**Implementation of EduLightHub Learning Platform**

**1. Core Structure and Layout**

The foundation of the platform would start with the HTML structure and Tailwind CSS styling. Here's how the main layout might be structured:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>EduLightHub - Learning Platform for BTech Students</title>

<link href="https://cdn.jsdelivr.net/npm/tailwindcss@2.2.19/dist/tailwind.min.css" rel="stylesheet">

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0-beta3/css/all.min.css">

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

</head>

<body class="bg-gray-50 font-sans">

<!-- Navigation Header -->

<header class="bg-white shadow-sm fixed w-full z-10">

<nav class="container mx-auto px-4 py-3 flex justify-between items-center">

<div class="flex items-center space-x-2">

<img src="logo.svg" alt="EduLightHub Logo" class="h-8">

<span class="text-indigo-600 font-bold text-xl">EduLightHub</span>

</div>

<div class="hidden md:flex space-x-6 text-gray-600">

<a href="#dashboard" class="hover:text-indigo-600 transition-colors">Dashboard</a>

<a href="#courses" class="hover:text-indigo-600 transition-colors">Courses</a>

<a href="#certificates" class="hover:text-indigo-600 transition-colors">Certificates</a>

<a href="#resources" class="hover:text-indigo-600 transition-colors">Resources</a>

</div>

<!-- User Profile Dropdown -->

<div class="relative" id="userDropdown">

<button class="flex items-center space-x-2 focus:outline-none">

<img src="avatar.jpg" alt="User Avatar" class="h-8 w-8 rounded-full">

<span class="hidden md:inline text-gray-700">Rahul Singh</span>

<i class="fas fa-chevron-down text-xs text-gray-500"></i>

</button>

<!-- Dropdown Menu (Hidden by Default) -->

<div class="absolute right-0 mt-2 w-48 bg-white rounded-md shadow-lg py-1 hidden" id="dropdownMenu">

<a href="#profile" class="block px-4 py-2 text-sm text-gray-700 hover:bg-gray-100">

<i class="fas fa-user mr-2"></i> Profile

</a>

<a href="#settings" class="block px-4 py-2 text-sm text-gray-700 hover:bg-gray-100">

<i class="fas fa-cog mr-2"></i> Settings

</a>

<a href="#logout" class="block px-4 py-2 text-sm text-red-600 hover:bg-gray-100">

<i class="fas fa-sign-out-alt mr-2"></i> Logout

</a>

</div>

</div>

</nav>

</header>

<!-- Main Content -->

<main class="container mx-auto px-4 pt-24 pb-12">

<!-- Dashboard Section -->

<section id="dashboard" class="mb-12">

<!-- Dashboard Content Goes Here -->

</section>

<!-- Courses Section -->

<section id="courses" class="mb-12">

<!-- Courses Content Goes Here -->

</section>

<!-- Other Sections... -->

</main>

<!-- Footer -->

<footer class="bg-gray-800 text-white py-8">

<div class="container mx-auto px-4">

<!-- Footer Content -->

</div>

</footer>

<!-- Modal Templates (Hidden by Default) -->

<div id="loginModal" class="fixed inset-0 bg-black bg-opacity-50 flex items-center justify-center z-50 hidden">

<!-- Login Form -->

</div>

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

</body>

</html>

This foundation sets up a responsive layout with a fixed navigation header, main content sections, and a footer. The user dropdown and modals are initially hidden and will be toggled with JavaScript.

## 2. Dashboard Implementation

The dashboard would show student progress metrics and active courses:

