**Browser Tabs Management:**

**Implement browser tab management where each open tab is a node in a singly linked list. Users can open new tabs, close specific ones, and switch between them. Optimize the solution for handling thousands of open tabs efficiently.**

<html lang="en">

<head>

<title>Browser Tabs Management</title>

<style>

body {

font-family: Arial, sans-serif;

display: flex;

flex-direction: column;

align-items: center;

margin: 20px;

}

#tab-list {

display: flex;

flex-wrap: wrap;

max-width: 80%;

margin-top: 20px;

}

.tab {

padding: 10px 15px;

margin: 5px;

border-radius: 5px;

cursor: pointer;

background-color: #4CAF50;

color: white;

}

.active-tab {

background-color: #2E7D32;

}

button {

padding: 10px 20px;

margin: 5px;

font-size: 1rem;

cursor: pointer;

}

</style>

</head>

<body>

<h1>Browser Tabs Management</h1>

<div>

<button onclick="openNewTab()">Open New Tab</button>

<button onclick="closeCurrentTab()">Close Current Tab</button>

<button onclick="switchToNextTab()">Switch to Next Tab</button>

</div>

<div id="tab-list"></div>

<script>

class TabNode {

constructor(id) {

this.id = id;

this.next = null;

}

}

class TabManager {

constructor() {

this.head = null;

this.current = null;

this.idCounter = 1;

}

openNewTab() {

const newTab = new TabNode(this.idCounter++);

if (!this.head) {

this.head = newTab;

this.current = newTab;

} else {

let temp = this.head;

while (temp.next) {

temp = temp.next;

}

temp.next = newTab;

}

this.renderTabs();

}

closeCurrentTab() {

if (!this.current) return;

if (this.current === this.head) {

this.head = this.current.next;

this.current = this.head;

} else {

let prev = this.head;

while (prev.next && prev.next !== this.current) {

prev = prev.next;

}

prev.next = this.current.next;

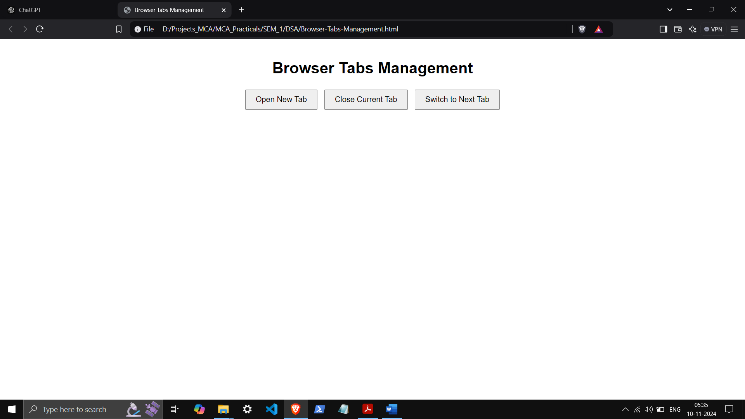
this.current = prev.next || this.head;

}

this.renderTabs();

}

switchToNextTab() {

 if (!this.current) return;

this.current = this.current.next || this.head;

this.renderTabs();

}

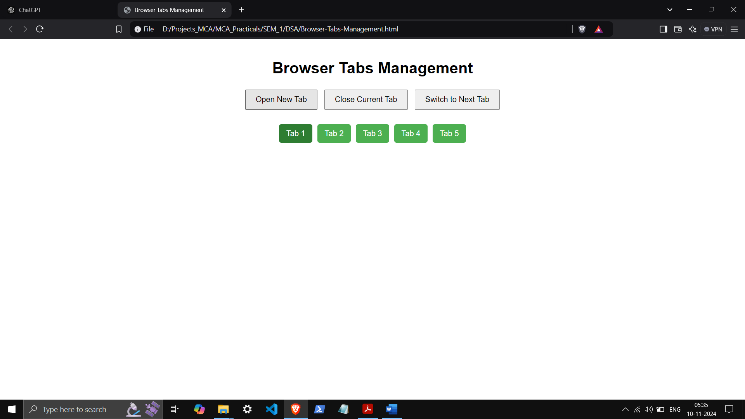
renderTabs() {

const tabList = document.getElementById("tab-list");

tabList.innerHTML = '';

let temp = this.head;

while (temp) {

 const tabElement = document.createElement("div");

tabElement.classList.add("tab");

if (temp === this.current) {

tabElement.classList.add("active-tab");

}

tabElement.textContent = `Tab ${temp.id}`;

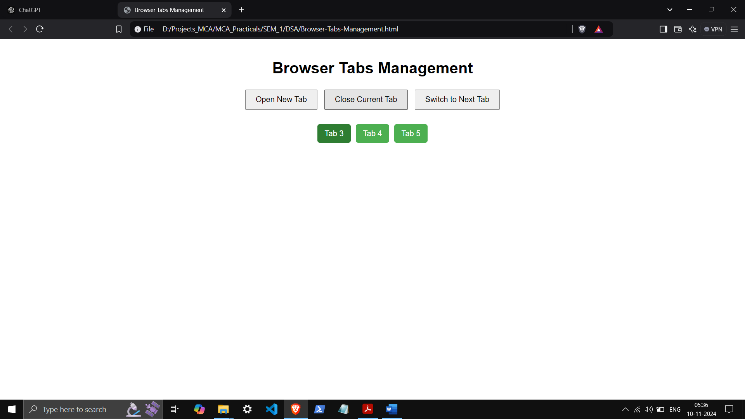
tabList.appendChild(tabElement);

temp = temp.next;

}

}

}



const tabManager = new TabManager();

function openNewTab() {

tabManager.openNewTab();

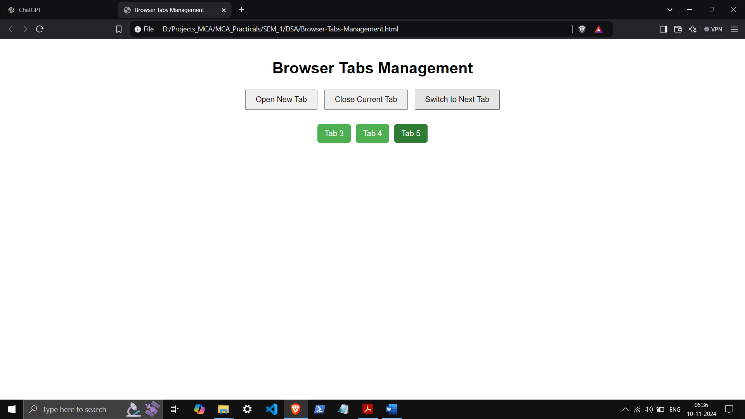
}

function closeCurrentTab() {

tabManager.closeCurrentTab();

}

function switchToNextTab() {

 tabManager.switchToNextTab();

}

</script>

</body>

</html>