**SE**

**User stories 68-71:**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>SCAD Office - Internship Portal</title>

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

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

</head>

<body>

<header>

<div class="container">

<h1>SCAD Office Internship Portal</h1>

<p>Connecting students with internship opportunities</p>

</div>

</header>

<div class="container">

<div class="header-controls">

<div class="tabs">

<button class="tab-btn active" data-tab="appointments">Appointments</button>

<button class="tab-btn" data-tab="online-users">Online Users</button>

</div>

<div class="notification-icon">

<i class="fas fa-bell fa-2x"></i>

<span class="notification-badge" id="notificationCount">0</span>

</div>

</div>

<div id="appointmentsTab" class="tab-content active">

<div id="appointmentsContainer">

<!-- Appointment cards will be dynamically inserted here -->

</div>

</div>

<div id="onlineUsersTab" class="tab-content">

<div class="online-users-header">

<h3>Currently Online</h3>

<span class="online-count" id="onlineCount">0 users online</span>

</div>

<div id="onlineUsersContainer">

<!-- Online users will be dynamically inserted here -->

</div>

</div>

</div>

<div class="notification-panel" id="notificationPanel">

<div class="notification-header">

<h3>Notifications</h3>

<div class="notification-actions">

<button id="markAllRead">Mark all as read</button>

<span class="close-panel">&times;</span>

</div>

</div>

<div id="notificationsList">

<!-- Notifications will be dynamically inserted here -->

</div>

</div>

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

</body>

</html>

**User stories 72-78**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>SCAD Office - Internship Video Call</title>

<style>

:root {

--primary-color: #2c3e50;

--secondary-color: #3498db;

--accent-color: #e74c3c;

--light-color: #ecf0f1;

--dark-color: #2c3e50;

--success-color: #2ecc71;

}

\* {

margin: 0;

padding: 0;

box-sizing: border-box;

font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;

}

body {

background-color: #f5f5f5;

color: var(--dark-color);

}

.container {

display: flex;

min-height: 100vh;

}

/\* Sidebar styles \*/

.sidebar {

width: 300px;

background-color: var(--primary-color);

color: white;

padding: 20px;

display: flex;

flex-direction: column;

}

.logo {

text-align: center;

margin-bottom: 30px;

}

.logo h1 {

font-size: 1.5rem;

margin-bottom: 5px;

color: var(--light-color);

}

.logo p {

font-size: 0.9rem;

color: #bdc3c7;

}

.nav-menu {

flex-grow: 1;

}

.nav-item {

padding: 12px 15px;

margin-bottom: 5px;

border-radius: 5px;

cursor: pointer;

display: flex;

align-items: center;

transition: background-color 0.3s;

}

.nav-item:hover {

background-color: rgba(255, 255, 255, 0.1);

}

.nav-item.active {

background-color: var(--secondary-color);

}

.nav-item i {

margin-right: 10px;

font-size: 1.2rem;

}

.user-profile {

display: flex;

align-items: center;

padding: 15px 0;

border-top: 1px solid rgba(255, 255, 255, 0.1);

}

.user-avatar {

width: 40px;

height: 40px;

border-radius: 50%;

background-color: var(--secondary-color);

display: flex;

align-items: center;

justify-content: center;

margin-right: 10px;

font-weight: bold;

}

.user-info h4 {

font-size: 0.9rem;

margin-bottom: 3px;

}

.user-info p {

font-size: 0.8rem;

color: #bdc3c7;

}

/\* Main content styles \*/

.main-content {

flex-grow: 1;

display: flex;

flex-direction: column;

padding: 20px;

}

.header {

display: flex;

justify-content: space-between;

align-items: center;

margin-bottom: 20px;

}

.header h2 {

color: var(--primary-color);

}

.notification-bell {

position: relative;

cursor: pointer;

font-size: 1.5rem;

color: var(--primary-color);

}

.notification-count {

position: absolute;

top: -5px;

right: -5px;

background-color: var(--accent-color);

color: white;

border-radius: 50%;

width: 20px;

height: 20px;

display: flex;

align-items: center;

justify-content: center;

font-size: 0.7rem;