<section id="dashboard" class="mb-12">

<h2 class="text-2xl font-bold text-gray-800 mb-6">Your Learning Dashboard</h2>

<!-- Learning Metrics -->

<div class="grid grid-cols-1 md:grid-cols-4 gap-4 mb-8">

<!-- Courses In Progress -->

<div class="bg-white rounded-lg shadow p-6 flex items-center space-x-4">

<div class="bg-blue-100 p-3 rounded-full">

<i class="fas fa-book-open text-blue-600 text-xl"></i>

</div>

<div>

<p class="text-sm text-gray-500">Courses in Progress</p>

<p class="text-2xl font-bold text-gray-800">4</p>

</div>

</div>

<!-- Completed Courses -->

<div class="bg-white rounded-lg shadow p-6 flex items-center space-x-4">

<div class="bg-green-100 p-3 rounded-full">

<i class="fas fa-check-circle text-green-600 text-xl"></i>

</div>

<div>

<p class="text-sm text-gray-500">Completed Courses</p>

<p class="text-2xl font-bold text-gray-800">12</p>

</div>

</div>

<!-- Learning Hours -->

<div class="bg-white rounded-lg shadow p-6 flex items-center space-x-4">

<div class="bg-purple-100 p-3 rounded-full">

<i class="fas fa-clock text-purple-600 text-xl"></i>

</div>

<div>

<p class="text-sm text-gray-500">Learning Hours</p>

<p class="text-2xl font-bold text-gray-800">86</p>

</div>

</div>

<!-- Certificates Earned -->

<div class="bg-white rounded-lg shadow p-6 flex items-center space-x-4">

<div class="bg-yellow-100 p-3 rounded-full">

<i class="fas fa-medal text-yellow-600 text-xl"></i>

</div>

<div>

<p class="text-sm text-gray-500">Certificates Earned</p>

<p class="text-2xl font-bold text-gray-800">8</p>

</div>

</div>

</div>

<!-- Progress Section -->

<div class="bg-white rounded-lg shadow p-6 mb-8">

<h3 class="text-lg font-semibold text-gray-800 mb-4">Your Learning Progress</h3>

<!-- Progress Bar -->

<div class="mb-6">

<div class="flex justify-between mb-1">

<span class="text-sm font-medium text-gray-700">Overall Completion</span>

<span class="text-sm font-medium text-gray-700">68%</span>

</div>

<div class="w-full bg-gray-200 rounded-full h-2.5">

<div class="bg-indigo-600 h-2.5 rounded-full" style="width: 68%"></div>

</div>

</div>

<!-- Course Progress List -->

<div class="space-y-4">

<!-- Course 1 -->

<div class="border-b pb-4">

<div class="flex justify-between mb-1">

<span class="font-medium text-gray-800">Data Structures & Algorithms</span>

<span class="text-sm font-medium text-gray-700">85%</span>

</div>

<div class="w-full bg-gray-200 rounded-full h-2.5">

<div class="bg-green-500 h-2.5 rounded-full" style="width: 85%"></div>

</div>

</div>

<!-- Course 2 -->

<div class="border-b pb-4">

<div class="flex justify-between mb-1">

<span class="font-medium text-gray-800">Database Management Systems</span>

<span class="text-sm font-medium text-gray-700">72%</span>

</div>

<div class="w-full bg-gray-200 rounded-full h-2.5">

<div class="bg-indigo-500 h-2.5 rounded-full" style="width: 72%"></div>

</div>

</div>

<!-- More courses... -->

</div>

</div>

</section>

## 3. Course Management System

Let's implement the course catalog with filtering options:

<section id="courses" class="mb-12">

<div class="flex justify-between items-center mb-6">

<h2 class="text-2xl font-bold text-gray-800">Course Catalog</h2>

<!-- Filters -->

<div class="flex space-x-2">

<select class="bg-white border border-gray-300 text-gray-700 rounded-md px-3 py-2 text-sm">

<option value="all">All Categories</option>

<option value="programming">Programming</option>

<option value="algorithms">Algorithms</option>

<option value="database">Database Systems</option>

<option value="networking">Computer Networks</option>

</select>

<select class="bg-white border border-gray-300 text-gray-700 rounded-md px-3 py-2 text-sm">

<option value="all">All Levels</option>

<option value="beginner">Beginner</option>

<option value="intermediate">Intermediate</option>

<option value="advanced">Advanced</option>

</select>

</div>

</div>

<!-- Course Grid -->

<div class="grid grid-cols-1 md:grid-cols-2 lg:grid-cols-3 gap-6">

<!-- Course Card 1 -->

<div class="bg-white rounded-lg shadow overflow-hidden transition-transform hover:transform hover:scale-105">

<div class="relative">

<img src="course1.jpg" alt="Data Structures" class="w-full h-48 object-cover">

<span class="absolute top-2 right-2 bg-green-500 text-white text-xs font-bold px-2 py-1 rounded">

Beginner

</span>

</div>

<div class="p-5">

<div class="flex justify-between items-center mb-2">

<h3 class="text-lg font-bold text-gray-800">Data Structures</h3>

<div class="flex items-center space-x-1">

<i class="fas fa-star text-yellow-400"></i>

<span class="text-sm font-medium">4.8</span>

</div>

</div>

<p class="text-gray-600 text-sm mb-4">Master fundamental data structures with practical exercises</p>

<div class="flex justify-between text-sm text-gray-500 mb-4">

<span><i class="far fa-clock mr-1"></i> 24 hours</span>

<span><i class="far fa-list-alt mr-1"></i> 8 modules</span>

</div>

<!-- Progress Bar (For enrolled courses) -->

<div class="mb-4">

<div class="flex justify-between mb-1 text-xs">

<span>Progress</span>

<span>85%</span>

</div>

<div class="w-full bg-gray-200 rounded-full h-1.5">

<div class="bg-green-500 h-1.5 rounded-full" style="width: 85%"></div>

</div>

</div>

<a href="#continue-course" class="block text-center bg-indigo-600 text-white py-2 px-4 rounded hover:bg-indigo-700 transition-colors">

Continue Learning

</a>

</div>

</div>

<!-- Course Card 2 (Not enrolled) -->

<div class="bg-white rounded-lg shadow overflow-hidden transition-transform hover:transform hover:scale-105">

<div class="relative">

<img src="course2.jpg" alt="Computer Networks" class="w-full h-48 object-cover">

<span class="absolute top-2 right-2 bg-yellow-500 text-white text-xs font-bold px-2 py-1 rounded">

Intermediate

</span>

</div>

<div class="p-5">

<div class="flex justify-between items-center mb-2">

<h3 class="text-lg font-bold text-gray-800">Computer Networks</h3>

<div class="flex items-center space-x-1">

<i class="fas fa-star text-yellow-400"></i>

<span class="text-sm font-medium">4.6</span>

</div>

</div>

<p class="text-gray-600 text-sm mb-4">Learn networking concepts with practical examples</p>

<div class="flex justify-between text-sm text-gray-500 mb-4">

<span><i class="far fa-clock mr-1"></i> 32 hours</span>

<span><i class="far fa-list-alt mr-1"></i> 12 modules</span>

</div>

<a href="#enroll-course" class="block text-center bg-white border border-indigo-600 text-indigo-600 py-2 px-4 rounded hover:bg-indigo-50 transition-colors">

Enroll Now

</a>

</div>

</div>

<!-- More course cards... -->

</div>

</section>

## 4. Authentication System

The authentication system would be implemented with modals for login and signup:

<!-- Login Modal -->

<div id="loginModal" class="fixed inset-0 bg-black bg-opacity-50 flex items-center justify-center z-50 hidden">

<div class="bg-white rounded-lg shadow-lg max-w-md w-full mx-4">

<div class="flex justify-between items-center p-6 border-b">

<h3 class="text-xl font-bold text-gray-800">Welcome Back</h3>

<button class="text-gray-400 hover:text-gray-600" id="closeLoginModal">

<i class="fas fa-times"></i>

</button>

</div>

<form class="p-6">

<div class="mb-4">

<label for="email" class="block text-gray-700 text-sm font-medium mb-2">Email Address</label>

<input type="email" id="email" class="w-full px-3 py-2 border border-gray-300 rounded-md focus:outline-none focus:ring-2 focus:ring-indigo-500" required>

</div>

<div class="mb-6">

<label for="password" class="block text-gray-700 text-sm font-medium mb-2">Password</label>

<input type="password" id="password" class="w-full px-3 py-2 border border-gray-300 rounded-md focus:outline-none focus:ring-2 focus:ring-indigo-500" required>

<a href="#forgot-password" class="text-xs text-indigo-600 hover:text-indigo-800 mt-1 inline-block">

Forgot your password?

