Τίτλος Εργασίας: Van gogh Web app

Συνοπτική Περιγραφή:  
Η εφαρμογή "Van Gogh Web App" είναι ένας διαδικτυακός τόπος αφιερωμένος στον Vincent Van Gogh. Περιλαμβάνει βιογραφικές πληροφορίες, πίνακες, εκθέσεις, συνδέσμους και εργαλεία διαχείρισης για τον διαχειριστή. Η εφαρμογή είναι δυναμική και ανταποκρίνεται στις επιλογές του χρήστη, προσαρμόζοντας το περιεχόμενό της ανάλογα με την επιλογή από το κεντρικό μενού.

### 1. Διάταξη της Ιστοσελίδας

Η ιστοσελίδα χωρίζεται σε τρία κύρια μέρη:

* **Κεντρικό Μενού (header και nav)**
* **Κύριο Περιεχόμενο (main)**

### 2. Δυναμική Εμφάνιση και Απόκρυψη Τμημάτων

Για να επιτύχουμε τη δυναμική εμφάνιση και απόκρυψη τμημάτων, χρησιμοποιούμε JavaScript για την αλλαγή των κλάσεων CSS. Συγκεκριμένα, χρησιμοποιούμε τη μέθοδο add και remove για να προσθέσουμε ή να αφαιρέσουμε την κλάση d-none, η οποία κρύβει τα στοιχεία.

1. **Μορφοποίηση και Διάταξη Ιστοσελίδας**
2. **Δυναμική Εμφάνιση και Απόκρυψη Τμημάτων**
3. **Web Services**
4. **Βάση Δεδομένων ή Αρχεία Δεδομένων**

### 1. Μορφοποίηση και Διάταξη Ιστοσελίδας

Η μορφοποίηση και η διάταξη της ιστοσελίδας υλοποιήθηκαν με τη χρήση HTML για τη δομή και CSS για τη μορφοποίηση. Χρησιμοποιήθηκε επίσης το Bootstrap framework για να επιτευχθεί μια πιο μοντέρνα και ανταποκρινόμενη διάταξη.

#### HTML Δομή:

Το HTML αρχείο περιλαμβάνει τα εξής βασικά τμήματα:

* **Header:** Περιέχει το λογότυπο και τον τίτλο της ιστοσελίδας.
* **Κεντρικό Μενού (nav):** Τοποθετημένο στην κορυφή της σελίδας για την πλοήγηση μεταξύ των κύριων ενοτήτων.
* **Πλευρικό Μενού (aside):** Τοποθετημένο στα αριστερά, περιέχει υπομενού που εμφανίζονται ανάλογα με την επιλογή στο κεντρικό μενού.
* **Κύριο Περιεχόμενο (main):** Το τμήμα όπου φορτώνεται δυναμικά το περιεχόμενο της κάθε ενότητας.
* **Footer:** Περιέχει πληροφορίες πνευματικών δικαιωμάτων.

Παράδειγμα HTML:

HTML

<!DOCTYPE html>  
<html lang="el">  
<head>  
 <meta charset="UTF-8">  
 <meta name="viewport" content="width=device-width, initial-scale=1.0">  
 <title>Vincent Van Gogh Biography</title>  
 <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">  
 <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.3/css/all.min.css">  
 <link rel="stylesheet" href="css/styles.css">  
</head>  
<body>  
 <header class="bg-dark text-white text-center py-3">  
 <div class="container">  
 <img src="images/vangogh.jpg" alt="Vincent Van Gogh" class="rounded-circle" width="50" height="50">  
 <h1 class="d-inline-block ml-3">Vincent Van Gogh</h1>  
 </div>  
 </header>  
   
