

javascript

```

        .where(firebase.firestore.FieldPath.documentId(), 'in', eventIds)
        .get();

    return eventsSnapshot.docs.map(doc => {
        return {
            id: doc.id,
            ...doc.data()
        };
    });
} catch (error) {
    console.error('Error getting past events:', error);
    return [];
}
}

async getVibeChecks() {
    if (!this.currentUser) return [];

    try {
        const db = firebase.firestore();

        // Get vibe checks posted by the user
        const snapshot = await db.collection('vibeChecks')
            .where('userId', '==', this.currentUser.uid)
            .orderBy('timestamp', 'desc')
            .get();

        return snapshot.docs.map(doc => {
            return {
                id: doc.id,
                ...doc.data()
            };
        });
    } catch (error) {
        console.error('Error getting vibe checks:', error);
        return [];
    }
}

async updatePreferences(preferences) {
    if (!this.currentUser) return false;

    try {

```

```
const db = firebase.firestore();
await db.collection('users').doc(this.currentUser.uid).update({
  'preferences': preferences,
  updatedAt: new Date()
});

// Refresh user data
return await this.loadUserData();
} catch (error) {
  console.error('Error updating preferences:', error);
  throw error;
}
}
}

// Create a singleton instance
const userProfile = new UserProfile();

export default userProfile;
```

2. Event Detail Page and RSVP System (Duration: 7 days)

2.1. Create Event Detail Page (Duration: 3 days)


```

<!-- Create file: pages/app/event-detail.html -->
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Event Details - The Play</title>
  <!-- Include your standard CSS and JS files here -->
  <link href="https://cdnjs.cloudflare.com/ajax/libs/tailwindcss/2.2.19/tailwind.min.css" rel="
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0-beta3/
  <link rel="stylesheet" href="../../assets/styles/main.css">
  <link rel="stylesheet" href="../../assets/styles/variables.css">
  <link rel="stylesheet" href="../../assets/styles/layout.css">
  <link rel="stylesheet" href="../../assets/styles/components.css">

  <!-- Firebase Libraries -->
  <script src="https://www.gstatic.com/firebasejs/9.22.2/firebase-app-compat.js"></script>
  <script src="https://www.gstatic.com/firebasejs/9.22.2/firebase-auth-compat.js"></script>
  <script src="https://www.gstatic.com/firebasejs/9.22.2/firebase-firestore-compat.js"></script>
  <script src="https://www.gstatic.com/firebasejs/9.22.2/firebase-storage-compat.js"></script>
  <script src="../../scripts/firebase/config.js"></script>

  <!-- Leaflet Map CSS -->
  <link rel="stylesheet" href="https://unpkg.com/leaflet@1.9.4/dist/leaflet.css" />
</head>
<body class="bg-gray-100">
  <!-- Header Container -->
  <div id="header-container"></div>

  <!-- Main Content -->
  <div class="container mx-auto px-4 py-8">
    <div class="max-w-5xl mx-auto">
      <!-- Back Button -->
      <div class="mb-6">
        <a href="javascript:history.back()" class="inline-flex items-center text-gray-600 hover
          <i class="fas fa-arrow-left mr-2"></i> Back to Events
        </a>
      </div>

      <!-- Event Loading Placeholder -->
      <div id="event-loading" class="bg-white rounded-xl shadow-md p-10 text-center">
        <div class="animate-spin rounded-full h-12 w-12 border-t-2 border-b-2 border-primary mx
        <p class="text-gray-600">Loading event details...</p>
      </div>
    </div>
  </div>