</a>

</div>

<button type="submit" class="w-full bg-indigo-600 text-white py-2 px-4 rounded-md hover:bg-indigo-700 focus:outline-none focus:ring-2 focus:ring-indigo-500 focus:ring-offset-2">

Sign In

</button>

<div class="mt-4 text-center text-sm">

<span class="text-gray-600">Don't have an account?</span>

<a href="#" id="showSignupModal" class="text-indigo-600 hover:text-indigo-800 ml-1">Sign up</a>

</div>

</form>

</div>

</div>

## 5. JavaScript for Interactive Elements

Let's implement the JavaScript for toggling the user dropdown and modals:

// app.js

// DOM Elements

const userDropdown = document.getElementById('userDropdown');

const dropdownMenu = document.getElementById('dropdownMenu');

const loginModal = document.getElementById('loginModal');

const closeLoginModal = document.getElementById('closeLoginModal');

const showSignupModal = document.getElementById('showSignupModal');

// Toggle User Dropdown

userDropdown.addEventListener('click', function(e) {

e.stopPropagation();

dropdownMenu.classList.toggle('hidden');

});

// Close dropdown when clicking outside

document.addEventListener('click', function() {

if (!dropdownMenu.classList.contains('hidden')) {

dropdownMenu.classList.add('hidden');

}

});

// Modal Control Functions

function openModal(modalId) {

const modal = document.getElementById(modalId);

modal.classList.remove('hidden');

document.body.style.overflow = 'hidden'; // Prevent scrolling when modal is open

}

function closeModal(modalId) {

const modal = document.getElementById(modalId);

modal.classList.add('hidden');

document.body.style.overflow = 'auto'; // Re-enable scrolling

}

// Modal Event Listeners

closeLoginModal.addEventListener('click', function() {

closeModal('loginModal');

});

// Close modals when clicking outside

loginModal.addEventListener('click', function(e) {

if (e.target === loginModal) {

closeModal('loginModal');

}

});

// Switch between login and signup modals

showSignupModal.addEventListener('click', function(e) {

e.preventDefault();

closeModal('loginModal');

openModal('signupModal');

});

// Course Filter Functionality

const categoryFilter = document.querySelector('select[value="all"]');

const levelFilter = document.querySelector('select[value="all"]');

function filterCourses() {

const category = categoryFilter.value;

const level = levelFilter.value;

// Get all course cards

const courseCards = document.querySelectorAll('.course-card');

courseCards.forEach(card => {

const cardCategory = card.dataset.category;

const cardLevel = card.dataset.level;

// Check if the card matches both filters or if filter is set to "all"

const categoryMatch = category === 'all' || cardCategory === category;

const levelMatch = level === 'all' || cardLevel === level;

// Show or hide the card based on filter match

if (categoryMatch && levelMatch) {

card.style.display = 'block';

} else {

card.style.display = 'none';

}

});

}

// Add event listeners to filters

if (categoryFilter && levelFilter) {

categoryFilter.addEventListener('change', filterCourses);

levelFilter.addEventListener('change', filterCourses);

}

// AI Chatbot Functionality

const chatbotButton = document.getElementById('openChatbot');

const chatbotModal = document.getElementById('chatbotModal');

const chatMessages = document.getElementById('chatMessages');

const chatInput = document.getElementById('chatInput');

const sendMessage = document.getElementById('sendMessage');

function initializeChatbot() {

if (!chatbotButton || !chatbotModal) return;

chatbotButton.addEventListener('click', function() {

openModal('chatbotModal');

});

sendMessage.addEventListener('click', function() {

const message = chatInput.value.trim();

if (message) {

addMessage('user', message);

chatInput.value = '';

// Simulate AI response (in real implementation, this would call an API)

setTimeout(() => {

processMessage(message);

}, 1000);

}

});

// Allow sending message with Enter key

chatInput.addEventListener('keypress', function(e) {

if (e.key === 'Enter') {

sendMessage.click();

}

});

}

function addMessage(sender, text) {

const messageDiv = document.createElement('div');

messageDiv.classList.add('mb-4', 'flex');

if (sender === 'user') {

messageDiv.classList.add('justify-end');

messageDiv.innerHTML = `

<div class="bg-indigo-100 rounded-lg py-2 px-4 max-w-[70%]">

<p class="text-gray-800">${text}</p>

</div>

`;

} else {

messageDiv.classList.add('justify-start');

messageDiv.innerHTML = `

<div class="bg-white rounded-lg py-2 px-4 max-w-[70%] shadow">

<p class="text-gray-800">${text}</p>

</div>

`;

}

chatMessages.appendChild(messageDiv);

chatMessages.scrollTop = chatMessages.scrollHeight;

}

function processMessage(message) {

// Simple keyword-based responses (in a real implementation, this would use NLP/ML)

message = message.toLowerCase();

if (message.includes('hello') || message.includes('hi')) {

addMessage('bot', 'Hello! How can I help you with your studies today?');

} else if (message.includes('course')) {

addMessage('bot', 'We have many courses available. Are you looking for beginner, intermediate, or advanced level courses?');