 <!-- Κεντρικό Μενού -->  
 <nav class="bg-dark text-white p-3">  
 <ul class="nav">  
 <li class="nav-item"><a class="nav-link text-white" href="#biography" onclick="showSubMenu('biography')">Βιογραφία</a></li>  
 <li class="nav-item"><a class="nav-link text-white" href="#tables" onclick="showSubMenu('tables')">Πίνακες</a></li>  
 <li class="nav-item"><a class="nav-link text-white" href="#reports" onclick="showSubMenu('reports')">Εκθέσεις</a></li>  
 <li class="nav-item"><a class="nav-link text-white" href="#links" onclick="showSubMenu('links')">Σύνδεσμοι</a></li>  
 <li class="nav-item"><a class="nav-link text-white" href="#management" onclick="showSubMenu('management')">Διαχείριση</a></li>  
 </ul>  
 </nav>  
  
 <div class="container-fluid">  
 <div class="row">  
 <!-- Πλευρικό Μενού -->  
 <aside id="sidebar" class="col-md-3 bg-light p-3">  
 <div id="biography-submenu" class="submenu">  
 <h4>Βιογραφία</h4>  
 <p>Πληροφορίες για τη ζωή του Vincent Van Gogh.</p>  
 </div>  
 <div id="tables-submenu" class="submenu d-none">  
 <h4>Πίνακες</h4>  
 <ul>  
 <li><a href="#" onclick="showTable('Paintings')">Κατηγορία 1</a></li>  
 <li><a href="#" onclick="showTable('Drawings')">Κατηγορία 2</a></li>  
 </ul>  
 </div>  
 <div id="reports-submenu" class="submenu d-none">  
 <h4>Εκθέσεις</h4>  
 <p>Πληροφορίες για τις εκθέσεις.</p>  
 </div>  
 <div id="links-submenu" class="submenu d-none">  
 <h4>Σύνδεσμοι</h4>  
 <p>Χρήσιμοι σύνδεσμοι.</p>  
 </div>  
 <div id="management-submenu" class="submenu d-none">  
 <h4>Διαχείριση</h4>  
 <p>Εργαλεία διαχείρισης.</p>  
 </div>  
 </aside>  
  
 <!-- Κύριο Περιεχόμενο -->  
 <main id="main-content" class="col-md-9 ml-sm-auto col-lg-9 px-md-4 py-3">  
 <!-- Το κύριο περιεχόμενο θα φορτωθεί δυναμικά εδώ -->  
 </main>  
 </div>  
 </div>  
   
 <footer class="bg-dark text-white text-center py-3">  
 <p>&copy; 2025 Vincent Van Gogh Web Application</p>  
 </footer>  
   
 <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>  
 <script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.5.4/dist/umd/popper.min.js"></script>  
 <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>  
 <script src="js/app.js"></script>  
 <script>  
 function showSubMenu(menu) {  
 const submenus = document.querySelectorAll('.submenu');  
 submenus.forEach(submenu => submenu.classList.add('d-none'));  
 document.gopetElementById(`${menu}-submenu`).classList.remove('d-none');  
 }  
 </script>  
</body>  
</html>

#### CSS Μορφοποίηση:

Χρησιμοποιούμε το Bootstrap για να πετύχουμε μια ανταποκρινόμενη και σύγχρονη διάταξη. Προσθέσαμε επίσης προσαρμοσμένα στυλ για να ταιριάζουν με το θέμα της ιστοσελίδας.

CSS

body {  
 font-family: 'Arial', sans-serif;  
}  
  
header {  
 float: none;  
 width: 100%;  
}  
  
nav {  
 float: none;  
 width: 100%;  
}  
  
#sidebar {  
 float: left;  
 width: 25%;  
 height: 100%;  
 background-color: #f8f9fa;  
}  
  
#main-content {  
 float: left;  
 width: 75%;  
 padding: 20px;  
}  
  
footer {  
 clear: both;  
 width: 100%;  
}  
  
.submenu {  
 display: none;  
}  
  
.d-none {  
 display: none;  
}

### 2. Δυναμική Εμφάνιση και Απόκρυψη Τμημάτων

Η JavaScript χρησιμοποιείται για τη δυναμική εμφάνιση και απόκρυψη τμημάτων της σελίδας ανάλογα με την επιλογή του χρήστη στο κεντρικό μενού.

#### JavaScript Λειτουργίες:

Το αρχείο app.js περιέχει τις λειτουργίες για τη φόρτωση του περιεχομένου, τη διαχείριση της σύνδεσης του διαχειριστή, και τη διαχείριση αναφορών και συνδέσμων.

