📘 Project Code Documentation (Selected Files & Folders)

## global.css

/\* global.css \*/  
  
 /\* Color Variables \*/  
 :root {  
 --primary: #1e90ff;  
 --secondary: #22c55e;  
 --bg-light: #e0f7fa;  
 --bg-secondary: #d0f8ce;  
 --card-bg: rgba(255, 255, 255, 0.95);  
 --card-border: rgba(209, 213, 219, 0.6);  
 --text-main: #0f172a;  
 --text-muted: #64748b;  
 --text-light: #94a3b8;  
 --shadow-light: rgba(30, 144, 255, 0.08);  
 --shadow-medium: rgba(0, 0, 0, 0.08);  
 --shadow-strong: rgba(0, 0, 0, 0.15);  
 --radius: 12px;  
 --radius-medium: 10px;  
 --radius-small: 8px;  
 --transition: all 0.3s cubic-bezier(0.4, 0, 0.2, 1);  
 --border-subtle: rgba(0, 0, 0, 0.06);  
 --navbar-height: 80px;  
 }  
  
 /\* Reset \*/  
 \* {  
 margin: 0;  
 padding: 0;  
 box-sizing: border-box;  
 }  
  
 body {  
 font-family: 'Inter', sans-serif;  
 background: linear-gradient(135deg, var(--bg-light), var(--bg-secondary));  
 color: var(--text-main);  
 min-height: 100vh;  
 line-height: 1.6;  
 -webkit-font-smoothing: antialiased;  
 padding-top: var(--navbar-height);  
 }  
  
 /\* NAVBAR \*/  
 .fresh-navbar {  
 position: fixed;  
 top: 0;  
 left: 0;  
 right: 0;  
 height: var(--navbar-height);  
 z-index: 1000;  
 display: flex;  
 justify-content: space-between;  
 align-items: center;  
 padding: 12px 24px;  
 background: linear-gradient(135deg, rgba(224, 247, 250, 0.95), rgba(208, 248, 206, 0.95));  
 backdrop-filter: blur(12px);  
 border-bottom: 1px solid var(--border-subtle);  
 box-shadow: 0 1px 3px var(--shadow-light), 0 1px 2px var(--shadow-medium);  
 }  
  
 .logo {  
 font-weight: bold;  
 font-size: 20px;  
 background: linear-gradient(135deg, var(--primary), var(--secondary));  
 -webkit-background-clip: text;  
 -webkit-text-fill-color: transparent;  
 }  
  
 .nav-wrapper {  
 display: flex;  
 align-items: center;  
 }  
  
 .nav-links {  
 list-style: none;  
 display: flex;  
 gap: 8px;  
 align-items: center;  
 }  
  
 .nav-links li a {  
 padding: 10px 20px;  
 border-radius: 24px;  
 font-size: 14px;  
 font-weight: 500;  
 text-decoration: none;  
 color: var(--text-main);  
 background: var(--card-bg);  
 border: 1px solid var(--border-subtle);  
 transition: var(--transition);  
 box-shadow: 0 1px 3px var(--shadow-light);  
 }  
  
 .nav-links li a:hover {  
 transform: translateY(-1px);  
 box-shadow: 0 4px 12px var(--shadow-light);  
 background: linear-gradient(to right, #4fc3f7, #81c784);  
 color: white;  
 border-color: transparent;  
 }  
  
 .nav-links li a.active {  
 color: white;  
 background: linear-gradient(to right, #4fc3f7, #81c784);  
 font-weight: 600;  
 border-color: transparent;  
 box-shadow: 0 2px 8px rgba(30, 144, 255, 0.25);  
 }  
  
 .nav-links li a.logout {  
 background: rgba(239, 68, 68, 0.1);  
 color: #dc2626;  
 font-weight: 600;  
 border-color: rgba(239, 68, 68, 0.2);  
 }  
  
 .nav-links li a.logout:hover {  
 background: #dc2626;  
 color: white;  
 border-color: #dc2626;  
 }  
  
 /\* SPORTS SCROLL \*/  
 .sports-scroll {  
 display: flex;  
 gap: 12px;  
 overflow-x: auto;  
 padding: 16px 24px;  
 background: linear-gradient(135deg, rgba(224, 247, 250, 0.4), rgba(208, 248, 206, 0.4));  
 scrollbar-width: thin;  
 scrollbar-color: var(--primary) transparent;  
 }  
  