} else if (message.includes('assignment')) {

addMessage('bot', 'You can find your pending assignments in the dashboard under "Tasks". Would you like me to show you how to submit them?');

} else if (message.includes('certificate')) {

addMessage('bot', 'Certificates are issued automatically upon successful completion of a course. You can view and download your certificates from the "Certificates" section.');

} else {

addMessage('bot', 'I\'m not sure I understand. Could you rephrase your question or choose from common topics like courses, assignments, progress, or certificates?');

}

}

// Initialize chatbot when DOM is loaded

document.addEventListener('DOMContentLoaded', function() {

initializeChatbot();

});

// Voice Input Functionality for Notes

const voiceButton = document.getElementById('voiceInput');

const notesField = document.getElementById('notes');

function initializeVoiceInput() {

if (!voiceButton || !notesField) return;

voiceButton.addEventListener('click', function() {

// Check if browser supports SpeechRecognition

if ('SpeechRecognition' in window || 'webkitSpeechRecognition' in window) {

const SpeechRecognition = window.SpeechRecognition || window.webkitSpeechRecognition;

const recognition = new SpeechRecognition();

recognition.lang = 'en-US';

recognition.interimResults = false;

recognition.onstart = function() {

voiceButton.classList.add('recording');

voiceButton.innerHTML = '<i class="fas fa-microphone text-red-500"></i> Recording...';

};

recognition.onresult = function(event) {

const transcript = event.results[0][0].transcript;

notesField.value += transcript + ' ';

};

recognition.onerror = function(event) {

console.error('Speech recognition error', event.error);

voiceButton.classList.remove('recording');

voiceButton.innerHTML = '<i class="fas fa-microphone"></i> Voice Input';

};

recognition.onend = function() {

voiceButton.classList.remove('recording');

voiceButton.innerHTML = '<i class="fas fa-microphone"></i> Voice Input';

};

recognition.start();

} else {

alert('Sorry, your browser does not support speech recognition.');

}

});

}

// Initialize voice input when DOM is loaded

document.addEventListener('DOMContentLoaded', function() {

initializeVoiceInput();

});

## 6. AI Chatbot Implementation

Let's create the chatbot interface:

<!-- AI Chatbot Button (Fixed at bottom right) -->

<button id="openChatbot" class="fixed bottom-6 right-6 bg-indigo-600 text-white rounded-full p-4 shadow-lg hover:bg-indigo-700 focus:outline-none">

<i class="fas fa-robot text-xl"></i>

</button>

<!-- Chatbot Modal -->

<div id="chatbotModal" class="fixed inset-0 bg-black bg-opacity-50 flex items-center justify-center z-50 hidden">

<div class="bg-gray-100 rounded-lg shadow-lg w-full max-w-md h-[500px] flex flex-col">

<div class="bg-indigo-600 text-white px-4 py-3 rounded-t-lg flex justify-between items-center">

<h3 class="font-medium">EduLight Assistant</h3>

<button id="closeChatbot" class="text-white">

<i class="fas fa-times"></i>

</button>

</div>

<div id="chatMessages" class="flex-1 overflow-y-auto p-4 space-y-4">

<!-- Bot welcome message -->

<div class="flex justify-start mb-4">

<div class="bg-white rounded-lg py-2 px-4 max-w-[70%] shadow">

<p class="text-gray-800">Hi there! I'm your learning assistant. How can I help you today?</p>

</div>

</div>

<!-- Messages will be added here dynamically -->

</div>

<div class="border-t p-4 bg-white rounded-b-lg">

<div class="flex space-x-2">

<input type="text" id="chatInput" class="flex-1 border border-gray-300 rounded-l-lg px-4 py-2 focus:outline-none focus:ring-2 focus:ring-indigo-500" placeholder="Type your question...">

<button id="sendMessage" class="bg-indigo-600 text-white px-4 py-2 rounded-r-lg hover:bg-indigo-700">

<i class="fas fa-paper-plane"></i>

</button>

</div>

</div>

</div>

</div>

## 7. Course Content Page

Let's implement a course content page with video player and interactive quizzes:

<div class="container mx-auto px-4 pt-24 pb-12">

<!-- Course Header -->

<div class="bg-white rounded-lg shadow-md p-6 mb-6">

<div class="flex items-center justify-between mb-4">

<h1 class="text-2xl font-bold text-gray-800">Data Structures & Algorithms</h1>

<span class="bg-green-100 text-green-800 text-xs font-semibold px-3 py-1 rounded-full">

85% Complete

</span>

</div>

<div class="flex flex-wrap gap-4 text-sm text-gray-600 mb-4">

<span><i class="fas fa-user mr-1"></i> Instructor: Dr. Patel</span>

<span><i class="fas fa-calendar mr-1"></i> Updated: March 2025</span>

<span><i class="fas fa-clock mr-1"></i> 24 hours total</span>

<span><i class="fas fa-signal mr-1"></i> Beginner</span>

</div>

<div class="w-full bg-gray-200 rounded-full h-2.5 mb-2">

<div class="bg-green-500 h-2.5 rounded-full" style="width: 85%"></div>

</div>

</div>

<!-- Course Content -->

<div class="grid grid-cols-1 lg:grid-cols-3 gap-6">

<!-- Video Player and Content Area -->

<div class="lg:col-span-2">

<!-- Video Player -->

<div class="bg-black rounded-lg shadow-md mb-6 aspect-video">

<video id="videoPlayer" class="w-full h-full rounded-lg" controls>

<source src="lecture1.mp4" type="video/mp4">

Your browser does not support the video tag.

