






```
.marker-art { border-color:  #3B82F6; color:  #3B82F6; }  
.marker-food { border-color:  #10B981; color:  #10B981; }  
.marker-nightlife { border-color:  #EC4899; color:  #EC4899; }  
.marker-sports { border-color:  #F59E0B; color:  #F59E0B; }  
.marker-community { border-color:  #8B5CF6; color:  #8B5CF6; }  
.marker-other { border-color:  #6B7280; color:  #6B7280; }  
</style>
```

```
</head> <body class="bg-gray-100"> <!-- Header Container --> <div id="header-container"> </div>  
<div class="flex"> <!-- Sidebar (hidden on mobile) --> <div class="hidden lg:block lg:w-1/4 xl:w-1/5 bg-  
white border-r border-gray-200"> <div class="p-4 map-sidebar"> <h2 class="text-xl font-bold mb-  
4">Filters</h2>
```

```
<!-- Category Filter -->
```

```
<div class="mb-6">
```

```
  <h3 class="text-sm font-medium mb-2">Event Category</h3>
```

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

```
    <label class="flex items-center">
```

```
      <input type="checkbox" class="category-filter mr-2" value="music" checked>
```

```
      <span class="text-sm">Music</span>
```

```
    </label>
```

```
    <label class="flex items-center">
```

```
      <input type="checkbox" class="category-filter mr-2" value="art" checked>
```

```
      <span class="text-sm">Art & Culture</span>
```

```
    </label>
```

```
    <label class="flex items-center">
```

```
      <input type="checkbox" class="category-filter mr-2" value="food" checked>
```

```
      <span class="text-sm">Food & Drink</span>
```

```
    </label>
```

```
    <label class="flex items-center">
```

```
      <input type="checkbox" class="category-filter mr-2" value="nightlife" checked>
```

```
      <span class="text-sm">Nightlife</span>
```

```
    </label>
```

```
    <label class="flex items-center">
```

```
      <input type="checkbox" class="category-filter mr-2" value="sports" checked>
```

```
      <span class="text-sm">Sports & Fitness</span>
```

```
    </label>
```

```
    <label class="flex items-center">
```

```
      <input type="checkbox" class="category-filter mr-2" value="community" checked>
```

```
      <span class="text-sm">Community</span>
```

```
    </label>
```

```
    <label class="flex items-center">
```

```
      <input type="checkbox" class="category-filter mr-2" value="other" checked>
```

```
      <span class="text-sm">Other</span>
```

```
    </label>
```

```
  </div>
```

```
</div>
```

```
<!-- Date Filter -->
```

```
<div class="mb-6">
```

```
  <h3 class="text-sm font-medium mb-2">Date Range</h3>
```

```
  <div class="space-y-3">
```

```
    <label class="block">
```

```
      <span class="text-xs text-gray-600">Start Date</span>
```

```
      <input type="date" id="filter-start-date" class="w-full mt-1 p-2 border border-
```

```
gray-300 rounded-lg">
```

```

    </label>
    <label class="block">
      <span class="text-xs text-gray-600">End Date</span>
      <input type="date" id="filter-end-date" class="w-full mt-1 p-2 border border-gray-
300 rounded-lg">
    </label>
  </div>
</div>

<!-- Price Filter -->
<div class="mb-6">
  <h3 class="text-sm font-medium mb-2">Price</h3>
  <div class="space-y-2">
    <label class="flex items-center">
      <input type="checkbox" id="filter-free" class="mr-2" checked>
      <span class="text-sm">Free Events</span>
    </label>
    <label class="flex items-center">
      <input type="checkbox" id="filter-paid" class="mr-2" checked>
      <span class="text-sm">Paid Events</span>
    </label>
  </div>
</div>

<!-- Apply Filters Button -->
<button id="apply-filters" class="btn-primary w-full">Apply Filters</button>

<!-- Reset Filters Button -->
<button id="reset-filters" class="mt-2 w-full py-2 px-4 border border-gray-300 rounded-
lg text-gray-700 hover:bg-gray-50">Reset Filters</button>

<div class="border-t border-gray-200 my-6 pt-6">
  <h2 class="text-xl font-bold mb-4">Events List</h2>
  <div id="events-list" class="space-y-4">
    <!-- Events will be dynamically added here -->
    <div id="events-loading" class="text-center py-8">
      <div class="animate-spin rounded-full h-8 w-8 border-t-2 border-b-2 border-primary
mx-auto mb-2"></div>
      <p class="text-gray-600">Loading events...</p>
    </div>
    <div id="no-events" class="hidden text-center py-8">
      <p class="text-gray-600">No events found. Try adjusting your filters.</p>
    </div>
  </div>
</div>