Αρχικοποιούμε μια μεταβλητή isAdminLoggedIn για να ελέγξουμε αν ο διαχειριστής είναι συνδεδεμένος και, όταν φορτωθεί η σελίδα, καλούμε τη λειτουργία loadContent για να φορτώσουμε το περιεχόμενο της βιογραφίας. Η λειτουργία loadContent φορτώνει δυναμικά το περιεχόμενο της σελίδας από ένα API. Ανάλογα με το τμήμα (section) που ζητείται, η λειτουργία διαμορφώνει το περιεχόμενο και το εμφανίζει στο main-content.

Αν ο διαχειριστής είναι συνδεδεμένος, εμφανίζει επιλογές διαχείρισης και ένα κουμπί αποσύνδεσης. Αν όχι, φορτώνει τη φόρμα εισόδου. Η λειτουργία login επιτρέπει στον διαχειριστή να συνδεθεί, ενώ η λειτουργία logout επιτρέπει την αποσύνδεση. Η εφαρμογή επιτρέπει στον διαχειριστή να προσθέτει και να διαγράφει αναφορές και συνδέσμους μέσω API κλήσεων.

JavaScript

let isAdminLoggedIn = false;  
  
document.addEventListener('DOMContentLoaded', () => {  
 loadContent('biography');  
});  
  
function loadContent(section) {  
 if (section === 'management') {  
 loadManagementContent();  
 return;  
 }  
  
 fetch(`/api/${section}`)  
 .then(response => response.json())  
 .then(data => {  
 const mainContent = document.getElementById('main-content');  
   
 if (section === 'biography') {  
 mainContent.innerHTML = `<h2>${data.title}</h2><p>${data.content}</p>`;  
 } else if (section === 'tables') {  
 mainContent.innerHTML = '<h2>Tables</h2>' +  
 data.categories.map(category => `<h3>${category}</h3><div class="flex-container">` +  
 data.tables[category].map(item => `<div class="flex-item">${item}</div>`).join('') + '</div>').join('');  
 } else if (section === 'reports') {  
 mainContent.innerHTML = '<h2>Reports</h2>' +  
 data.map(report => `<div class="report"><h3>${report.title}</h3><p>${report.content}</p>${isAdminLoggedIn ? `<button class="btn btn-danger" onclick="deleteReport('${report.\_id}')">Delete</button>` : ''}</div>`).join('');  
 } else if (section === 'links') {  
 mainContent.innerHTML = '<h2>Links</h2>' +  
 data.map(link => `<div class="link"><a href="${link.url}" target="\_blank">${link.url}</a><p>Category: ${link.category}</p>${isAdminLoggedIn ? `<button class="btn btn-danger" onclick="deleteLink('${link.\_id}')">Delete</button>` : ''}</div>`).join('');  
 }  
 })  
 .catch(error => {  
 console.error('Error:', error);  
 const mainContent = document.getElementById('main-content');  
 mainContent.innerHTML = '<p>Error loading content. Please try again later.</p>';  
 });  
}  
  
function loadManagementContent() {  
 const mainContent = document.getElementById('main-content');  
  
 if (isAdminLoggedIn) {  
 mainContent.innerHTML = `  
 <h2>Management</h2>  
 <button id="logout-button" class="btn btn-primary">Logout</button>  
 <div id="admin-options">  
 <h3>Management Options</h3>  
 <ul class="list-unstyled">  
 <li><a href="#" onclick="showAdminSection('reports')">Manage Reports</a></li>  
 <li><a href="#" onclick="showAdminSection('links')">Manage Links</a></li>  
 </ul>  
 </div>  
 `;  
 document.getElementById('logout-button').addEventListener('click', logout);  
 } else {  
 mainContent.innerHTML = `  
 <h2>Management</h2>  
 <form id="login-form">  
 <div class="form-group">  
 <label for="username">Username:</label>  
 <input type="text" id="username" name="username" class="form-control" required>  
 </div>  
 <div class="form-group">  
 <label for="password">Password:</label>  
 <input type="password" id="password" name="password" class="form-control" required>  
 </div>  
 <button type="submit" class="btn btn-primary">Login</button>  
 </form>  
 `;  
 document.getElementById('login-form').addEventListener('submit', login);  
 }  
}  
  