</video>

</div>

<!-- Content Tabs -->

<div class="bg-white rounded-lg shadow-md mb-6">

<div class="border-b">

<nav class="flex" aria-label="Tabs">

<button class="px-4 py-3 text-indigo-600 border-b-2 border-indigo-600 font-medium">

Lecture Notes

</button>

<button class="px-4 py-3 text-gray-500 hover:text-gray-700 font-medium">

Quiz

</button>

<button class="px-4 py-3 text-gray-500 hover:text-gray-700 font-medium">

Resources

</button>

<button class="px-4 py-3 text-gray-500 hover:text-gray-700 font-medium">

Discussion

</button>

</nav>

</div>

<div class="p-6">

<!-- Lecture Notes Content -->

<div id="lectureNotes">

<h2 class="text-xl font-bold mb-4">Array Data Structures</h2>

<p class="mb-4">An array is a collection of items stored at contiguous memory locations. The idea is to store multiple items of the same type together.</p>

<h3 class="text-lg font-semibold mb-2">Key Characteristics:</h3>

<ul class="list-disc pl-6 mb-4 space-y-1">

<li>Fixed size (in most programming languages)</li>

<li>Homogeneous elements (same data type)</li>

<li>O(1) time complexity for

<!-- Continuing from the course content page -->

<li>O(1) time complexity for random access</li>

<li>Elements are identified by index</li>

</ul>

<h3 class="text-lg font-semibold mb-2">Common Operations:</h3>

<ul class="list-disc pl-6 mb-4 space-y-1">

<li>Insertion - O(n) in worst case</li>

<li>Deletion - O(n) in worst case</li>

<li>Searching - O(n) for linear search, O(log n) for binary search (if sorted)</li>

<li>Traversal - O(n)</li>

</ul>

<div class="bg-gray-100 p-4 rounded-lg mb-4">

<h3 class="text-lg font-semibold mb-2">Code Example:</h3>

<pre class="bg-gray-800 text-white p-4 rounded overflow-x-auto">

// Array implementation in C++

#include &lt;iostream&gt;

using namespace std;

int main() {

// Declaring an array

int arr[5] = {10, 20, 30, 40, 50};

// Accessing array elements

cout << "Element at index 2: " << arr[2] << endl;

// Modifying array elements

arr[2] = 35;

cout << "Modified element at index 2: " << arr[2] << endl;

// Array traversal

cout << "Array elements: ";

for(int i = 0; i < 5; i++) {

cout << arr[i] << " ";

}

return 0;

}

</pre>

</div>

<div class="flex space-x-2 mb-4">

<button id="voiceInput" class="flex items-center space-x-1 bg-gray-100 hover:bg-gray-200 text-gray-700 px-3 py-2 rounded">

<i class="fas fa-microphone"></i>

<span>Voice Input</span>

</button>

<button class="flex items-center space-x-1 bg-gray-100 hover:bg-gray-200 text-gray-700 px-3 py-2 rounded">

<i class="fas fa-download"></i>

<span>Download Notes</span>

</button>

</div>

<div class="border rounded-lg p-4">

<h3 class="text-lg font-semibold mb-2">Your Notes</h3>

<textarea id="notes" class="w-full border rounded-md p-2 h-24" placeholder="Add your notes here..."></textarea>

<div class="flex justify-end mt-2">

<button class="bg-indigo-600 text-white px-4 py-2 rounded hover:bg-indigo-700">

Save Notes

</button>

</div>

</div>

</div>

<!-- Quiz Content (Hidden by default) -->

<div id="quizContent" class="hidden">

<h2 class="text-xl font-bold mb-4">Module Quiz: Arrays</h2>

<div class="space-y-6">

<!-- Question 1 -->

<div class="border rounded-lg p-4">

<p class="font-medium mb-3">1. What is the time complexity for accessing an element in an array?</p>

<div class="space-y-2">

<div class="flex items-center">

<input type="radio" id="q1a" name="q1" class="mr-2">

<label for="q1a">O(1)</label>

</div>

<div class="flex items-center">

<input type="radio" id="q1b" name="q1" class="mr-2">

<label for="q1b">O(log n)</label>

</div>

<div class="flex items-center">

<input type="radio" id="q1c" name="q1" class="mr-2">

<label for="q1c">O(n)</label>

</div>

<div class="flex items-center">

<input type="radio" id="q1d" name="q1" class="mr-2">

<label for="q1d">O(n²)</label>

</div>

</div>

</div>

<!-- Question 2 -->

<div class="border rounded-lg p-4">

<p class="font-medium mb-3">2. What is the worst-case time complexity for inserting an element at the beginning of an array?</p>

<div class="space-y-2">

<div class="flex items-center">

<input type="radio" id="q2a" name="q2" class="mr-2">

<label for="q2a">O(1)</label>

</div>

<div class="flex items-center">

<input type="radio" id="q2b" name="q2" class="mr-2">

<label for="q2b">O(log n)</label>

</div>

<div class="flex items-center">

<input type="radio" id="q2c" name="q2" class="mr-2">

<label for="q2c">O(n)</label>

</div>

<div class="flex items-center">

<input type="radio" id="q2d" name="q2" class="mr-2">

<label for="q2d">O(n²)</label>

</div>

</div>

</div>

<!-- Question 3 -->

<div class="border rounded-lg p-4">

<p class="font-medium mb-3">3. Consider an array A with n elements. What is the index of the last element?</p>