 .sports-scroll::-webkit-scrollbar {  
 height: 6px;  
 }  
  
 .sports-scroll::-webkit-scrollbar-thumb {  
 background: var(--primary);  
 border-radius: 3px;  
 }  
  
 /\* Make each sport-item clickable label \*/  
 .sport-item {  
 flex-shrink: 0;  
 position: relative;  
 display: inline-block;  
 cursor: pointer;  
 }  
  
 /\* Hide the checkbox input \*/  
 .sport-checkbox {  
 display: none;  
 }  
  
 /\* Base style for sport circle \*/  
 .sport-circle {  
 width: 70px;  
 height: 70px;  
 border-radius: 50%;  
 background: var(--card-bg);  
 font-size: 11px;  
 font-weight: 500;  
 text-align: center;  
 display: flex;  
 flex-direction: column;  
 align-items: center;  
 justify-content: center;  
 gap: 3px;  
 box-shadow: 0 2px 8px rgba(0,0,0,0.1);  
 transition: var(--transition);  
 border: 2px solid transparent;  
 }  
  
 /\* Hover effect \*/  
 .sport-circle:hover {  
 transform: translateY(-2px);  
 background: rgba(30, 144, 255, 0.05);  
 border-color: rgba(30, 144, 255, 0.2);  
 }  
  
 /\* Selected effect when checkbox is checked \*/  
 .sport-checkbox:checked + .sport-circle {  
 background: linear-gradient(135deg, var(--primary), var(--secondary));  
 color: white;  
 box-shadow: 0 4px 12px rgba(30, 144, 255, 0.3);  
 border-color: rgba(30, 144, 255, 0.4);  
 transform: translateY(-2px);  
 position: relative;  
 }  
  
 /\* Add check mark icon \*/  
 .sport-checkbox:checked + .sport-circle::after {  
 content: "✓";  
 position: absolute;  
 top: 1px;  
 right: 25px;  
 font-size: 16px;  
 font-weight: bold;  
 color: white;  
 }

————————————————————————————————————————

## navbar.php

<?php  
/\*\*  
 \* Player Navbar Component  
 \* For Player only with active page conditions  
 \*/  
  
$current\_page = basename($\_SERVER['PHP\_SELF'], '.php');  
$current\_folder = basename(dirname($\_SERVER['PHP\_SELF']));  
  
// Determine active page based on conditions  
$active\_page = '';  
  
if ($current\_page == 'HomePage' || $current\_folder == 'HomePage') {  
 $active\_page = 'HomePage';  
} elseif ($current\_page == 'FindPlayer' || $current\_folder == 'FindPlayer') {  
 $active\_page = 'FindPlayer';  
} elseif ($current\_page == 'JoinGroup' || $current\_folder == 'JoinGroup') {  
 $active\_page = 'JoinGroup';  
} elseif ($current\_page == 'BookVenue' || $current\_folder == 'BookVenue') {  
 $active\_page = 'BookVenue';  
} elseif ($current\_page == 'Favorites' || $current\_folder == 'Favorites') {  
 $active\_page = 'Favorites';  
} elseif ($current\_page == 'Chats' || $current\_folder == 'Chats') {  
 $active\_page = 'Chats';  
} elseif ($current\_page == 'MyBookings' || $current\_folder == 'MyBookings') {  
 $active\_page = 'MyBookings';  
}  
  
// Function to check if page is active  
function isActive($pageName) {  
 global $active\_page;  
 return ($active\_page === $pageName) ? 'active' : '';  
}  
?>  
  