}

/\* Video call section \*/

.video-call-container {

flex-grow: 1;

display: flex;

flex-direction: column;

background-color: white;

border-radius: 10px;

box-shadow: 0 2px 10px rgba(0, 0, 0, 0.1);

overflow: hidden;

}

.video-display {

flex-grow: 1;

display: flex;

flex-direction: column;

position: relative;

background-color: #1a1a1a;

}

.remote-video {

flex-grow: 1;

object-fit: cover;

}

.local-video {

position: absolute;

bottom: 20px;

right: 20px;

width: 200px;

height: 150px;

border-radius: 5px;

border: 2px solid white;

object-fit: cover;

box-shadow: 0 2px 5px rgba(0, 0, 0, 0.2);

}

.call-controls {

display: flex;

justify-content: center;

padding: 15px;

background-color: white;

border-top: 1px solid #eee;

}

.control-btn {

width: 50px;

height: 50px;

border-radius: 50%;

border: none;

margin: 0 10px;

cursor: pointer;

display: flex;

align-items: center;

justify-content: center;

font-size: 1.2rem;

transition: all 0.3s;

color: white;

}

.control-btn:hover {

transform: scale(1.1);

}

.mute-btn {

background-color: var(--secondary-color);

}

.video-btn {

background-color: var(--secondary-color);

}

.share-btn {

background-color: #9b59b6;

}

.end-call-btn {

background-color: var(--accent-color);

}

.call-btn {

background-color: var(--success-color);

}

/\* Notification panel \*/

.notification-panel {

position: fixed;

top: 0;

right: -400px;

width: 400px;

height: 100vh;

background-color: white;

box-shadow: -2px 0 10px rgba(0, 0, 0, 0.1);

transition: right 0.3s;

z-index: 100;

display: flex;

flex-direction: column;

}

.notification-panel.open {

right: 0;

}

.notification-header {

padding: 20px;

border-bottom: 1px solid #eee;

display: flex;

justify-content: space-between;

align-items: center;

}

.notification-header h3 {

color: var(--primary-color);

}

.close-notifications {

background: none;

border: none;

font-size: 1.5rem;

cursor: pointer;

color: var(--dark-color);

}

.notification-list {

flex-grow: 1;

overflow-y: auto;

padding: 10px;

}

.notification-item {

padding: 15px;

margin-bottom: 10px;

border-radius: 5px;

background-color: #f9f9f9;

display: flex;

align-items: center;

}

.notification-item.unread {

background-color: #e3f2fd;

border-left: 3px solid var(--secondary-color);

}

.notification-icon {

width: 40px;

height: 40px;

border-radius: 50%;

background-color: var(--secondary-color);

color: white;

display: flex;

align-items: center;

justify-content: center;

margin-right: 15px;

font-size: 1.2rem;

}

.notification-content h4 {

font-size: 0.9rem;

margin-bottom: 5px;

}

.notification-content p {

font-size: 0.8rem;

color: #666;

}

.notification-time {

font-size: 0.7rem;

color: #999;

margin-top: 3px;

}

/\* Incoming call modal \*/

.incoming-call-modal {

position: fixed;

top: 0;

left: 0;

width: 100%;

height: 100%;

background-color: rgba(0, 0, 0, 0.7);

display: flex;

justify-content: center;

align-items: center;

z-index: 200;

display: none;

}

.incoming-call-box {

background-color: white;

border-radius: 10px;

padding: 30px;

text-align: center;

max-width: 400px;

width: 90%;

}

.caller-info {

margin-bottom: 20px;

}

.caller-avatar {

width: 80px;

height: 80px;

border-radius: 50%;

background-color: var(--secondary-color);

color: white;

display: flex;

align-items: center;

justify-content: center;

margin: 0 auto 15px;

font-size: 2rem;

font-weight: bold;

}

.caller-info h3 {

margin-bottom: 5px;

color: var(--dark-color);

}

.caller-info p {

color: #666;

font-size: 0.9rem;

}

.call-actions {

display: flex;

justify-content: center;

gap: 20px;

}

.accept-call {

background-color: var(--success-color);

color: white;

border: none;

border-radius: 50%;

width: 60px;

height: 60px;

font-size: 1.5rem;

cursor: pointer;

display: flex;

align-items: center;

justify-content: center;