<div class="space-y-2">

<div class="flex items-center">

<input type="radio" id="q3a" name="q3" class="mr-2">

<label for="q3a">n</label>

</div>

<div class="flex items-center">

<input type="radio" id="q3b" name="q3" class="mr-2">

<label for="q3b">n-1</label>

</div>

<div class="flex items-center">

<input type="radio" id="q3c" name="q3" class="mr-2">

<label for="q3c">n+1</label>

</div>

<div class="flex items-center">

<input type="radio" id="q3d" name="q3" class="mr-2">

<label for="q3d">0</label>

</div>

</div>

</div>

<div class="flex justify-end">

<button class="bg-indigo-600 text-white px-6 py-2 rounded hover:bg-indigo-700">

Submit Quiz

</button>

</div>

</div>

</div>

</div>

</div>

</div>

<!-- Course Sidebar -->

<div class="lg:col-span-1">

<!-- Module Navigator -->

<div class="bg-white rounded-lg shadow-md mb-6">

<div class="p-4 border-b">

<h3 class="font-semibold">Course Content</h3>

</div>

<div class="p-4">

<div class="space-y-4">

<!-- Module 1 -->

<div>

<div class="flex items-center justify-between cursor-pointer">

<span class="font-medium">Module 1: Introduction to Data Structures</span>

<i class="fas fa-chevron-down text-gray-500"></i>

</div>

<div class="ml-4 mt-2 space-y-2">

<!-- Lecture -->

<div class="flex items-center justify-between py-2 pl-2 border-l-2 border-indigo-600">

<div class="flex items-center">

<i class="fas fa-play-circle text-indigo-600 mr-2"></i>

<span class="text-sm">1.1 Array Data Structures</span>

</div>

<span class="text-xs bg-green-100 text-green-800 px-2 py-1 rounded">Completed</span>

</div>

<!-- Quiz -->

<div class="flex items-center justify-between py-2 pl-2 border-l-2 border-gray-200">

<div class="flex items-center">

<i class="fas fa-question-circle text-gray-500 mr-2"></i>

<span class="text-sm">Quiz: Arrays</span>

</div>

<span class="text-xs bg-yellow-100 text-yellow-800 px-2 py-1 rounded">Pending</span>

</div>

<!-- Assignment -->

<div class="flex items-center justify-between py-2 pl-2 border-l-2 border-gray-200">

<div class="flex items-center">

<i class="fas fa-file-alt text-gray-500 mr-2"></i>

<span class="text-sm">Assignment: Array Implementation</span>

</div>

<span class="text-xs bg-gray-100 text-gray-800 px-2 py-1 rounded">Not Started</span>

</div>

</div>

</div>

<!-- Module 2 -->

<div>

<div class="flex items-center justify-between cursor-pointer">

<span class="font-medium">Module 2: Linked Lists</span>

<i class="fas fa-chevron-right text-gray-500"></i>

</div>

</div>

<!-- Module 3 -->

<div>

<div class="flex items-center justify-between cursor-pointer">

<span class="font-medium">Module 3: Stacks & Queues</span>

<i class="fas fa-chevron-right text-gray-500"></i>

</div>

</div>

<!-- Module 4 -->

<div>

<div class="flex items-center justify-between cursor-pointer">

<span class="font-medium">Module 4: Trees & Graphs</span>

<i class="fas fa-chevron-right text-gray-500"></i>

</div>

</div>

</div>

</div>

</div>

<!-- Instructor Info -->

<div class="bg-white rounded-lg shadow-md mb-6">

<div class="p-4 border-b">

<h3 class="font-semibold">Instructor</h3>

</div>

<div class="p-4">

<div class="flex items-center space-x-3 mb-3">

<img src="instructor.jpg" alt="Dr. Patel" class="h-12 w-12 rounded-full object-cover">

<div>

<h4 class="font-medium">Dr. Anita Patel</h4>

<p class="text-sm text-gray-600">PhD, Computer Science</p>

</div>

</div>

<p class="text-sm text-gray-700 mb-3">

Dr. Patel has over 10 years of experience teaching data structures and algorithms at premier technical universities.

</p>

<a href="#contact-instructor" class="text-indigo-600 text-sm hover:text-indigo-800">

<i class="fas fa-envelope mr-1"></i> Contact Instructor

</a>

</div>

</div>

</div>

</div>

</div>

<!-- Certificate Generation System -->

<div class="container mx-auto px-4 py-12" id="certificates">

<h2 class="text-2xl font-bold text-gray-800 mb-6">Your Certificates</h2>

<div class="grid grid-cols-1 md:grid-cols-2 lg:grid-cols-3 gap-6">

<!-- Certificate Card 1 -->

<div class="bg-white rounded-lg shadow-md overflow-hidden">

<div class="relative">

<img src="certificate1.jpg" alt="Certificate" class="w-full h-40 object-cover">

<div class="absolute top-0 left-0 w-full h-full bg-black bg-opacity-50 flex items-center justify-center opacity-0 hover:opacity-100 transition-opacity">

<button class="bg-white text-gray-800 px-4 py-2 rounded-md hover:bg-gray-100">

<i class="fas fa-eye mr-1"></i> View

</button>

</div>

</div>

<div class="p-4">

<h3 class="font-bold text-gray-800 mb-1">Introduction to Web Development</h3>

<p class="text-gray-600 text-sm mb-3">Completed on: March 15, 2025</p>

<div class="flex justify-between text-sm">

<a href="#download" class="text-indigo-600 hover:text-indigo-800">

<i class="fas fa-download mr-1"></i> Download

</a>

<a href="#share" class="text-indigo-600 hover:text-indigo-800">

<i class="fas fa-share-alt mr-1"></i> Share

</a>

</div>

</div>

</div>

<!-- Certificate Card 2 -->

<div class="bg-white rounded-lg shadow-md overflow-hidden">

<div class="relative">

<img src="certificate2.jpg" alt="Certificate" class="w-full h-40 object-cover">

<div class="absolute top-0 left-0 w-full h-full bg-black bg-opacity-50 flex items-center justify-center opacity-0 hover:opacity-100 transition-opacity">