```

```

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

<!-- Mobile Filter Panel (hidden by default) -->
<div id="mobile-filter-panel" class="fixed inset-0 bg-black bg-opacity-50 z-50 hidden">
  <div class="bg-white rounded-xl m-4 max-w-sm relative">
    <div class="p-4 border-b border-gray-200">
      <div class="flex justify-between items-center">
        <h2 class="text-xl font-bold">Filters</h2>
        <button id="close-filters" class="text-gray-500 hover:text-gray-700">
          <i class="fas fa-times"></i>
        </button>
      </div>
    </div>
  </div>

  <div class="p-4 overflow-y-auto" style="max-height: 70vh;">
    <!-- Category Filter -->
    <div class="mb-6">
      <h3 class="text-sm font-medium mb-2">Event Category</h3>
      <div class="space-y-2">
        <label class="flex items-center">
          <input type="checkbox" class="mobile-category-filter mr-2" value="music"
checked>
          <span class="text-sm">Music</span>
        </label>
        <label class="flex items-center">
          <input type="checkbox" class="mobile-category-filter mr-2" value="art" checked>
          <span class="text-sm">Art & Culture</span>
        </label>
        <label class="flex items-center">
          <input type="checkbox" class="mobile-category-filter mr-2" value="food" checked>
          <span class="text-sm">Food & Drink</span>
        </label>
        <label class="flex items-center">
          <input type="checkbox" class="mobile-category-filter mr-2" value="nightlife"
checked>
          <span class="text-sm">Nightlife</span>
        </label>
        <label class="flex items-center">
          <input type="checkbox" class="mobile-category-filter mr-2" value="sports"
checked>
          <span class="text-sm">Sports & Fitness</span>
        </label>

```

```

        <label class="flex items-center">
            <input type="checkbox" class="mobile-category-filter mr-2" value="community"
checked>
            <span class="text-sm">Community</span>
        </label>
        <label class="flex items-center">
            <input type="checkbox" class="mobile-category-filter mr-2" value="other"
checked>
            <span class="text-sm">Other</span>
        </label>
    </div>
</div>

<!-- Mobile Date Filter -->
<div class="mb-6">
    <h3 class="text-sm font-medium mb-2">Date Range</h3>
    <div class="space-y-3">
        <label class="block">
            <span class="text-xs text-gray-600">Start Date</span>
            <input type="date" id="mobile-filter-start-date" class="w-full mt-1 p-2 border
border-gray-300 rounded-lg">
        </label>
        <label class="block">
            <span class="text-xs text-gray-600">End Date</span>
            <input type="date" id="mobile-filter-end-date" class="w-full mt-1 p-2 border
border-gray-300 rounded-lg">
        </label>
    </div>
</div>

<!-- Mobile Price Filter -->
<div class="mb-6">
    <h3 class="text-sm font-medium mb-2">Price</h3>
    <div class="space-y-2">
        <label class="flex items-center">
            <input type="checkbox" id="mobile-filter-free" class="mr-2" checked>
            <span class="text-sm">Free Events</span>
        </label>
        <label class="flex items-center">
            <input type="checkbox" id="mobile-filter-paid" class="mr-2" checked>
            <span class="text-sm">Paid Events</span>
        </label>
    </div>
</div>

```

```
</div>
```

```
<div class="p-4 border-t border-gray-200">
```

```
  <button id="mobile-apply-filters" class="btn-primary w-full mb-2">Apply  
Filters</button>
```

```
  <button id="mobile-reset-filters" class="w-full py-2 px-4 border border-gray-300  
rounded-lg text-gray-700 hover:bg-gray-50">Reset Filters</button>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
<!-- Map Container -->
```

```
<div class="w-full lg:w-3/4 xl:w-4/5 relative">
```

```
  <!-- Mobile Filter Button -->
```

```
  <button id="show-filters" class="lg:hidden absolute top-4 left-4 bg-white p-3 rounded-lg  
shadow-md z-10">
```

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

```
  </button>
```

```
  <!-- Mobile List Button -->
```

```
  <button id="show-list" class="lg:hidden absolute top-4 right-4 bg-white p-3 rounded-lg  
shadow-md z-10">
```