```

```
</div>
```

```
<!-- Event Details -->
```

```
<div id="event-details" class="hidden">
```

```
  <!-- Hero Image -->
```

```
  <div class="relative rounded-xl overflow-hidden h-80 bg-gray-200 mb-6">
```

```
    
```

```
    <div class="absolute top-4 right-4 flex space-x-2">
```

```
      <button id="share-event" class="bg-white bg-opacity-90 p-2 rounded-full hover:bg-op
```

```
        <i class="fas fa-share-alt"></i>
```

```
    </button>
```

```
    <button id="bookmark-event" class="bg-white bg-opacity-90 p-2 rounded-full hover:bg
```

```
      <i class="fas fa-bookmark"></i>
```

```
    </button>
```

```
  </div>
```

```
  <div class="absolute top-4 left-4 vibe-score">
```

```
    <i class="fas fa-fire mr-1"></i> <span id="event-score">0.0</span>
```

```
  </div>
```

```
</div>
```

```
<!-- Event Header -->
```

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

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

```
    <h1 id="event-title" class="text-3xl font-bold">Event Title</h1>
```

```
    <div id="event-price" class="text-xl font-semibold text-purple-600">Free</div>
```

```
  </div>
```

```
<div class="flex flex-wrap gap-4 mb-6">
```

```
  <div class="flex items-center text-gray-600">
```

```
    <i class="fas fa-calendar mr-2"></i>
```

```
    <span id="event-date">Date TBD</span>
```

```
  </div>
```

```
  <div class="flex items-center text-gray-600">
```

```
    <i class="fas fa-clock mr-2"></i>
```

```
    <span id="event-time">Time TBD</span>
```

```
  </div>
```

```
  <div class="flex items-center text-gray-600">
```

```
    <i class="fas fa-map-marker-alt mr-2"></i>
```

```
    <span id="event-location">Location TBD</span>
```

```
  </div>
```

```
  <div class="flex items-center text-gray-600">
```

```
    <i class="fas fa-users mr-2"></i>
```

```
    <span id="event-attendees">0</span> attending
```

```
  </div>
```

```

</div>

<p id="event-description" class="text-gray-700 mb-6">Event description will appear here</p>

<div class="flex justify-between items-center">
  <div class="flex items-center">
    <div class="w-10 h-10 rounded-full bg-gray-200 overflow-hidden mr-3">
      
    </div>
    <div>
      <p class="text-sm text-gray-500">Organized by</p>
      <p id="organizer-name" class="font-medium">Organizer Name</p>
    </div>
  </div>

  <button id="rsvp-button" class="btn-primary text-center px-8">RSVP Now</button>
</div>
</div>

<!-- Event Location Map -->
<div class="bg-white rounded-xl shadow-md p-6 mb-6">
  <h2 class="text-xl font-bold mb-4">Location</h2>
  <div id="event-map" class="h-64 rounded-lg bg-gray-200 mb-4"></div>
  <div class="flex justify-between items-center">
    <p id="event-address" class="text-gray-700">Full address will appear here.</p>
    <a id="directions-link" href="#" target="_blank" class="text-purple-600 hover:text-purple-500">
      <i class="fas fa-directions mr-1"></i> Get Directions
    </a>
  </div>
</div>
</div>

<!-- Vibe Checks Section -->
<div class="bg-white rounded-xl shadow-md p-6 mb-6">
  <div class="flex justify-between items-center mb-4">
    <h2 class="text-xl font-bold">Vibe Checks</h2>
    <button id="post-vibe-btn" class="text-sm btn-secondary">
      <i class="fas fa-plus mr-1"></i> Post Vibe Check
    </button>
  </div>
</div>

<div id="vibes-container">
  <!-- Vibe checks will be loaded here -->
  <div class="text-center py-8 hidden" id="no-vibes">
    <p class="text-gray-500">No vibe checks yet. Be the first to post!</p>
  </div>
</div>