<!-- Player Navbar -->  
<link rel="stylesheet" href="../global.css">  
<nav class="fresh-navbar">  
 <div class="logo">Book&Play</div>  
 <div class="nav-wrapper">  
 <ul class="nav-links">  
 <li><a href="../HomePage/HomePage.php" class="<?php echo isActive('HomePage'); ?>">Home Page</a></li>  
 <li><a href="../FindPlayer/FindPlayer.php" class="<?php echo isActive('FindPlayer'); ?>">Find Player</a></li>  
 <li><a href="../JoinGroup/JoinGroup.php" class="<?php echo isActive('JoinGroup'); ?>">Join Group</a></li>  
 <li><a href="../BookVenue/BookVenue.php" class="<?php echo isActive('BookVenue'); ?>">Book Venue</a></li>  
 <li><a href="../Favorites/Favorites.php" class="<?php echo isActive('Favorites'); ?>">My Favorites</a></li>  
 <li><a href="../Chats/Chats.php" class="<?php echo isActive('Chats'); ?>">My Chats</a></li>  
 <li><a href="../MyBookings/MyBookings.php" class="<?php echo isActive('MyBookings'); ?>">My Bookings</a></li>  
 <li><a href="../../auth/Login\_Page/Login.php" class="logout">Logout</a></li>  
 </ul>  
 </div>  
</nav>  
  
<!-- Debug Information -->  
<script>  
console.log('=== PLAYER NAVBAR DEBUG ===');  
console.log('Current Page:', '<?php echo $current\_page; ?>');  
console.log('Current Folder:', '<?php echo $current\_folder; ?>');  
console.log('Active Page:', '<?php echo $active\_page; ?>');  
</script>

————————————————————————————————————————

## sportsScroll.html

<section class="sports-scroll" id="sportsContainer"></section>  
  
<script>  
document.addEventListener("DOMContentLoaded", () => {  
 fetch("fetch\_sports.php")  
 .then(res => res.json())  
 .then(data => {  
 if (data.success) {  
 const container = document.getElementById("sportsContainer");  
 container.innerHTML = "";  
  
 data.sports.forEach(sport => {  
 const label = document.createElement("label");  
 label.className = "sport-item";  
  
 const input = document.createElement("input");  
 input.type = "checkbox";  
 input.className = "sport-checkbox";  
  
 const circle = document.createElement("div");  
 circle.className = "sport-circle";  
 circle.textContent = sport;  
 circle.dataset.sport = sport;  
  
 input.addEventListener("change", () => {  
 const checked = document.querySelectorAll(".sport-checkbox:checked");  
 const selectedSports = Array.from(checked).map(cb => cb.nextElementSibling.dataset.sport);  
 filterVenuesBySports(selectedSports);  
 });  
  
 label.appendChild(input);  
 label.appendChild(circle);  
 container.appendChild(label);  
 });  
 }  
 })  
 .catch(err => console.error("Failed to fetch sports", err));  
});  
</script>

————————————————————————————————————————

## BookingDetails\BookingDetails.css

/\* BookingDetails.css - FreshGlow UI Style \*/  
  
/\* Color Variables \*/  
:root {  
 --primary: #1e90ff;  
 --secondary: #22c55e;  
 --bg-light: #e0f7fa;  
 --bg-secondary: #d0f8ce;  
 --card-bg: rgba(255, 255, 255, 0.95);  
 --card-border: rgba(209, 213, 219, 0.6);  
 --text-main: #0f172a;  
 --text-muted: #64748b;  
 --text-light: #94a3b8;  
 --shadow-light: rgba(30, 144, 255, 0.08);  
 --shadow-medium: rgba(0, 0, 0, 0.08);  
 --shadow-strong: rgba(0, 0, 0, 0.15);  
 --radius: 12px;  
 --radius-medium: 10px;  
 --radius-small: 8px;  
 --transition: all 0.3s cubic-bezier(0.4, 0, 0.2, 1);  
 --border-subtle: rgba(0, 0, 0, 0.06);  
 --accent-orange: #ff6b35;  
 --accent-red: #ff4757;  
 --accent-green: #4CAF50;  
 --warning: #ff9800;  
 --purple: #9c27b0;  
}  
  
/\* Reset and Base Styles \*/  
\* {  
 margin: 0;  
 padding: 0;  
 box-sizing: border-box;  
}  
  
body {  
 font-family: 'Inter', sans-serif;  
 line-height: 1.6;  
 color: var(--text-main);  
 background: linear-gradient(135deg, var(--bg-light), var(--bg-secondary));  
 min-height: 100vh;  
 -webkit-font-smoothing: antialiased;  
}  
  