function showTable(category) {  
 fetch(`/api/tables`)  
 .then(response => response.json())  
 .then(data => {  
 const mainContent = document.getElementById('main-content');  
 mainContent.innerHTML = `<h2>${category}</h2>`;  
 const tableItems = data.tables[category] || [];  
 mainContent.innerHTML += `  
 <div class="flex-container">  
 ${tableItems.map(item => `<div class="flex-item">${item}</div>`).join('')}  
 </div>  
 `;  
 })  
 .catch(error => console.error('Error:', error));  
}  
  
function login(event) {  
 event.preventDefault();  
 const username = document.getElementById('username').value;  
 const password = document.getElementById('password').value;  
  
 fetch('/login', {  
 method: 'POST',  
 headers: { 'Content-Type': 'application/json' },  
 body: JSON.stringify({ username, password })  
 })  
 .then(response => response.json())  
 .then(data => {  
 if (data.success) {  
 isAdminLoggedIn = true;  
 loadManagementContent();  
 } else {  
 alert('Login failed');  
 }  
 })  
 .catch(error => console.error('Error:', error));  
}  
  
function logout() {  
 fetch('/logout', { method: 'POST' })  
 .then(response => response.json())  
 .then(data => {  
 if (data.success) {  
 isAdminLoggedIn = false;  
 loadManagementContent();  
 }  
 })  
 .catch(error => console.error('Error:', error));  
}  
  
function showAdminSection(section) {  
 const mainContent = document.getElementById('main-content');  
 if (section === 'reports') {  
 mainContent.innerHTML = `  
 <h3>Manage Reports</h3>  
 <form id="report-form">  
 <div class="form-group">  
 <label for="report-title">Title:</label>  
 <input type="text" id="report-title" name="title" class="form-control" required>  
 </div>  
 <div class="form-group">  
 <label for="report-content">Content:</label>  
 <textarea id="report-content" name="content" class="form-control" required></textarea>  
 </div>  
 <button type="submit" class="btn btn-primary">Add Report</button>  
 </form>  
 `;  
 document.getElementById('report-form').addEventListener('submit', addReport);  
 } else if (section === 'links') {  
 mainContent.innerHTML = `  
 <h3>Manage Links</h3>  
 <form id="link-form">  
 <div class="form-group">  
 <label for="link-url">URL:</label>  
 <input type="url" id="link-url" name="url" class="form-control" required>  
 </div>  
 <div class="form-group">  
 <label for="link-category">Category:</label>  
 <input type="text" id="link-category" name="category" class="form-control" required>  
 </div>  
 <button type="submit" class="btn btn-primary">Add Link</button>  
 </form>  
 `;  
 document.getElementById('link-form').addEventListener('submit', addLink);  
 }  
}  
  
function addReport(event) {  
 event.preventDefault();  
 const title = document.getElementById('report-title').value;  
 const content = document.getElementById('report-content').value;  
  
 fetch('/api/reports', {  
 method: 'POST',  
 headers: { 'Content-Type': 'application/json' },  
 body: JSON.stringify({ title, content })  
 })  
 .then(response => response.json())  
 .then(data => {  
 if (data.success) {  
 alert('Report added successfully');  
 loadContent('reports'); // Load reports again to show the newly added report  
 } else {  
 alert('Failed to add report');  
 }  
 })  
 .catch(error => console.error('Error:', error));  
}  
  
function addLink(event) {  
 event.preventDefault();  
 const url = document.getElementById('link-url').value;  
 const category = document.getElementById('link-category').value;  
  
 fetch('/api/links', {  
 method: 'POST',  
 headers: { 'Content-Type': 'application/json' },  
 body: JSON.stringify({ url, category })  
 })  
 .then(response => response.json())  
 .then(data => {  
 if (data.success) {  
 alert('Link added successfully');  
 loadContent('links'); // Load links again to show the newly added link  
 } else {  
 alert('Failed to add link');  
 }  
 })  
 .catch(error => console.error('Error:', error));  
}  
  