```

```

        </div>
    </div>
</div>

<!-- Similar Events -->
<div class="bg-white rounded-xl shadow-md p-6">
    <h2 class="text-xl font-bold mb-4">You Might Also Like</h2>
    <div id="similar-events" class="grid grid-cols-1 md:grid-cols-3 gap-4">
        <!-- Similar events will be loaded here -->
    </div>
</div>
</div>

<!-- Error Message (Hidden by default) -->
<div id="event-error" class="hidden bg-white rounded-xl shadow-md p-6 text-center">
    <i class="fas fa-exclamation-circle text-red-500 text-4xl mb-4"></i>
    <h2 class="text-xl font-bold mb-2">Event Not Found</h2>
    <p class="text-gray-600 mb-6">The event you're looking for doesn't exist or has been re
    <a href="index.html" class="btn-primary inline-block">Discover Events</a>
</div>
</div>
</div>

<!-- RSVP Modal Container -->
<div id="modal-container"></div>

<!-- Leaflet Map JS -->
<script src="https://unpkg.com/leaflet@1.9.4/dist/leaflet.js"></script>

<!-- Scripts -->
<script type="module">
    import { loadComponent } from '../scripts/utils/component-loader.js';
    import errorHandler from '../scripts/utils/error-handler.js';
    import { formatEventDate, formatTime } from '../scripts/utils/date-format.js';
    import { toggleBookmark } from '../scripts/components/events.js';

    // Global variables
    let eventId = null;
    let eventData = null;
    let map = null;
    let userHasRSVPd = false;

    document.addEventListener('DOMContentLoaded', function() {
        // Load header component

```



```

loadComponent('header-container', '../components/header.html');

// Load RSVP modal
loadComponent('modal-container', '../components/modals/rsvp-modal.html');

// Get event ID from URL
const urlParams = new URLSearchParams(window.location.search);
eventId = urlParams.get('id');

if (!eventId) {
  showError('No event ID provided');
  return;
}

// Load event data
loadEventDetails(eventId);

// Set up share button
document.getElementById('share-event').addEventListener('click', shareEvent);

// Set up post vibe button
document.getElementById('post-vibe-btn').addEventListener('click', function() {
  import('../scripts/components/modals.js')
    .then(module => {
      // Pre-fill the event name in the vibe check modal
      const eventInput = document.querySelector('#vibeCheckModal .event-input');
      if (eventInput && eventData) {
        eventInput.value = eventData.title;
        eventInput.disabled = true; // Lock it to this event
      }

      module.openModal('vibeCheckModal');
    });
});
});

async function loadEventDetails(eventId) {
  try {
    const db = firebase.firestore();
    const doc = await db.collection('events').doc(eventId).get();

    if (!doc.exists) {
      showError('Event not found');
      return;
    }
  }
}

```

```

    }

    // Save event data
    eventData = {
      id: doc.id,
      ...doc.data()
    };

    // Populate UI with event data
    populateEventDetails(eventData);

    // Check if user has RSVP'd
    checkUserRSVP(eventId);

    // Load vibe checks for this event
    loadVibeChecks(eventId);

    // Load similar events
    loadSimilarEvents(eventData.category);

    // Hide loading, show details
    document.getElementById('event-loading').classList.add('hidden');
    document.getElementById('event-details').classList.remove('hidden');
  } catch (error) {
    console.error('Error loading event:', error);
    showError('Failed to load event details');
  }
}

function populateEventDetails(event) {
  // Set page title
  document.title = event.title + ' - The Play';

  // Set event details
  document.getElementById('event-title').textContent = event.title;
  document.getElementById('event-description').textContent = event.description;
  document.getElementById('event-price').textContent = event.price || 'Free';
  document.getElementById('event-score').textContent = event.score || '8.5';
  document.getElementById('event-attendees').textContent = event.attendees || '0';
  document.getElementById('event-location').textContent = event.location || 'TBD';
  document.getElementById('event-address').textContent = event.address || event.location ||

  // Set date and time
  if (event.dateTime) {