/\* Venue Section \*/  
.venue-section {  
 background: var(--card-bg);  
 margin: 24px;  
 border-radius: var(--radius);  
 overflow: hidden;  
 box-shadow: 0 4px 12px var(--shadow-light), 0 2px 4px var(--shadow-medium);  
 border: 1px solid var(--card-border);  
}  
  
.venue-banner {  
 position: relative;  
 height: 280px;  
 overflow: hidden;  
}  
  
.back-btn {  
 position: absolute;  
 top: 20px;  
 left: 20px;  
 z-index: 10;  
 background: rgba(255, 255, 255, 0.9);  
 border: none;  
 border-radius: var(--radius-small);  
 padding: 10px 16px;  
 font-size: 14px;  
 font-weight: 600;  
 color: var(--text-main);  
 cursor: pointer;  
 transition: var(--transition);  
 backdrop-filter: blur(10px);  
 box-shadow: 0 2px 8px rgba(0, 0, 0, 0.1);  
}  
  
.back-btn:hover {  
 background: rgba(255, 255, 255, 1);  
 transform: translateY(-1px);  
 box-shadow: 0 4px 12px rgba(0, 0, 0, 0.15);  
}  
  
.venue-image {  
 width: 100%;  
 height: 100%;  
 object-fit: cover;  
}  
  
.venue-overlay {  
 position: absolute;  
 bottom: 0;  
 left: 0;  
 right: 0;  
 background: linear-gradient(transparent, rgba(15, 23, 42, 0.85));  
 color: white;  
 padding: 40px 24px 24px;  
}  
  
.venue-name {  
 font-size: 2.25rem;  
 font-weight: 700;  
 margin-bottom: 8px;  
 text-shadow: 0 2px 4px rgba(0, 0, 0, 0.3);  
}  
  
.venue-location {  
 font-size: 1.125rem;  
 opacity: 0.9;  
 font-weight: 500;  
}  
.venue-sport {  
 font-size: 1rem;  
 color: rgba(255, 255, 255, 0.9);  
 font-weight: 500;  
 text-transform: capitalize;  
 margin-bottom: 8px;  
}  
  
.venue-rating {  
 display: flex;  
 align-items: center;  
 gap: 8px;  
}  
  
.rating-stars {  
 font-size: 1rem;  
 line-height: 1;  
}  
  
.rating-value {  
 font-size: 1rem;  
 color: rgba(255, 255, 255, 0.95);  
 font-weight: 600;  
}  
.booking-info {  
 padding: 24px;  
}  
  
.info-grid {  
 display: grid;  
 grid-template-columns: repeat(auto-fit, minmax(180px, 1fr));  
 gap: 16px;  
 margin-bottom: 20px;  
}  
  
.info-card {  
 background: rgba(224, 247, 250, 0.3);  
 padding: 20px;  
 border-radius: var(--radius-medium);  
 display: flex;  
 flex-direction: column;  
 align-items: center;  
 text-align: center;  
 transition: var(--transition);  
 border: 1px solid var(--border-subtle);  
}  
  
.info-card:hover {  
 transform: translateY(-2px);  
 box-shadow: 0 4px 12px var(--shadow-light);  
 background: rgba(30, 144, 255, 0.05);  
}  
  
.total-price {  
 background: linear-gradient(135deg, var(--primary), var(--secondary));  
 color: white;  
 border-color: transparent;  
}  
  
.total-price:hover {  
 background: linear-gradient(135deg, var(--primary), var(--secondary));  
 transform: translateY(-2px);  
 box-shadow: 0 4px 12px rgba(30, 144, 255, 0.25);  
}  
  
.privacy-card {  
 background: rgba(156, 39, 176, 0.1);  
 border: 1px solid rgba(156, 39, 176, 0.2);  
}  
  
.privacy-card:hover {  
 background: rgba(156, 39, 176, 0.15);  
 border-color: rgba(156, 39, 176, 0.3);  
}  
  
.info-label {  
 font-size: 0.875rem;  
 color: var(--text-muted);  
 margin-bottom: 4px;  
 font-weight: 500;  
}  
  
.total-price .info-label,  
.privacy-card .info-label {  
 color: rgba(255, 255, 255, 0.9);  
}  
  