function deleteReport(id) {  
 fetch(`/api/reports/${id}`, {  
 method: 'DELETE'  
 })  
 .then(response => response.json())  
 .then(data => {  
 if (data.success) {  
 alert('Report deleted successfully');  
 loadContent('reports'); // Load reports again to reflect the deletion  
 } else {  
 alert('Failed to delete report');  
 }  
 })  
 .catch(error => console.error('Error:', error));  
}  
  
function deleteLink(id) {  
 fetch(`/api/links/${id}`, {  
 method: 'DELETE'  
 })  
 .then(response => response.json())  
 .then(data => {  
 if (data.success) {  
 alert('Link deleted successfully');  
 loadContent('links'); // Load links again to reflect the deletion  
 } else {  
 alert('Failed to delete link');  
 }  
 })  
 .catch(error => console.error('Error:', error));  
}

### 3. Web Services

Η εφαρμογή χρησιμοποιεί AJAX κλήσεις για να ανακτήσει δεδομένα από τον διακομιστή. Οι κλήσεις γίνονται μέσω της fetch API, η οποία επιτρέπει την ανάκτηση δεδομένων από τη βάση δεδομένων ή τα αρχεία δεδομένων στον διακομιστή.

#### Παράδειγμα AJAX Κλήσης:

JavaScript

fetch(`/api/${section}`)  
 .then(response => response.json())  
 .then(data => {  
 // Επεξεργασία των δεδομένων που ανακτήθηκαν  
 })  
 .catch(error => {  
 console.error('Error:', error);  
 });

### 4. Βάση Δεδομένων ή Αρχεία Δεδομένων

Τα δεδομένα της εφαρμογής αποθηκεύονται σε μια βάση δεδομένων ή σε αρχεία JSON. Οι κλήσεις προς το API ανακτούν τα δεδομένα αυτά και τα προβάλλουν στην ιστοσελίδα.

#### Παράδειγμα Δεδομένων σε JSON:

JSON

{  
 "biography": {  
 "title": "Vincent Van Gogh",  
 "content": "Ο Vincent Van Gogh ήταν ένας Ολλανδός μετα-ιμπρεσιονιστής ζωγράφος..."  
 },  
 "tables": {  
 "categories": ["Κατηγορία 1", "Κατηγορία 2"],  
 "tables": {  
 "Κατηγορία 1": ["Πίνακας 1", "Πίνακας 2"],  
 "Κατηγορία 2": ["Πίνακας 3", "Πίνακας 4"]  
 }

let isAdminLoggedIn = false;  
  
document.addEventListener('DOMContentLoaded', () => {  
 loadContent('biography');  
});

Αρχικοποιούμε μια μεταβλητή isAdminLoggedIn για να ελέγξουμε αν ο διαχειριστής είναι συνδεδεμένος και, όταν φορτωθεί η σελίδα, καλούμε τη λειτουργία loadContent για να φορτώσουμε το περιεχόμενο της βιογραφίας.

#### Λειτουργία loadContent(section)

JavaScript

function loadContent(section) {  
 if (section === 'management') {  
 loadManagementContent();  
 return;  
 }  
  
 fetch(`/api/${section}`)  
 .then(response => response.json())  
 .then(data => {  
 const mainContent = document.getElementById('main-content');  
   