```

```

    const dateTime = event.dateTime.toDate ? event.dateTime.toDate() : new Date(event.dateTime.toDate());
    document.getElementById('event-date').textContent = formatEventDate(dateTime);
    document.getElementById('event-time').textContent = formatTime(dateTime);
}

// Set image
if (event.imageUrl) {
    document.getElementById('event-image').src = event.imageUrl;
}

// Set organizer info
if (event.createdBy) {
    loadOrganizerInfo(event.createdBy);
}

// Setup map
if (event.location) {
    setupMap(event.location, event.coordinates);
}

// Setup directions link
const directionsLink = document.getElementById('directions-link');
if (event.coordinates) {
    directionsLink.href = `https://www.google.com/maps/dir/?api=1&destination=${event.coordinates}`;
} else if (event.location) {
    directionsLink.href = `https://www.google.com/maps/dir/?api=1&destination=${encodeURIComponent(event.location.address)}`;
}

// Setup bookmark button
const bookmarkBtn = document.getElementById('bookmark-event');
checkIfBookmarked(event.id).then(isBookmarked => {
    if (isBookmarked) {
        bookmarkBtn.querySelector('i').classList.add('text-yellow-500');
    }
});

bookmarkBtn.addEventListener('click', function() {
    toggleBookmark(event.id, this);
});

// Setup RSVP button
document.getElementById('rsvp-button').addEventListener('click', function() {
    openRSVPModal(event);
});

```

```
}
```

```
async function loadOrganizerInfo(organizerId) {
  try {
    const db = firebase.firestore();
    const doc = await db.collection('users').doc(organizerId).get();

    if (doc.exists) {
      const organizer = doc.data();
      document.getElementById('organizer-name').textContent = organizer.displayName || 'Eve';

      if (organizer.photoURL) {
        document.getElementById('organizer-photo').src = organizer.photoURL;
      }
    }
  } catch (error) {
    console.error('Error loading organizer info:', error);
  }
}
```

```
function setupMap(location, coordinates) {
  // Initialize map
  map = L.map('event-map').setView([29.4252, -98.4946], 13); // Default to San Antonio

  // Add tile layer
  L.tileLayer('https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png', {
    attribution: '&copy; <a href="https://www.openstreetmap.org/copyright">OpenStreetMap</a>',
  }).addTo(map);

  // Add marker
  if (coordinates) {
    // If we have exact coordinates
    L.marker([coordinates.lat, coordinates.lng]).addTo(map);
    map.setView([coordinates.lat, coordinates.lng], 15);
  } else {
    // If we only have a text location, geocode it
    geocodeLocation(location);
  }
}
```

```
async function geocodeLocation(locationString) {
  try {
    const response = await fetch(`https://nominatim.openstreetmap.org/search?format=json&q=${locationString}`);
    const data = await response.json();
  } catch (error) {
    console.error('Error geocoding location:', error);
  }
}
```

```

if (data.length > 0) {
  const lat = parseFloat(data[0].lat);
  const lon = parseFloat(data[0].lon);

  L.marker([lat, lon]).addTo(map);
  map.setView([lat, lon], 15);

  // Save coordinates to event document for future use
  if (eventId) {
    const db = firebase.firestore();
    db.collection('events').doc(eventId).update({
      coordinates: {
        lat: lat,
        lng: lon
      }
    }).catch(error => {
      console.error('Error updating coordinates:', error);
    });
  }
} catch (error) {
  console.error('Error geocoding location:', error);
}

}

async function checkIfBookmarked(eventId) {
  const user = firebase.auth().currentUser;
  if (!user) return false;

  try {
    const db = firebase.firestore();
    const snapshot = await db.collection('bookmarks')
      .where('userId', '==', user.uid)
      .where('eventId', '==', eventId)
      .get();

    return !snapshot.empty;
  } catch (error) {
    console.error('Error checking bookmark status:', error);
    return false;
  }
}

```

```

async function checkUserRSVP(eventId) {
  const user = firebase.auth().currentUser;
  if (!user) return;

  try {
    const db = firebase.firestore();
    const snapshot = await db.collection('attendees')
      .where('userId', '==', user.uid)
      .where('eventId', '==', eventId)
      .get();

    userHasRSVPd = !snapshot.empty;

    // Update RSVP button
    const rsvpButton = document.getElementById('rsvp-button');
    if (userHasRSVPd) {
      rsvpButton.textContent = 'Cancel RSVP';
      rsvpButton.classList.remove('btn-primary');
      rsvpButton.classList.add('btn-secondary');
    } else {
      rsvpButton.textContent = 'RSVP Now';
      rsvpButton.classList.add('btn-primary');
      rsvpButton.classList.remove('btn-secondary');
    }
  } catch (error) {
    console.error('Error checking RSVP status:', error);
  }
}

async function loadVibeChecks(eventId) {
  try {
    const db = firebase.firestore();
    const snapshot = await db.collection('vibeChecks')
      .where('eventId', '==', eventId)
      .orderBy('timestamp', 'desc')
      .limit(10)
      .get();

    const vibesContainer = document.getElementById('vibes-container');
    const noVibes = document.getElementById('no-vibes');

    if (snapshot.empty) {
      noVibes.classList.remove('hidden');
      return;
    }
  }
}