}

.reject-call {

background-color: var(--accent-color);

color: white;

border: none;

border-radius: 50%;

width: 60px;

height: 60px;

font-size: 1.5rem;

cursor: pointer;

display: flex;

align-items: center;

justify-content: center;

}

/\* Responsive adjustments \*/

@media (max-width: 768px) {

.container {

flex-direction: column;

}

.sidebar {

width: 100%;

padding: 15px;

}

.main-content {

padding: 15px;

}

.local-video {

width: 120px;

height: 90px;

}

.notification-panel {

width: 100%;

right: -100%;

}

}

</style>

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

</head>

<body>

<div class="container">

<!-- Sidebar -->

<div class="sidebar">

<div class="logo">

<h1>SCAD Office</h1>

<p>Internship Portal</p>

</div>

<div class="nav-menu">

<div class="nav-item">

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

<span>Dashboard</span>

</div>

<div class="nav-item">

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

<span>Internships</span>

</div>

<div class="nav-item active">

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

<span>Video Calls</span>

</div>

<div class="nav-item">

<i class="fas fa-calendar-alt"></i>

<span>Schedule</span>

</div>

<div class="nav-item">

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

<span>Companies</span>

</div>

<div class="nav-item">

<i class="fas fa-file-alt"></i>

<span>Documents</span>

</div>

<div class="nav-item">

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

<span>Settings</span>

</div>

</div>

<div class="user-profile">

<div class="user-avatar">JD</div>

<div class="user-info">

<h4>John Doe</h4>

<p>SCAD Coordinator</p>

</div>

</div>

</div>

<!-- Main Content -->

<div class="main-content">

<div class="header">

<h2>Internship Video Call</h2>

<div class="notification-bell" id="notificationBell">

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

<span class="notification-count">3</span>

</div>

</div>

<div class="video-call-container">

<div class="video-display">

<video class="remote-video" id="remoteVideo" autoplay playsinline></video>

<video class="local-video" id="localVideo" autoplay playsinline muted></video>

</div>

<div class="call-controls">

<button class="control-btn mute-btn" id="muteBtn">

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

</button>

<button class="control-btn video-btn" id="videoBtn">

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

</button>

<button class="control-btn share-btn" id="shareBtn">

<i class="fas fa-share-square"></i>

</button>

<button class="control-btn end-call-btn" id="endCallBtn">

<i class="fas fa-phone-slash"></i>

</button>

</div>

</div>

</div>

<!-- Notification Panel -->

<div class="notification-panel" id="notificationPanel">

<div class="notification-header">

<h3>Notifications</h3>

<button class="close-notifications" id="closeNotifications">

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

</button>

</div>

<div class="notification-list">

<div class="notification-item unread">

<div class="notification-icon">

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

</div>

<div class="notification-content">

<h4>Missed Call</h4>

<p>Jane Smith from TechCorp tried to call you</p>

<div class="notification-time">10 minutes ago</div>

</div>

</div>

<div class="notification-item unread">

<div class="notification-icon">

<i class="fas fa-user-clock"></i>

</div>

<div class="notification-content">

<h4>Call Ended</h4>

<p>Michael Johnson left the call unexpectedly</p>

<div class="notification-time">1 hour ago</div>

</div>

</div>

<div class="notification-item">

<div class="notification-icon">

<i class="fas fa-calendar-check"></i>

</div>

<div class="notification-content">

<h4>Meeting Scheduled</h4>

<p>New interview scheduled with DesignHub for tomorrow</p>

<div class="notification-time">2 days ago</div>

</div>

</div>

</div>

</div>

<!-- Incoming Call Modal -->

<div class="incoming-call-modal" id="incomingCallModal">

<div class="incoming-call-box">

<div class="caller-info">

<div class="caller-avatar">AS</div>

<h3>Alex Smith</h3>

<p>Internship Coordinator at InnovateTech</p>

</div>

<div class="call-actions">

<button class="accept-call" id="acceptCall">

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

</button>

<button class="reject-call" id="rejectCall">

<i class="fas fa-phone-slash"></i>

</button>

</div>

</div>

</div>

</div>

<script>

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

// DOM Elements

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

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

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

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

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

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

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

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

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

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

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

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

// Mock video streams (in a real app, these would be from WebRTC)

navigator.mediaDevices.getUserMedia({ video: true, audio: true })

