class EventManager {

constructor() {

this.events = JSON.parse(localStorage.getItem('events')) || [];

this.users = JSON.parse(localStorage.getItem('users')) || [];

this.currentUser = JSON.parse(localStorage.getItem('currentUser')) || null;

this.groups = JSON.parse(localStorage.getItem('groups')) || [];

this.forums = JSON.parse(localStorage.getItem('forums')) || [];

this.initializeEventListeners();

this.populateFilters();

this.initializeGroups();

this.initializeForums();

this.updateJoinButton();

}

initializeEventListeners() {

document.getElementById('searchActivities').addEventListener('input', (e) => {

this.filterEvents(e.target.value);

});

['category', 'location', 'date'].forEach(filter => {

document.getElementById(`${filter}Filter`).addEventListener('change', () => {

this.applyFilters();

});

});

document.getElementById('newActivityForm').addEventListener('submit', (e) => {

e.preventDefault();

this.createNewActivity();

});

document.querySelector('.close').addEventListener('click', () => {

this.closeModal();

});

document.querySelector('.btn-join').addEventListener('click', () => {

if (this.currentUser) {

this.logout();

} else {

this.showLoginModal();

}

});

document.querySelector('.btn-explore').addEventListener('click', () => {

document.getElementById('events').scrollIntoView({ behavior: 'smooth' });

});

this.initializeFooterModalListeners();

this.initializeAuthModalListeners();

this.initializeForumModalListeners();

}

updateJoinButton() {

const joinButton = document.querySelector('.btn-join');

if (this.currentUser) {

joinButton.textContent = `Logout (${this.currentUser.username})`;

} else {

joinButton.textContent = 'Join Community';

}

}

initializeAuthModalListeners() {

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

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

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

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

const showSignupLink = document.getElementById('showSignup');

const showLoginLink = document.getElementById('showLogin');

const closeButtons = document.querySelectorAll('.close');

const openModal = (modal) => {

modal.style.display = 'flex';

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

};

const closeModal = (modal) => {

modal.style.display = 'none';

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

};

loginForm.addEventListener('submit', (e) => {

e.preventDefault();

const username = document.getElementById('loginUsername').value;

const password = document.getElementById('loginPassword').value;

this.login(username, password);

});

signupForm.addEventListener('submit', (e) => {

e.preventDefault();

const username = document.getElementById('signupUsername').value;

const email = document.getElementById('signupEmail').value;

const password = document.getElementById('signupPassword').value;

const confirmPassword = document.getElementById('signupConfirmPassword').value;

this.signup(username, email, password, confirmPassword);

});

showSignupLink.addEventListener('click', (e) => {

e.preventDefault();

closeModal(loginModal);

openModal(signupModal);

});

showLoginLink.addEventListener('click', (e) => {

e.preventDefault();

closeModal(signupModal);

openModal(loginModal);

});

closeButtons.forEach(button => {

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

const modal = button.closest('.modal');

closeModal(modal);

});

});

window.addEventListener('click', (e) => {

if (e.target.classList.contains('modal')) {

closeModal(e.target);

}

});

}

initializeForumModalListeners() {

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

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

const closeButtons = document.querySelectorAll('.close');

const openModal = (modal) => {

modal.style.display = 'flex';

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

};

const closeModal = (modal) => {

modal.style.display = 'none';

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

};

newPostForm.addEventListener('submit', (e) => {

e.preventDefault();

const forumId = parseInt(forumModal.dataset.forumId, 10);

const content = document.getElementById('newPostContent').value;

this.createPost(forumId, content);

});

closeButtons.forEach(button => {

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

const modal = button.closest('.modal');

closeModal(modal);

});

});

window.addEventListener('click', (e) => {

if (e.target.classList.contains('modal')) {

closeModal(e.target);

}

});

}

showLoginModal() {

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

loginModal.style.display = 'flex';

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

}

login(username, password) {

const user = this.users.find(u => u.username === username && u.password === password);

if (user) {

this.currentUser = user;

localStorage.setItem('currentUser', JSON.stringify(this.currentUser));

alert(`Welcome back, ${username}!`);

this.closeModal();

this.updateJoinButton();

this.renderGroups();

this.renderForums();

} else {

alert('Invalid username or password. Please try again.');

}

}

signup(username, email, password, confirmPassword) {

if (password !== confirmPassword) {

alert('Passwords do not match. Please try again.');

return;

}

if (this.users.some(u => u.username === username)) {

alert('Username already exists. Please choose a different username.');

return;

}

if (this.users.some(u => u.email === email)) {

alert('Email already registered. Please use a different email.');

return;

}

const newUser = {

id: Date.now(),

username,

email,

password,

joinedGroups: [],

joinedForums: []