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

```
  </button>
```

```
<!-- Map -->
```

```
<div id="map" class="map-container"></div>
```

```
</div>
```

```
</div> <!-- Mobile Events List (hidden by default) --> <div id="mobile-events-list" class="fixed inset-0  
bg-white z-50 hidden overflow-y-auto"> <div class="sticky top-0 bg-white border-b border-gray-200 p-  
4 flex justify-between items-center"> <h2 class="text-xl font-bold">Events</h2> <button id="close-list"  
class="text-gray-500 hover:text-gray-700"> <i class="fas fa-times"></i> </button> </div>
```

```

<div id="mobile-events-container" class="p-4 space-y-4">
  <!-- Events will be dynamically added here -->
  <div id="mobile-events-loading" class="text-center py-8">
    <div class="animate-spin rounded-full h-8 w-8 border-t-2 border-b-2 border-primary mx-
auto mb-2"></div>
    <p class="text-gray-600">Loading events...</p>
  </div>
  <div id="mobile-no-events" class="hidden text-center py-8">
    <p class="text-gray-600">No events found. Try adjusting your filters.</p>
  </div>
</div>

```

```

</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 eventService from '../scripts/services/event-service.js'; import { formatEventDate } from
'../scripts/utils/date-format.js'; import errorHandler from '../scripts/utils/error-handler.js'; // Global
variables let map; let markers = []; let allEvents = []; let filteredEvents = [];
document.addEventListener('DOMContentLoaded', function() { // Load header component
loadComponent('header-container', '../components/header.html'); // Initialize map initMap(); // Set up
filters setupFilters(); // Load events loadEvents(); // Set up mobile UI setupMobileUI(); }); function initMap()
{ // Initialize map centered on San Antonio map = L.map('map').setView([29.4252, -98.4946], 12); // 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> contributors' }).addTo(map); //
Try to get user's location if (navigator.geolocation) { navigator.geolocation.getCurrentPosition( position
=> { // Center map on user's location map.setView([position.coords.latitude, position.coords.longitude],
13); // Add user marker L.marker([position.coords.latitude, position.coords.longitude], { icon: L.divIcon({
className: 'user-location-marker', html: '<div class="w-6 h-6 rounded-full bg-blue-500 border-2 border-
white"></div>', iconSize: [24, 24], iconAnchor: [12, 12] }) }).addTo(map) .bindPopup('Your Location')
.openPopup(); }, error => { console.warn('Geolocation error:', error); } ); } } async function loadEvents() { try
{ // Show loading states document.getElementById('events-loading').classList.remove('hidden');
document.getElementById('mobile-events-loading').classList.remove('hidden'); // Hide no events
messages document.getElementById('no-events').classList.add('hidden');
document.getElementById('mobile-no-events').classList.add('hidden'); // Get all events allEvents = await
eventService.getEvents({ dateRange: { start: new Date() } }, 100); // Apply filters applyFilters(); // Hide
loading states document.getElementById('events-loading').classList.add('hidden');
document.getElementById('mobile-events-loading').classList.add('hidden'); } catch (error) {
console.error('Error loading events:', error); errorHandler.showToast('Failed to load events', 'error'); // Hide
loading states document.getElementById('events-loading').classList.add('hidden');
document.getElementById('mobile-events-loading').classList.add('hidden'); } } function applyFilters() { //