 if (section === 'biography') {  
 mainContent.innerHTML = `<h2>${data.title}</h2><p>${data.content}</p>`;  
 } else if (section === 'tables') {  
 mainContent.innerHTML = '<h2>Tables</h2>' +  
 data.categories.map(category => `<h3>${category}</h3><div class="flex-container">` +  
 data.tables[category].map(item => `<div class="flex-item">${item}</div>`).join('') + '</div>').join('');  
 } else if (section === 'reports') {  
 mainContent.innerHTML = '<h2>Reports</h2>' +  
 data.map(report => `<div class="report"><h3>${report.title}</h3><p>${report.content}</p>${isAdminLoggedIn ? `<button class="btn btn-danger" onclick="deleteReport('${report.\_id}')">Delete</button>` : ''}</div>`).join('');  
 } else if (section === 'links') {  
 mainContent.innerHTML = '<h2>Links</h2>' +  
 data.map(link => `<div class="link"><a href="${link.url}" target="\_blank">${link.url}</a><p>Category: ${link.category}</p>${isAdminLoggedIn ? `<button class="btn btn-danger" onclick="deleteLink('${link.\_id}')">Delete</button>` : ''}</div>`).join('');  
 }  
 })  
 .catch(error => {  
 console.error('Error:', error);  
 const mainContent = document.getElementById('main-content');  
 mainContent.innerHTML = '<p>Error loading content. Please try again later.</p>';  
 });  
}

Η λειτουργία loadContent φορτώνει δυναμικά το περιεχόμενο της σελίδας από ένα API. Ανάλογα με το τμήμα (section) που ζητείται, η λειτουργία διαμορφώνει το περιεχόμενο και το εμφανίζει στο main-content.

#### Λειτουργία loadManagementContent

JavaScript

function loadManagementContent() {  
 const mainContent = document.getElementById('main-content');  
  
 if (isAdminLoggedIn) {  
 mainContent.innerHTML = `  
 <h2>Management</h2>  
 <button id="logout-button" class="btn btn-primary">Logout</button>  
 <div id="admin-options">  
 <h3>Management Options</h3>  
 <ul class="list-unstyled">  
 <li><a href="#" onclick="showAdminSection('reports')">Manage Reports</a></li>  
 <li><a href="#" onclick="showAdminSection('links')">Manage Links</a></li>  
 </ul>  
 </div>  
 `;  
 document.getElementById('logout-button').addEventListener('click', logout);  
 } else {  
 mainContent.innerHTML = `  
 <h2>Management</h2>  
 <form id="login-form">  
 <div class="form-group">  
 <label for="username">Username:</label>  
 <input type="text" id="username" name="username" class="form-control" required>  
 </div>  
 <div class="form-group">  
 <label for="password">Password:</label>  
 <input type="password" id="password" name="password" class="form-control" required>  
 </div>  
 <button type="submit" class="btn btn-primary">Login</button>  
 </form>  
 `;  
 document.getElementById('login-form').addEventListener('submit', login);  
 }  
}

Αυτή η λειτουργία φορτώνει το περιεχόμενο της διαχείρισης. Αν ο διαχειριστής είναι συνδεδεμένος, εμφανίζει επιλογές διαχείρισης και ένα κουμπί αποσύνδεσης. Αν όχι, φορτώνει τη φόρμα εισόδου.

#### Λειτουργία showTable

JavaScript

function showTable(category) {  
 fetch(`/api/tables`)  
 .then(response => response.json())  
 .then(data => {  
 const mainContent = document.getElementById('main-content');  
 mainContent.innerHTML = `<h2>${category}</h2>`;  
 const tableItems = data.tables[category] || [];  
 mainContent.innerHTML += `  
 <div class="flex-container">  
 ${tableItems.map(item => `<div class="flex-item">${item}</div>`).join('')}  
 </div>  
 `;  
 })  
 .catch(error => console.error('Error:', error));  
}

#### Λειτουργίες Εισόδου και Εξόδου

JavaScript

function login(event) {  
 event.preventDefault();  
 const username = document.getElementById('username').value;  
 const password = document.getElementById('password').value;  
  
 fetch('/login', {  
 method: 'POST',  
 headers: { 'Content-Type': 'application/json' },  
 body: JSON.stringify({ username, password })  
 })  
 .then(response => response.json())  
 .then(data => {  
 if (data.success) {  
 isAdminLoggedIn = true;  
 loadManagementContent();  
 } else {  
 alert('Login failed');  
 }  
 })  
 .catch(error => console.error('Error:', error));  
}  
  
function logout() {  
 fetch('/logout', { method: 'POST' })  
 .then(response => response.json())  
 .then(data => {  
 if (data.success) {  
 isAdminLoggedIn = false;  
 loadManagementContent();  
 }  
 })  
 .catch(error => console.error('Error:', error));  
}

Η λειτουργία login επιτρέπει στον διαχειριστή να συνδεθεί, ενώ η λειτουργία logout επιτρέπει την αποσύνδεση.