.privacy-card .info-label {  
 color: var(--purple);  
}  
  
.info-value {  
 font-size: 1.25rem;  
 font-weight: 700;  
 color: var(--text-main);  
}  
  
.total-price .info-value {  
 color: white;  
 font-size: 1.375rem;  
}  
  
/\* Privacy Toggle \*/  
.privacy-toggle {  
 background: linear-gradient(135deg, var(--purple), #ba68c8);  
 color: white;  
 border: none;  
 padding: 8px 16px;  
 border-radius: 20px;  
 cursor: pointer;  
 font-weight: 600;  
 font-size: 0.875rem;  
 transition: var(--transition);  
 display: flex;  
 align-items: center;  
 gap: 6px;  
}  
  
.privacy-toggle:hover {  
 transform: translateY(-1px);  
 box-shadow: 0 4px 12px rgba(156, 39, 176, 0.3);  
}  
  
.privacy-toggle.private {  
 background: linear-gradient(135deg, var(--accent-red), #ff6b7a);  
}  
  
.privacy-toggle.private:hover {  
 box-shadow: 0 4px 12px rgba(255, 71, 87, 0.3);  
}  
  
/\* Password Section \*/  
.password-section {  
 margin-top: 20px;  
}  
  
.password-card {  
 background: linear-gradient(135deg, rgba(255, 71, 87, 0.1), rgba(255, 107, 122, 0.1));  
 border: 1px solid rgba(255, 71, 87, 0.2);  
 border-radius: var(--radius-medium);  
 padding: 20px;  
 text-align: center;  
}  
  
.password-card h3 {  
 color: var(--accent-red);  
 margin-bottom: 12px;  
 font-size: 1.125rem;  
}  
  
.password-display {  
 display: flex;  
 align-items: center;  
 justify-content: center;  
 gap: 12px;  
 margin-bottom: 8px;  
}  
  
.password-display code {  
 background: rgba(255, 255, 255, 0.9);  
 padding: 8px 16px;  
 border-radius: var(--radius-small);  
 font-family: 'Monaco', 'Consolas', monospace;  
 font-size: 1rem;  
 font-weight: 600;  
 color: var(--text-main);  
 border: 1px solid var(--border-subtle);  
}  
  
.copy-btn {  
 background: var(--accent-red);  
 color: white;  
 border: none;  
 padding: 8px 12px;  
 border-radius: var(--radius-small);  
 cursor: pointer;  
 font-size: 0.875rem;  
 font-weight: 600;  
 transition: var(--transition);  
}  
  
.copy-btn:hover {  
 background: #ff3838;  
 transform: translateY(-1px);  
}  
  
.password-note {  
 font-size: 0.875rem;  
 color: var(--text-muted);  
 font-style: italic;  
}  
  
/\* Players Section \*/  
.players-section {  
 background: var(--card-bg);  
 margin: 24px;  
 border-radius: var(--radius);  
 padding: 24px;  
 box-shadow: 0 4px 12px var(--shadow-light), 0 2px 4px var(--shadow-medium);  
 border: 1px solid var(--card-border);  
}  
  
.section-header {  
 display: flex;  
 justify-content: space-between;  
 align-items: center;  
 margin-bottom: 20px;  
 flex-wrap: wrap;  
 gap: 12px;  
}  
  
.section-title {  
 font-size: 1.5rem;  
 color: var(--text-main);  
 font-weight: 600;  
}  
  
.host-controls {  
 display: flex;  
 gap: 8px;  
 flex-wrap: wrap;  
}  
  
.edit-prices-btn,  
.save-prices-btn,  
.cancel-edit-btn {  
 background: linear-gradient(135deg, var(--primary), #4fc3f7);  
 color: white;  
 border: none;  
 padding: 8px 16px;  
 border-radius: 20px;  
 cursor: pointer;  
 font-weight: 600;  
 font-size: 0.875rem;  
 transition: var(--transition);  
}  
  
.save-prices-btn {  
 background: linear-gradient(135deg, var(--accent-green), #66bb6a);  
}  
  
.cancel-edit-btn {  
 background: linear-gradient(135deg, var(--accent-red), #ff6b7a);  
}  
  