```

```

Get filter values const categoryFilters = Array.from(document.querySelectorAll('.category-
filter:checked')).map(cb => cb.value); const startDate = document.getElementById('filter-start-date').value
? new Date(document.getElementById('filter-start-date').value) : null; const endDate =
document.getElementById('filter-end-date').value ? new Date(document.getElementById('filter-end-
date').value) : null; const showFree = document.getElementById('filter-free').checked; const showPaid =
document.getElementById('filter-paid').checked; // Filter events filteredEvents = allEvents.filter(event => {
// Category filter if (!categoryFilters.includes(event.category)) return false; // Date filter if (startDate) {
const eventDate = event.dateTime ? new Date(event.dateTime.seconds * 1000) : null; if (eventDate &&
eventDate < startDate) return false; } if (endDate) { const eventDate = event.dateTime ? new
Date(event.dateTime.seconds * 1000) : null; if (eventDate && eventDate > endDate) return false; } // Price
filter const isFree = !event.price || event.price === 'Free' || event.price === '$0'; if (isFree && !showFree)
return false; if (!isFree && !showPaid) return false; return true; }); // Update map and lists updateMap();
updateEventsList(); updateMobileEventsList(); } function updateMap() { // Remove existing markers
markers.forEach(marker => map.removeLayer(marker)); markers = []; // Add markers for filtered events
filteredEvents.forEach(event => { // Skip events without coordinates if (!event.coordinates &&
!event.location) return; // Get coordinates let coords; if (event.coordinates) { coords =
[event.coordinates.lat, event.coordinates.lng]; } else { // Geocode the location (for demonstration - in a
production app, you would do this once and store the result) geocodeLocation(event); return; } // Create
custom marker const marker = L.marker(coords, { icon: L.divIcon({ className: '', html: `<div
class="custom-marker marker-${event.category || 'other'}"> <i class="fas
${getCategoryIcon(event.category)}"> </i> </div>`, iconSize: [36, 36], iconAnchor: [18, 18] }) }).addTo(map);
// Create popup content const popupContent = document.createElement('div');
popupContent.className = 'event-popup-content'; // Format date let dateStr = 'Date TBD'; if
(event.dateTime) { const dateTime = new Date(event.dateTime.seconds * 1000); dateStr =
formatEventDate(dateTime); } popupContent.innerHTML = `<div class="bg-gray-200 h-32">  </div> <div class="p-3"> <h3 class="font-bold text-lg">${event.title}</h3> <p class="text-sm
text-gray-600 mb-1">${dateStr}</p> <p class="text-sm text-gray-600 mb-2"> <i class="fas fa-map-
marker-alt mr-1"> </i> ${event.location || 'Location TBD'} </p> <a href="event-detail.html?
id=${event.id}" class="btn-primary text-xs py-1 px-3 inline-block">View Details</a> </div> `; // Bind
popup to marker marker.bindPopup(popupContent, { className: 'event-popup', maxWidth: 250,
closeButton: false }); // Add marker to array markers.push(marker); }); // If no markers, show message if
(markers.length === 0) { errorHandler.showToast('No events found with the current filters', 'info'); } else if
(markers.length === 1) { // If only one marker, open its popup markers[0].openPopup(); } } function
updateEventsList() { const container = document.getElementById('events-list'); // Clear existing events
(except the loading and no events elements) Array.from(container.children).forEach(child => { if (child.id
!== 'events-loading' && child.id !== 'no-events') { container.removeChild(child); } }); // Show no events
message if no events if (filteredEvents.length === 0) { document.getElementById('no-

```



```

events').classList.remove('hidden'); return; } else { document.getElementById('no-
events').classList.add('hidden'); } // Add events to list filteredEvents.forEach(event => { const eventElement
= createEventListItem(event); container.appendChild(eventElement); }); } function
updateMobileEventsList() { const container = document.getElementById('mobile-events-container'); //
Clear existing events (except the loading and no events elements)
Array.from(container.children).forEach(child => { if (child.id !== 'mobile-events-loading' && child.id !==
'mobile-no-events') { container.removeChild(child); } }); // Show no events message if no events if
(filteredEvents.length === 0) { document.getElementById('mobile-no-events').classList.remove('hidden');
return; } else { document.getElementById('mobile-no-events').classList.add('hidden'); } // Add events to list
filteredEvents.forEach(event => { const eventElement = createEventListItem(event);
container.appendChild(eventElement); }); } function createEventListItem(event) { const div =
document.createElement('div'); div.className = 'bg-white rounded-lg shadow-sm overflow-hidden flex';
// Format date let dateStr = 'Date TBD'; if (event.dateTime) { const dateTime = new
Date(event.dateTime.seconds * 1000); dateStr = formatEventDate(dateTime); } div.innerHTML = `<div
class="w-20 h-20 bg-gray-200 flex-shrink-0">  </div> <div class="p-3 flex-grow"> <h3
class="font-medium 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> ${event.location || 'Location TBD'} </p> </div> `; // Add click event to navigate to event detail
div.addEventListener('click', function() { window.location.href = `event-detail.html?id=${event.id}`; }); // If
coordinates, add click event to center map on event if (event.coordinates) { const centerBtn =
document.createElement('button'); centerBtn.className = 'p-2 text-gray-600 hover:text-gray-900 flex-
shrink-0 self-center mr-2'; centerBtn.innerHTML = '<i class="fas fa-map-marker-alt"></i>';
centerBtn.addEventListener('click', function(e) { e.stopPropagation(); // Prevent parent click event // Center
map on event map.setView([event.coordinates.lat, event.coordinates.lng], 15); // Find and open the
marker popup markers.forEach(marker => { const markerLatLng = marker.getLatLng(); if
(markerLatLng.lat === event.coordinates.lat && markerLatLng.lng === event.coordinates.lng) {
marker.openPopup(); } }); // Close mobile events list document.getElementById('mobile-events-
list').classList.add('hidden'); }); div.appendChild(centerBtn); } return div; } async function
geocodeLocation(event) { try { const response = await
fetch(`https://nominatim.openstreetmap.org/search?
format=json&q=${encodeURIComponent(event.location + ', San Antonio, TX')}`); const data = await
response.json(); if (data.length > 0) { const lat = parseFloat(data[0].lat); const lon = parseFloat(data[0].lon);
// Save coordinates to event event.coordinates = { lat: lat, lng: lon }; // Add marker for event const marker
= L.marker([lat, lon], { icon: L.divIcon({ className: '', html: `<div class="custom-marker
marker-${event.category || 'other'}"><i class="fas ${getCategoryIcon(event.category)}"></i></div>`,
iconSize: [36, 36], iconAnchor: [18, 18] }) }).addTo(map); // Create popup content const popupContent =
document.createElement('div'); popupContent.className = 'event-popup-content'; // Format date let

