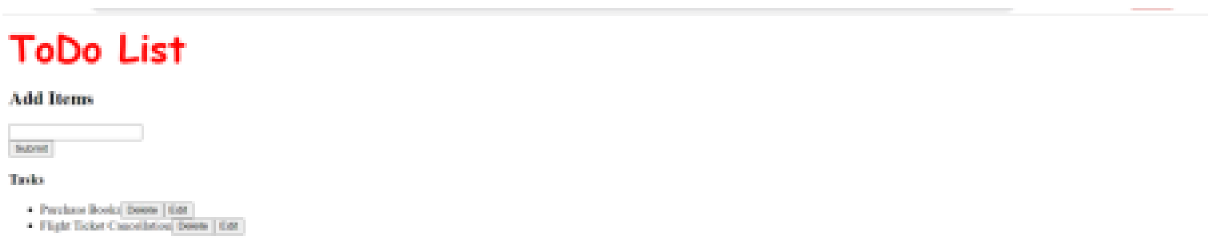
Flight Ticket Cancellation

EDIT

Tasks

- Purchase Books Delete Edit
- Flight Ticket Booking Delete Edit

//Edit a task list created prior

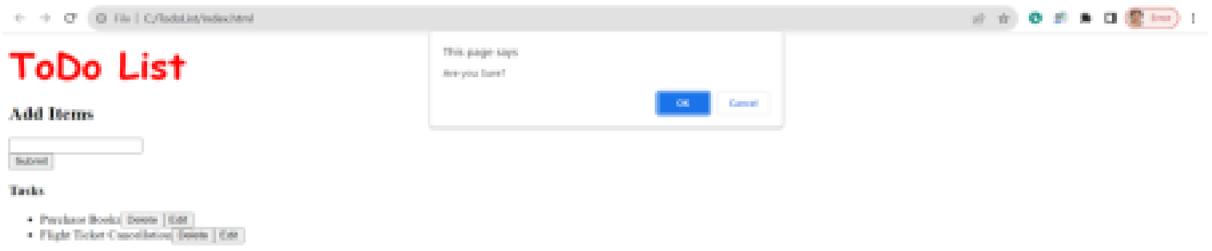

ToDo List

Add Items

Tasks

- Purchase Books Delete Edit
- Flight Ticket Cancellation Delete Edit

// **Delete a Todo list from the created lists**


File | C:/TodoList/index.html

This page says

Are you Sure?

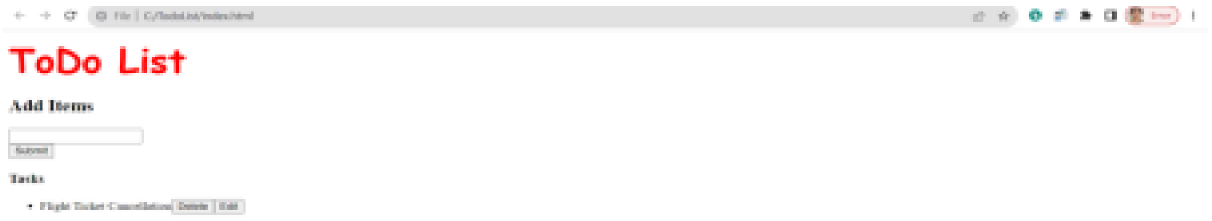
ToDo List

Add Items

Tasks

- Purchase Books Delete Edit
- Flight Ticket Cancellation Delete Edit

//After deletion


File | C:/TodoList/index.html

ToDo List

Add Items

Tasks

- Flight Ticket Cancellation Delete Edit

RESULT:

Thus, a simple web application using various front-end technologies was developed to perform list management for TO-DO list of users successfully.