#### Λειτουργίες Προσθήκης και Διαγραφής Αναφορών και Συνδέσμων

JavaScript

function showAdminSection(section) {  
 const mainContent = document.getElementById('main-content');  
 if (section === 'reports') {  
 mainContent.innerHTML = `  
 <h3>Manage Reports</h3>  
 <form id="report-form">  
 <div class="form-group">  
 <label for="report-title">Title:</label>  
 <input type="text" id="report-title" name="title" class="form-control" required>  
 </div>  
 <div class="form-group">  
 <label for="report-content">Content:</label>  
 <textarea id="report-content" name="content" class="form-control" required></textarea>  
 </div>  
 <button type="submit" class="btn btn-primary">Add Report</button>  
 </form>  
 `;  
 document.getElementById('report-form').addEventListener('submit', addReport);  
 } else if (section === 'links') {  
 mainContent.innerHTML = `  
 <h3>Manage Links</h3>  
 <form id="link-form">  
 <div class="form-group">  
 <label for="link-url">URL:</label>  
 <input type="url" id="link-url" name="url" class="form-control" required>  
 </div>  
 <div class="form-group">  
 <label for="link-category">Category:</label>  
 <input type="text" id="link-category" name="category" class="form-control" required>  
 </div>  
 <button type="submit" class="btn btn-primary">Add Link</button>  
 </form>  
 `;  
 document.getElementById('link-form').addEventListener('submit', addLink);  
 }  
}  
  
function addReport(event) {  
 event.preventDefault();  
 const title = document.getElementById('report-title').value;  
 const content = document.getElementById('report-content').value;  
  
 fetch('/api/reports', {  
 method: 'POST',  
 headers: { 'Content-Type': 'application/json' },  
 body: JSON.stringify({ title, content })  
 })  
 .then(response => response.json())  
 .then(data => {  
 if (data.success) {  
 alert('Report added successfully');  
 loadContent('reports'); // Load reports again to show the newly added report  
 } else {  
 alert('Failed to add report');  
 }  
 })  
 .catch(error => console.error('Error:', error));  
}  
  
function addLink(event) {  
 event.preventDefault();  
 const url = document.getElementById('link-url').value;  
 const category = document.getElementById('link-category').value;  
  
 fetch('/api/links', {  
 method: 'POST',  
 headers: { 'Content-Type': 'application/json' },  
 body: JSON.stringify({ url, category })  
 })  
 .then(response => response.json())  
 .then(data => {  
 if (data.success) {  
 alert('Link added successfully');  
 loadContent('links'); // Load links again to show the newly added link  
 } else {  
 alert('Failed to add link');  
 }  
 })  
 .catch(error => console.error('Error:', error));  
}  
  
function deleteReport(id) {  
 fetch(`/api/reports/${id}`, {  
 method: 'DELETE'  
 })  
 .then(response => response.json())  
 .then(data => {  
 if (data.success) {  
 alert('Report deleted successfully');  
 loadContent('reports'); // Load reports again to reflect the deletion  
 } else {  
 alert('Failed to delete report');  
 }  
 })  
 .catch(error => console.error('Error:', error));  
}  
  
function deleteLink(id) {  
 fetch(`/api/links/${id}`, {  
 method: 'DELETE'  
 })  
 .then(response => response.json())  
 .then(data => {  
 if (data.success) {  
 alert('Link deleted successfully');  
 loadContent('links'); // Load links again to reflect the deletion  
 } else {  
 alert('Failed to delete link');  
 }  
 })  
 .catch(error => console.error('Error:', error));  
}

Αυτές οι λειτουργίες επιτρέπουν στον διαχειριστή να προσθέτει και να διαγράφει αναφορές και συνδέσμους μέσω API κλήσεων.