.edit-prices-btn:hover,  
.save-prices-btn:hover,  
.cancel-edit-btn:hover {  
 transform: translateY(-1px);  
 box-shadow: 0 4px 12px rgba(30, 144, 255, 0.25);  
}  
  
.save-prices-btn:hover {  
 box-shadow: 0 4px 12px rgba(76, 175, 80, 0.25);  
}  
  
.cancel-edit-btn:hover {  
 box-shadow: 0 4px 12px rgba(255, 71, 87, 0.25);  
}  
  
/\* Horizontal Scroll Container \*/  
.players-scroll-container {  
 position: relative;  
 display: flex;  
 align-items: center;  
 gap: 16px;  
 width: 100%;  
 margin: 0 auto;  
}  
  
.players-horizontal-scroll {  
 display: flex;  
 gap: 20px;  
 overflow-x: auto;  
 overflow-y: hidden;  
 scroll-behavior: smooth;  
 padding: 12px 0;  
 flex: 1;  
 scrollbar-width: none;  
 -ms-overflow-style: none;  
 white-space: nowrap;  
 min-height: 300px;  
}  
  
.players-horizontal-scroll::-webkit-scrollbar {  
 display: none;  
}  
  
/\* Scroll Arrows \*/  
.scroll-arrow {  
 width: 50px;  
 height: 50px;  
 border-radius: 50%;  
 background: linear-gradient(135deg, var(--primary), var(--secondary));  
 color: white;  
 border: none;  
 cursor: pointer;  
 display: flex;  
 align-items: center;  
 justify-content: center;  
 font-size: 1.5rem;  
 font-weight: bold;  
 transition: var(--transition);  
 box-shadow: 0 4px 12px rgba(30, 144, 255, 0.3);  
 flex-shrink: 0;  
 z-index: 10;  
 position: relative;  
}  
  
.scroll-arrow:hover:not(:disabled) {  
 transform: translateY(-2px);  
 box-shadow: 0 6px 20px rgba(30, 144, 255, 0.4);  
}  
  
.scroll-arrow:active {  
 transform: translateY(0);  
}  
  
.scroll-arrow:disabled {  
 opacity: 0.3;  
 cursor: not-allowed;  
 transform: none;  
}  
  
.scroll-left span {  
 margin-left: -3px;  
}  
  
.scroll-right span {  
 margin-right: -3px;  
}  
  
/\* Player Cards in Horizontal Layout \*/  
.player-card {  
 background: rgba(224, 247, 250, 0.3);  
 border-radius: var(--radius-medium);  
 padding: 20px;  
 text-align: center;  
 position: relative;  
 transition: var(--transition);  
 border: 1px solid var(--border-subtle);  
 min-width: 200px;  
 max-width: 200px;  
 height: 260px;  
 flex-shrink: 0;  
 display: flex;  
 flex-direction: column;  
 justify-content: flex-start;  
 overflow: visible;  
}  
  
.player-card:hover {  
 transform: translateY(-3px);  
 box-shadow: 0 8px 20px var(--shadow-light);  
 background: rgba(30, 144, 255, 0.08);  
}  
  
.host-player {  
 background: linear-gradient(135deg, rgba(255, 215, 0, 0.2), rgba(255, 165, 0, 0.2));  
 border: 2px solid rgba(255, 140, 0, 0.4);  
}  
  
.host-player:hover {  
 background: linear-gradient(135deg, rgba(255, 215, 0, 0.25), rgba(255, 165, 0, 0.25));  
 border-color: rgba(255, 140, 0, 0.5);  
}  
  
.host-badge {  
 position: absolute;  
 top: -8px;  
 right: -8px;  
 background: linear-gradient(135deg, var(--accent-orange), #ff8c42);  
 color: white;  
 padding: 3px 8px;  
 border-radius: 12px;  
 font-size: 0.65rem;  
 font-weight: 700;  
 box-shadow: 0 3px 10px rgba(255, 107, 53, 0.4);  
}  
.me-player {  
 border: 2px solid #22c55e; /\* green border \*/  
 box-shadow: 0 0 15px #22c55e88; /\* green glow \*/  
 background-color: #ecfdf5; /\* soft green bg \*/  
}  
  