<button class="bg-white text-gray-800 px-4 py-2 rounded-md hover:bg-gray-100">

<i class="fas fa-eye mr-1"></i> View

</button>

</div>

</div>

<div class="p-4">

<h3 class="font-bold text-gray-800 mb-1">Database Systems</h3>

<p class="text-gray-600 text-sm mb-3">Completed on: February 2, 2025</p>

<div class="flex justify-between text-sm">

<a href="#download" class="text-indigo-600 hover:text-indigo-800">

<i class="fas fa-download mr-1"></i> Download

</a>

<a href="#share" class="text-indigo-600 hover:text-indigo-800">

<i class="fas fa-share-alt mr-1"></i> Share

</a>

</div>

</div>

</div>

</div>

</div>

<!-- JavaScript for Interactive Features -->

<script>

// Tab Switching Functionality

document.addEventListener('DOMContentLoaded', function() {

const tabButtons = document.querySelectorAll('nav button');

const tabContents = [

document.getElementById('lectureNotes'),

document.getElementById('quizContent')

// Add other content elements as needed

];

tabButtons.forEach((button, index) => {

button.addEventListener('click', () => {

// Reset all buttons

tabButtons.forEach(btn => {

btn.classList.remove('text-indigo-600', 'border-b-2', 'border-indigo-600');

btn.classList.add('text-gray-500');

});

// Highlight active button

button.classList.remove('text-gray-500');

button.classList.add('text-indigo-600', 'border-b-2', 'border-indigo-600');

// Show corresponding content

tabContents.forEach((content, i) => {

if (i === index) {

content.classList.remove('hidden');

} else {

content.classList.add('hidden');

}

});

});

});

// Module Dropdown Toggle

const moduleHeaders = document.querySelectorAll('.cursor-pointer');

moduleHeaders.forEach(header => {

header.addEventListener('click', () => {

const icon = header.querySelector('i');

const content = header.nextElementSibling;

if (content.style.display === 'none' || content.style.display === '') {

content.style.display = 'block';

icon.classList.remove('fa-chevron-right');

icon.classList.add('fa-chevron-down');

} else {

content.style.display = 'none';

icon.classList.remove('fa-chevron-down');

icon.classList.add('fa-chevron-right');

}

});

});

// Voice Input Functionality

const voiceButton = document.getElementById('voiceInput');

const notesField = document.getElementById('notes');

if (voiceButton && notesField) {

voiceButton.addEventListener('click', function() {

// Check if browser supports SpeechRecognition

if ('SpeechRecognition' in window || 'webkitSpeechRecognition' in window) {

const SpeechRecognition = window.SpeechRecognition || window.webkitSpeechRecognition;

const recognition = new SpeechRecognition();

recognition.lang = 'en-US';

recognition.interimResults = false;

recognition.onstart = function() {

voiceButton.classList.add('bg-red-100');

voiceButton.innerHTML = '<i class="fas fa-microphone-slash"></i><span>Recording...</span>';

};

recognition.onresult = function(event) {

const transcript = event.results[0][0].transcript;

notesField.value += transcript + ' ';

};

recognition.onerror = function(event) {

console.error('Speech recognition error', event.error);

voiceButton.classList.remove('bg-red-100');

voiceButton.innerHTML = '<i class="fas fa-microphone"></i><span>Voice Input</span>';

};

recognition.onend = function() {

voiceButton.classList.remove('bg-red-100');

voiceButton.innerHTML = '<i class="fas fa-microphone"></i><span>Voice Input</span>';

};

recognition.start();

} else {

alert('Sorry, your browser does not support speech recognition.');

}

});

}

});

</script>

<!-- JavaScript for Certificate Modal -->

<script>

document.addEventListener('DOMContentLoaded', function() {

const viewButtons = document.querySelectorAll('.rounded-md');

viewButtons.forEach(button => {

button.addEventListener('click', function() {

// Create modal dynamically

const modal = document.createElement('div');

modal.classList.add('fixed', 'inset-0', 'bg-black', 'bg-opacity-50', 'flex', 'items-center', 'justify-center', 'z-50');

modal.innerHTML = `

<div class="bg-white rounded-lg shadow-lg max-w-3xl w-full mx-4 p-6">

<div class="flex justify-between items-center mb-6">

<h3 class="text-xl font-bold text-gray-800">Certificate of Completion</h3>

<button class="text-gray-400 hover:text-gray-600 close-modal">

<i class="fas fa-times"></i>

</button>

</div>

<div class="border p-6 mb-6">

<div class="text-center">

<h2 class="text-3xl font-bold text-indigo-600 mb-2">EduLightHub</h2>

<p class="text-xl mb-6">Certificate of Completion</p>

<p class="text-lg mb-2">This certifies that</p>

<p class="text-2xl font-bold mb-2">Rahul Singh</p>

<p class="text-lg mb-6">has successfully completed the course</p>

<p class="text-2xl font-bold text-indigo-600 mb-6">Introduction to Web Development</p>

<div class="flex justify-between items-center">

<div class="text-left">

<p class="font-medium">Date</p>

<p>March 15, 2025</p>

</div>

<div class="text-right">

<p class="font-medium">Instructor</p>

<p>Dr. Anita Patel</p>

</div>

</div>

</div>

</div>

<div class="flex justify-end space-x-2">

<button class="bg-white border border-indigo-600 text-indigo-600 px-4 py-2 rounded hover:bg-indigo-50 close-modal">

Close

</button>

<button class="bg-indigo-600 text-white px-4 py-2 rounded hover:bg-indigo-700">

<i class="fas fa-download mr-1"></i> Download

</button>

</div>

</div>

`;

document.body.appendChild(modal);

document.body.style.overflow = 'hidden';

// Add event listener to close the modal

modal.querySelectorAll('.close-modal').forEach(closeBtn => {

closeBtn.addEventListener('click', function() {

document.body.removeChild(modal);

document.body.style.overflow = 'auto';

});

});

// Close modal when clicking outside

modal.addEventListener('click', function(e) {

if (e.target === modal) {

document.body.removeChild(modal);

document.body.style.overflow = 'auto';