.then(stream => {

localVideo.srcObject = stream;

// For demo purposes, we'll use the same stream for remote video

// In a real app, this would come from the remote peer

remoteVideo.srcObject = stream.clone();

})

.catch(err => {

console.error('Error accessing media devices:', err);

});

// Toggle notification panel

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

notificationPanel.classList.toggle('open');

});

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

notificationPanel.classList.remove('open');

});

// Toggle mute

let isMuted = false;

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

isMuted = !isMuted;

localVideo.srcObject.getAudioTracks()[0].enabled = !isMuted;

if (isMuted) {

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

muteBtn.style.backgroundColor = '#e74c3c';

} else {

muteBtn.innerHTML = '<i class="fas fa-microphone"></i>';

muteBtn.style.backgroundColor = '#3498db';

}

});

// Toggle video

let isVideoOff = false;

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

isVideoOff = !isVideoOff;

localVideo.srcObject.getVideoTracks()[0].enabled = !isVideoOff;

if (isVideoOff) {

videoBtn.innerHTML = '<i class="fas fa-video-slash"></i>';

videoBtn.style.backgroundColor = '#e74c3c';

localVideo.style.display = 'none';

} else {

videoBtn.innerHTML = '<i class="fas fa-video"></i>';

videoBtn.style.backgroundColor = '#3498db';

localVideo.style.display = 'block';

}

});

// Screen sharing

shareBtn.addEventListener('click', async function() {

try {

const screenStream = await navigator.mediaDevices.getDisplayMedia({

video: true,

audio: false

});

// Replace the remote video with screen share

// In a real app, you would send this stream to the remote peer

remoteVideo.srcObject = screenStream;

// When screen sharing stops

screenStream.getVideoTracks()[0].onended = function() {

remoteVideo.srcObject = localVideo.srcObject.clone();

};

shareBtn.innerHTML = '<i class="fas fa-check"></i>';

shareBtn.style.backgroundColor = '#2ecc71';

setTimeout(() => {

shareBtn.innerHTML = '<i class="fas fa-share-square"></i>';

shareBtn.style.backgroundColor = '#9b59b6';

}, 2000);

} catch (err) {

console.error('Error sharing screen:', err);

}

});

// End call

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

// Stop all tracks

localVideo.srcObject.getTracks().forEach(track => track.stop());

if (remoteVideo.srcObject) {

remoteVideo.srcObject.getTracks().forEach(track => track.stop());

}

// Add notification

addNotification('Call Ended', 'You ended the call with the student', 'fas fa-phone-slash');

// In a real app, you would also signal the remote peer to end the call

alert('Call ended successfully');

});

// Simulate incoming call after 3 seconds

setTimeout(() => {

incomingCallModal.style.display = 'flex';

}, 3000);

// Accept call

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

incomingCallModal.style.display = 'none';

addNotification('Call Started', 'You are now connected with Alex Smith', 'fas fa-phone');

});

// Reject call

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

incomingCallModal.style.display = 'none';

addNotification('Call Rejected', 'You rejected the call from Alex Smith', 'fas fa-phone-slash');

});

// Function to add a notification

function addNotification(title, message, icon) {

const notificationList = document.querySelector('.notification-list');

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

notificationItem.className = 'notification-item unread';

notificationItem.innerHTML = `

<div class="notification-icon">

<i class="${icon}"></i>

</div>

<div class="notification-content">

<h4>${title}</h4>

<p>${message}</p>

<div class="notification-time">Just now</div>

</div>

`;

notificationList.insertBefore(notificationItem, notificationList.firstChild);

// Update notification count

const count = document.querySelector('.notification-count');

count.textContent = parseInt(count.textContent) + 1;

}

// Simulate user leaving call (for demo purposes)

setTimeout(() => {

addNotification('Call Ended', 'Sarah Williams left the call unexpectedly', 'fas fa-user-clock');

}, 15000);

});

</script>

</body>

</html>