```

```

    }

    noVibes.classList.add('hidden');

    // Clear container except for the "no vibes" message
    Array.from(vibesContainer.children).forEach(child => {
        if (child.id !== 'no-vibes') {
            vibesContainer.removeChild(child);
        }
    });

    // Add vibe checks
    snapshot.forEach(doc => {
        const vibe = doc.data();
        const vibeElement = createVibeElement(doc.id, vibe);
        vibesContainer.appendChild(vibeElement);
    });
} catch (error) {
    console.error('Error loading vibe checks:', error);
}
}

function createVibeElement(id, vibe) {
    const div = document.createElement('div');
    div.className = 'border-b border-gray-200 py-4 last:border-0';

    // Format timestamp
    let timestampStr = 'Just now';
    if (vibe.timestamp) {
        const timestamp = vibe.timestamp.toDate ? vibe.timestamp.toDate() : new Date(vibe.timestamp);
        const now = new Date();
        const diffMs = now - timestamp;
        const diffMins = Math.floor(diffMs / 60000);

        if (diffMins < 60) {
            timestampStr = `${diffMins} min ago`;
        } else if (diffMins < 1440) {
            timestampStr = `${Math.floor(diffMins / 60)} hr ago`;
        } else {
            timestampStr = timestamp.toLocaleDateString();
        }
    }

    div.innerHTML = `

```

```

<div class="flex items-start">
  <div class="w-10 h-10 rounded-full bg-gray-200 overflow-hidden mr-3">
    
  <div class="flex-1">
    <div class="flex justify-between items-center mb-1">
      <p class="font-medium">${vibe.userDisplayName || 'User'}</p>
      <p class="text-xs text-gray-500">${timestampStr}</p>
    </div>
    <p class="text-gray-700 mb-3">${vibe.description || ''}</p>
    ${vibe.mediaUrls && vibe.mediaUrls.length ? `
      <div class="grid grid-cols-2 gap-2">
        ${vibe.mediaUrls.map(url => `
          <div class="bg-gray-100 rounded-lg overflow-hidden h-32">
            
          </div>
        `).join('')}
      </div>
    ` : ''}
  </div>
</div>
`;

return div;
}

```

```

async function loadSimilarEvents(category) {
  try {
    const db = firebase.firestore();
    const snapshot = await db.collection('events')
      .where('category', '==', category)
      .where(firebase.firestore.FieldPath.documentId(), '!=', eventId)
      .limit(3)
      .get();

    const container = document.getElementById('similar-events');
    container.innerHTML = '';

    if (snapshot.empty) {
      container.innerHTML = '<p class="col-span-3 text-center text-gray-500 py-8">No simila
      return;
    }

    snapshot.forEach(doc => {