}

});

});

});

});

</script>

<!-- AI Chatbot JavaScript -->

<script>

document.addEventListener('DOMContentLoaded', function() {

const chatbotButton = document.getElementById('openChatbot');

const chatbotModal = document.getElementById('chatbotModal');

const closeChatbot = document.getElementById('closeChatbot');

const chatMessages = document.getElementById('chatMessages');

const chatInput = document.getElementById('chatInput');

const sendMessage = document.getElementById('sendMessage');

if (!chatbotButton || !chatbotModal) return;

// Open chatbot modal

chatbotButton.addEventListener('click', function() {

chatbotModal.classList.remove('hidden');

document.body.style.overflow = 'hidden';

});

// Close chatbot modal

closeChatbot.addEventListener('click', function() {

chatbotModal.classList.add('hidden');

document.body.style.overflow = 'auto';

});

// Close modal when clicking outside

chatbotModal.addEventListener('click', function(e) {

if (e.target === chatbotModal) {

chatbotModal.classList.add('hidden');

document.body.style.overflow = 'auto';

}

});

// Send message function

function sendUserMessage() {

const message = chatInput.value.trim();

if (message) {

addMessage('user', message);

chatInput.value = '';

// Simulate AI response (in real implementation, this would call an API)

setTimeout(() => {

processMessage(message);

}, 1000);

}

}

// Send message on button click

sendMessage.addEventListener('click', sendUserMessage);

// Send message on Enter key

chatInput.addEventListener('keypress', function(e) {

if (e.key === 'Enter') {

sendUserMessage();

}

});

// Add message to chat

function addMessage(sender, text) {

const messageDiv = document.createElement('div');

messageDiv.classList.add('mb-4', 'flex');

if (sender === 'user') {

messageDiv.classList.add('justify-end');

messageDiv.innerHTML = `

<div class="bg-indigo-100 rounded-lg py-2 px-4 max-w-[70%]">

<p class="text-gray-800">${text}</p>

</div>

`;

} else {

messageDiv.classList.add('justify-start');

messageDiv.innerHTML = `

<div class="bg-white rounded-lg py-2 px-4 max-w-[70%] shadow">

<p class="text-gray-800">${text}</p>

</div>

`;

}

chatMessages.appendChild(messageDiv);

chatMessages.scrollTop = chatMessages.scrollHeight;

}

// Process user message and generate response

function processMessage(message) {

message = message.toLowerCase();

if (message.includes('hello') || message.includes('hi')) {

addMessage('bot', 'Hello! How can I help you with your studies today?');

} else if (message.includes('array') || message.includes('data structure')) {

addMessage('bot', 'Arrays are a fundamental data structure that store elements of the same type in contiguous memory locations. Would you like me to explain more about arrays or another data structure?');

} else if (message.includes('quiz') || message.includes('test')) {

addMessage('bot', 'You have a quiz on Arrays in Module 1. The quiz contains 3 questions and you can access it from the Course Content sidebar or by clicking the Quiz tab above the lecture notes.');

} else if (message.includes('assignment')) {

addMessage('bot', 'Your current assignment is on Array Implementation. The deadline is May 5, 2025. Would you like me to help you understand the requirements?');

} else if (message.includes('certificate')) {

addMessage('bot', 'You\'ll receive a certificate once you complete all modules and pass the final assessment. You currently have 2 certificates that you can view in the Certificates section.');

} else {

addMessage('bot', 'I\'m not sure I understand. Could you rephrase your question? I can help with course content, assignments, quizzes, and general questions about the platform.');

}

}

});

</script>

<!-- File Upload System for Content Sharing -->

<div class="container mx-auto px-4 py-12" id="resources">

<h2 class="text-2xl font-bold text-gray-800 mb-6">Resource Sharing</h2>

<div class="grid grid-cols-1 lg:grid-cols-3 gap-6">

<!-- Upload Section -->

<div class="lg:col-span-1">

<div class="bg-white rounded-lg shadow-md p-6">

<h3 class="text-lg font-semibold mb-4">Share Resources</h3>

<form>

<div class="mb-4">

<label for="resourceTitle" class="block text-sm font-medium text-gray-700 mb-1">Title</label>

<input type="text" id="resourceTitle" class="w-full border border-gray-300 rounded-md px-3 py-2 focus:outline-none focus:ring-2 focus:ring-indigo-500">

</div>

<div class="mb-4">

<label for="resourceType" class="block text-sm font-medium text-gray-700 mb-1">Resource Type</label>

<select id="resourceType" class="w-full border border-gray-300 rounded-md px-3 py-2 focus:outline-none focus:ring-2 focus:ring-indigo-500">

<option value="notes">Lecture Notes</option>

<option value="assignment">Assignment</option>

<option value="project">Project</option>

<option value="reference">Reference Material</option>

</select>

</div>

<div class="mb-4">

<label for="resourceCategory" class="block text-sm font-medium text-gray-700 mb-1">Category</label>

<select id="resourceCategory" class="w-full border border-gray-300 rounded-md px-3 py-2 focus:outline-none focus:ring-2 focus:ring-indigo-500">

<option value="ds">Data Structures</option>

<option value="algo">Algorithms</option>

<option value="db">Database Systems</option>

<option value="web">Web Development</option>

<option value="ml">Machine Learning</option>

</select>

</div>

<div class="mb-4">

<label for="resourceDescription" class="block text-sm font-medium text-gray-700 mb-1">Description</label>

<textarea id="resourceDescription" rows="3" class="w-full border border-gray-300 rounded-md px-3 py-2 focus:outline-none focus:ring-2 focus:ring-indigo-500"></textarea>

</div>

<div class="mb-6">

<label class="block text-sm font-medium text-gray-700 mb-1">File Upload</label>

<div class="border-2 border-dashed border