};

this.users.push(newUser);

localStorage.setItem('users', JSON.stringify(this.users));

alert('Sign up successful! Please log in.');

this.closeModal();

this.showLoginModal();

}

logout() {

this.currentUser = null;

localStorage.removeItem('currentUser');

alert('You have been logged out.');

this.updateJoinButton();

this.renderGroups();

this.renderForums();

}

joinGroup(groupId) {

if (!this.currentUser) {

alert('Please log in to join a group.');

this.showLoginModal();

return;

}

const userIndex = this.users.findIndex(u => u.id === this.currentUser.id);

if (userIndex !== -1) {

if (!this.users[userIndex].joinedGroups) {

this.users[userIndex].joinedGroups = [];

}

if (!this.users[userIndex].joinedGroups.includes(groupId)) {

this.users[userIndex].joinedGroups.push(groupId);

this.currentUser.joinedGroups = this.users[userIndex].joinedGroups;

localStorage.setItem('users', JSON.stringify(this.users));

localStorage.setItem('currentUser', JSON.stringify(this.currentUser));

const group = this.groups.find(g => g.id === groupId);

if (group) {

group.members += 1;

localStorage.setItem('groups', JSON.stringify(this.groups));

}

alert(`You have joined ${group.name}!`);

this.renderGroups();

} else {

alert('You are already a member of this group.');

}

}

}

joinForum(forumId) {

if (!this.currentUser) {

alert('Please log in to join a forum.');

this.showLoginModal();

return;

}

const userIndex = this.users.findIndex(u => u.id === this.currentUser.id);

if (userIndex !== -1) {

if (!this.users[userIndex].joinedForums) {

this.users[userIndex].joinedForums = [];

}

if (!this.users[userIndex].joinedForums.includes(forumId)) {

this.users[userIndex].joinedForums.push(forumId);

this.currentUser.joinedForums = this.users[userIndex].joinedForums;

localStorage.setItem('users', JSON.stringify(this.users));

localStorage.setItem('currentUser', JSON.stringify(this.currentUser));

const forum = this.forums.find(f => f.id === forumId);

if (forum) {

forum.members += 1;

localStorage.setItem('forums', JSON.stringify(this.forums));

}

alert(`You have joined ${forum.title}!`);

this.renderForums();

this.showForumModal(forumId);

} else {

this.showForumModal(forumId);

}

}

}

showForumModal(forumId) {

const forum = this.forums.find(f => f.id === forumId);

if (!forum) return;

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

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

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

forumModal.dataset.forumId = forumId;

forumModalTitle.textContent = forum.title;

forumPosts.innerHTML = '';

const posts = JSON.parse(localStorage.getItem(`forumPosts\_${forumId}`)) || [];

posts.forEach(post => {

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

postElement.className = 'forum-post';

postElement.innerHTML = `

<h4>${post.author}</h4>

<p>${post.content}</p>

<p class="post-meta">Posted on ${new Date(post.timestamp).toLocaleString()}</p>

`;

forumPosts.appendChild(postElement);

});

document.getElementById('newPostContent').value = '';

forumModal.style.display = 'flex';

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

}

createPost(forumId, content) {

if (!this.currentUser) {

alert('Please log in to post in the forum.');

this.showLoginModal();

return;

}

const forum = this.forums.find(f => f.id === forumId);

if (!forum) return;

const posts = JSON.parse(localStorage.getItem(`forumPosts\_${forumId}`)) || [];

const newPost = {

id: Date.now(),

author: this.currentUser.username,

content,

timestamp: Date.now()

};

posts.push(newPost);

localStorage.setItem(`forumPosts\_${forumId}`, JSON.stringify(posts));

forum.posts += 1;

localStorage.setItem('forums', JSON.stringify(this.forums));

this.showForumModal(forumId);

}

initializeFooterModalListeners() {

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

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

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

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

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

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

if (!termsModal || !privacyModal || !contactModal || !termsLink || !privacyLink || !contactLink) {

console.error('One or more modal or link elements are missing in the HTML.');

return;

}

const openModal = (modal) => {

modal.style.display = 'flex';

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

};

const closeModal = (modal) => {

modal.style.display = 'none';

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

};

termsLink.addEventListener('click', (e) => {

e.preventDefault();

openModal(termsModal);

});

privacyLink.addEventListener('click', (e) => {

e.preventDefault();

openModal(privacyModal);

});

contactLink.addEventListener('click', (e) => {

e.preventDefault();

openModal(contactModal);

});

const closeButtons = document.querySelectorAll('.close');

closeButtons.forEach(button => {

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

const modal = button.closest('.modal');

closeModal(modal);

});

});

window.addEventListener('click', (e) => {

if (e.target.classList.contains('modal')) {

closeModal(e.target);