```



```

const event = doc.data();
event.id = doc.id;

const eventElement = document.createElement('div');
eventElement.className = 'bg-gray-50 rounded-lg overflow-hidden hover:shadow-md trans

// Format date
let dateStr = 'Date TBD';
if (event.dateTime) {
  const dateTime = event.dateTime.toDate ? event.dateTime.toDate() : new Date(event.c
  dateStr = formatEventDate(dateTime);
}

eventElement.innerHTML = `
  <div class="h-32 bg-gray-200 relative">
    
      <i class="fas fa-fire mr-1"></i> ${event.score || '8.5'}
    </div>
  </div>
  <div class="p-3">
    <h3 class="font-bold text-sm line-clamp-1">${event.title}</h3>
    <p class="text-xs text-gray-600 mb-1">${dateStr}</p>
    <p class="text-xs text-gray-600 line-clamp-1">
      <i class="fas fa-map-marker-alt mr-1"></i> ${event.location || 'Location TBD'}
    </p>
  </div>
`;

eventElement.addEventListener('click', function() {
  window.location.href = `event-detail.html?id=${event.id}`;
});

container.appendChild(eventElement);
});
} catch (error) {
  console.error('Error loading similar events:', error);
}
}

function openRSVPModal(event) {
  if (userHasRSVPd) {
    // User already RSVP'd, ask if they want to cancel
    if (confirm('Are you sure you want to cancel your RSVP?')) {

```

```

        cancelRSVP(event.id);
    }
    return;
}

// Open RSVP modal
import('.././scripts/components/modals.js')
    .then(module => {
        // Set event data in the modal
        const eventTitleElement = document.querySelector('#rsvpModal .event-title');
        const eventDateElement = document.querySelector('#rsvpModal .event-date');
        const eventLocationElement = document.querySelector('#rsvpModal .event-location');

        if (eventTitleElement) eventTitleElement.textContent = event.title;
        if (eventDateElement && event.dateTime) {
            const dateTime = event.dateTime.toDate ? event.dateTime.toDate() : new Date(event.c
            eventDateElement.textContent = formatEventDate(dateTime);
        }
        if (eventLocationElement) eventLocationElement.textContent = event.location || 'Locat

        // Store event ID in the modal for RSVP submission
        const modalElement = document.getElementById('rsvpModal');
        if (modalElement) modalElement.setAttribute('data-event-id', event.id);

        // Open the modal
        module.openModal('rsvpModal');
    });
}

async function cancelRSVP(eventId) {
    const user = firebase.auth().currentUser;
    if (!user) {
        errorHandler.showToast('You must be logged in to cancel an RSVP', 'error');
        return;
    }

    try {
        const db = firebase.firestore();

        // Find and delete the attendance record
        const snapshot = await db.collection('attendees')
            .where('userId', '==', user.uid)
            .where('eventId', '==', eventId)
            .get();
    }

```

```

const batch = db.batch();

snapshot.forEach(doc => {
  batch.delete(doc.ref);
});

// Decrement attendee count
const eventRef = db.collection('events').doc(eventId);
batch.update(eventRef, {
  attendees: firebase.firestore.FieldValue.increment(-1)
});

await batch.commit();

// Update UI
userHasRSVPd = false;
const rsvpButton = document.getElementById('rsvp-button');
rsvpButton.textContent = 'RSVP Now';
rsvpButton.classList.add('btn-primary');
rsvpButton.classList.remove('btn-secondary');

// Decrement the displayed attendee count
const attendeesElement = document.getElementById('event-attendees');
const currentCount = parseInt(attendeesElement.textContent);
attendeesElement.textContent = Math.max(0, currentCount - 1);

// Show success message
errorHandler.showToast('Your RSVP has been canceled', 'success');
} catch (error) {
  console.error('Error canceling RSVP:', error);
  errorHandler.showToast('Failed to cancel RSVP', 'error');
}
}

function shareEvent() {
  // Build share data
  const shareData = {
    title: document.getElementById('event-title').textContent,
    text: document.getElementById('event-description').textContent,
    url: window.location.href
  };

  // Check if Web Share API is supported

```

```
if (navigator.share && typeof navigator.share === 'function') {
  navigator.share(shareData)
    .then(() => console.log('Shared successfully'))
    .catch(error => {
      console.error('Error sharing:', error);
      // Fall back to clipboard copy
      copyToClipboard();
    });
} else {
  // Fall back to clipboard copy
  copyToClipboard();
}
}
```

```
function copyToClipboard() {
  // Create a temporary input
  const input = document.createElement('input');
  input.value = window.location.href;
  document.body.appendChild(input);
  input.select();
  document.execCommand('copy');
  document.body.removeChild(input);
}
```