```

```

dateStr = 'Date TBD'; if (event.dateTime) { const dateTime = new Date(event.dateTime.seconds * 1000);
dateStr = formatEventDate(dateTime); } popupContent.innerHTML = ` <div class="bg-gray-200 h-32">
 </div> <div class="p-3"> <h3 class="font-bold text-lg">${event.title}</h3> <p class="text-sm
text-gray-600 mb-1">${dateStr}</p> <p class="text-sm text-gray-600 mb-2"> <i class="fas fa-map-
marker-alt mr-1"> </i> ${event.location || 'Location TBD'} </p> <a href="event-detail.html?
id=${event.id}" class="btn-primary text-xs py-1 px-3 inline-block">View Details</a> </div> `; // Bind
popup to marker marker.bindPopup(popupContent, { className: 'event-popup', maxWidth: 250,
closeButton: false }); // Add marker to array markers.push(marker); // Update event in Firestore (if created
by the user) const currentUser = firebase.auth().currentUser; if (currentUser && event.createdBy ===
currentUser.uid) { const db = firebase.firestore(); db.collection('events').doc(event.id).update({ coordinates:
{ lat: lat, lng: lng }).catch(error => { console.error('Error updating coordinates:', error); }); } } catch (error) {
console.error('Error geocoding location:', error); } } function setupFilters() { // Set up filter buttons
document.getElementById('apply-filters').addEventListener('click', applyFilters);
document.getElementById('reset-filters').addEventListener('click', resetFilters);
document.getElementById('mobile-apply-filters').addEventListener('click', function() { // Sync mobile filters
to desktop filters syncMobileFilters(); applyFilters(); closeMobileFilters(); });
document.getElementById('mobile-reset-filters').addEventListener('click', resetMobileFilters); // Set default
dates const today = new Date(); const endOfMonth = new Date(today.getFullYear(), today.getMonth() +
1, 0); document.getElementById('filter-start-date').valueAsDate = today; document.getElementById('filter-
end-date').valueAsDate = endOfMonth; document.getElementById('mobile-filter-start-date').valueAsDate
= today; document.getElementById('mobile-filter-end-date').valueAsDate = endOfMonth; } function
resetFilters() { // Reset category filters document.querySelectorAll('.category-filter').forEach(cb => {
cb.checked = true; }); // Reset date filters const today = new Date(); const endOfMonth = new
Date(today.getFullYear(), today.getMonth() + 1, 0); document.getElementById('filter-start-
date').valueAsDate = today; document.getElementById('filter-end-date').valueAsDate = endOfMonth; //
Reset price filters document.getElementById('filter-free').checked = true; document.getElementById('filter-
paid').checked = true; // Apply filters applyFilters(); } function resetMobileFilters() { // Reset category filters
document.querySelectorAll('.mobile-category-filter').forEach(cb => { cb.checked = true; }); // Reset date
filters const today = new Date(); const endOfMonth = new Date(today.getFullYear(), today.getMonth() +
1, 0); document.getElementById('mobile-filter-start-date').valueAsDate = today;
document.getElementById('mobile-filter-end-date').valueAsDate = endOfMonth; // Reset price filters
document.getElementById('mobile-filter-free').checked = true; document.getElementById('mobile-filter-
paid').checked = true; } function syncMobileFilters() { // Sync category filters
document.querySelectorAll('.mobile-category-filter').forEach(mobileCb => { const desktopCb =
document.querySelector('.category-filter[value="${mobileCb.value}"]'); if (desktopCb) {
desktopCb.checked = mobileCb.checked; } }); // Sync date filters document.getElementById('filter-start-
date').value = document.getElementById('mobile-filter-start-date').value;

```

```
document.getElementById('filter-end-date').value = document.getElementById('mobile-filter-end-  
date').value; // Sync price filters document.getElementById('filter-free').checked =  
document.getElementById('mobile-filter-free').checked; document.getElementById('filter-paid').checked =  
document.getElementById('mobile-filter-paid').
```