}

});

}

closeModal() {

const modals = document.querySelectorAll('.modal');

modals.forEach(modal => {

modal.style.display = 'none';

});

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

}

populateFilters() {

const categories = [...new Set(this.events.map(event => event.category))];

const locations = [...new Set(this.events.map(event => event.location))];

const dates = [...new Set(this.events.map(event => event.date))];

this.populateSelect('categoryFilter', categories, 'All Categories');

this.populateSelect('locationFilter', locations, 'All Locations');

this.populateSelect('dateFilter', dates, 'Any Date');

}

populateSelect(selectId, options, defaultOption) {

const select = document.getElementById(selectId);

select.innerHTML = `<option value="">${defaultOption}</option>`;

options.forEach(option => {

const opt = document.createElement('option');

opt.value = option;

opt.textContent = option;

select.appendChild(opt);

});

}

filterEvents(searchTerm) {

const filteredEvents = this.events.filter(event =>

event.title.toLowerCase().includes(searchTerm.toLowerCase()) ||

event.category.toLowerCase().includes(searchTerm.toLowerCase()) ||

event.description.toLowerCase().includes(searchTerm.toLowerCase())

);

this.renderEvents(filteredEvents);

}

applyFilters() {

let filtered = [...this.events];

const category = document.getElementById('categoryFilter').value;

const location = document.getElementById('locationFilter').value;

const date = document.getElementById('dateFilter').value;

if (category) filtered = filtered.filter(event => event.category === category);

if (location) filtered = filtered.filter(event => event.location === location);

if (date) filtered = filtered.filter(event => event.date === date);

this.renderEvents(filtered);

}

createNewActivity() {

const formData = {

title: document.getElementById('title').value,

category: document.getElementById('category').value,

location: document.getElementById('location').value,

date: document.getElementById('date').value,

time: document.getElementById('time').value,

description: document.getElementById('description').value,

capacity: document.getElementById('capacity').value

};

const newEvent = {

...formData,

id: Date.now(),

participants: []

};

this.events.push(newEvent);

localStorage.setItem('events', JSON.stringify(this.events));

this.renderEvents(this.events);

this.renderUpcomingEvents();

this.populateFilters();

this.resetForm();

alert('Event created successfully!');

}

deleteEvent(eventId) {

if (confirm('Are you sure you want to delete this event?')) {

this.events = this.events.filter(event => event.id !== eventId);

localStorage.setItem('events', JSON.stringify(this.events));

this.renderEvents(this.events);

this.populateFilters();

alert('Event deleted successfully!');

}

}

renderEvents(events) {

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

eventsGrid.innerHTML = '';

events.forEach(event => {

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

eventCard.className = 'event-card';

eventCard.innerHTML = `

<h3>${event.title}</h3>

<p><strong>Category:</strong> ${event.category}</p>

<p><strong>Location:</strong> ${event.location}</p>

<p><strong>Date:</strong> ${event.date}</p>

<p><strong>Time:</strong> ${event.time}</p>

<p><strong>Capacity:</strong> ${event.capacity}</p>

<button class="delete-btn" data-event-id="${event.id}">Delete Event</button>

`;

eventCard.addEventListener('click', (e) => {

if (!e.target.classList.contains('delete-btn')) {

this.showModal(event);

}

});

const deleteBtn = eventCard.querySelector('.delete-btn');

deleteBtn.addEventListener('click', (e) => {

e.stopPropagation();

this.deleteEvent(event.id);

});

eventsGrid.appendChild(eventCard);

});

this.renderUpcomingEvents();

}

showModal(event) {

document.getElementById('modalTitle').textContent = event.title;

document.getElementById('modalCategory').textContent = `Category: ${event.category}`;

document.getElementById('modalLocation').textContent = `Location: ${event.location}`;

document.getElementById('modalDate').textContent = `Date: ${event.date}`;

document.getElementById('modalTime').textContent = `Time: ${event.time}`;

document.getElementById('modalDescription').textContent = `Description: ${event.description}`;

document.getElementById('modalCapacity').textContent = `Capacity: ${event.capacity}`;

const modalContent = document.querySelector('.modal-content');

const deleteBtn = document.createElement('button');

deleteBtn.className = 'delete-btn';

deleteBtn.textContent = 'Delete Event';

deleteBtn.onclick = () => {

this.deleteEvent(event.id);

this.closeModal();

};

const existingDeleteBtn = modalContent.querySelector('.delete-btn');

if (existingDeleteBtn) {

existingDeleteBtn.remove();

}

document.querySelector('.btn-join-event').onclick = () => {

alert(`You have joined ${event.title}!`);

this.closeModal();

};

modalContent.appendChild(deleteBtn);

document.getElementById('eventModal