.me-badge {  
 position: absolute;  
 top: -8px;  
 right: -8px;  
 background: linear-gradient(135deg, #22c55e, #4ade80);  
 color: white;  
 padding: 3px 8px;  
 border-radius: 12px;  
 font-size: 0.65rem;  
 font-weight: 700;  
 box-shadow: 0 3px 10px rgba(34, 197, 94, 0.4);  
}  
  
.player-image {  
 width: 70px;  
 height: 70px;  
 border-radius: 50%;  
 object-fit: cover;  
 margin: 0 auto 12px auto;  
 cursor: pointer;  
 transition: var(--transition);  
 border: 2px solid var(--card-border);  
}  
  
.host-player .player-image {  
 border-color: rgba(255, 140, 0, 0.5);  
}  
  
.player-image:hover {  
 transform: scale(1.08);  
 box-shadow: 0 6px 15px var(--shadow-medium);  
}  
  
.player-name {  
 font-size: 1rem;  
 margin-bottom: 12px;  
 font-weight: 600;  
 color: var(--text-main);  
 white-space: nowrap;  
 overflow: hidden;  
 text-overflow: ellipsis;  
 line-height: 1.3;  
 height: auto;  
 min-height: 22px;  
 display: block;  
}  
  
.player-actions {  
 display: flex;  
 flex-direction: column;  
 gap: 8px;  
 margin-top: auto;  
}  
  
.switch-host-btn,  
.friend-btn {  
 border: none;  
 padding: 6px 12px;  
 border-radius: 14px;  
 cursor: pointer;  
 font-weight: 600;  
 font-size: 0.75rem;  
 transition: var(--transition);  
 white-space: nowrap;  
 overflow: hidden;  
 text-overflow: ellipsis;  
}  
  
.switch-host-btn {  
 background: linear-gradient(135deg, var(--primary), #4fc3f7);  
 color: white;  
}  
  
.switch-host-btn:hover:not(:disabled) {  
 transform: translateY(-1px);  
 box-shadow: 0 4px 10px rgba(30, 144, 255, 0.3);  
}  
  
.switch-host-btn:disabled {  
 background: var(--text-light);  
 cursor: not-allowed;  
 transform: none;  
}  
  
.friend-btn {  
 background: rgba(34, 197, 94, 0.1);  
 color: var(--secondary);  
 border: 1px solid rgba(34, 197, 94, 0.2);  
}  
  
.friend-btn:hover {  
 background: var(--secondary);  
 color: white;  
 transform: translateY(-1px);  
}  
  
.friend-btn.friends {  
 background: linear-gradient(135deg, var(--secondary), #66bb6a);  
 color: white;  
 border-color: transparent;  
}  
  
.friend-btn.friends:hover {  
 background: linear-gradient(135deg, #1db954, var(--secondary));  
}  
  
.friend-btn:disabled {  
 background: var(--text-light);  
 color: white;  
 cursor: not-allowed;  
 border-color: transparent;  
}  
  
.payment-amount-container {  
 margin-top: 6px;  
}  
  
.payment-amount {  
 background: linear-gradient(135deg, var(--accent-green), #66bb6a);  
 color: white;  
 padding: 6px 12px;  
 border-radius: 14px;  
 font-weight: 700;  
 font-size: 0.85rem;  
 display: inline-block;  
}  
  
.payment-input {  
 background: white;  
 border: 2px solid var(--primary);  
 padding: 6px 10px;  
 border-radius: 14px;  
 font-weight: 700;  
 font-size: 0.85rem;  
 text-align: center;  
 width: 70px;  
 outline: none;  
 transition: var(--transition);  
}  
  
.payment-input:focus {  
 border-color: var(--secondary);  
 box-shadow: 0 0 0 3px rgba(34, 197, 94, 0.1);  
}  
  
/\* Payment Section \*/  
.payment-section {  
 background: var(--card-bg);  
 margin: 24px;  
 border-radius: var(--radius);  
 padding: 24px;  
 box-shadow: 0 4px 12px var(--shadow-light), 0 2px 4px var(--shadow-medium);  
 border: 1px solid var(--card-border);  
}  
  
.payment-reminder {  
 background: linear-gradient(135deg, var(--accent-red), #ff6b7a);  
 border-radius: var(--radius-medium);  
 padding: 24px;  
 color: white;  
 text-align: center;  
}  
  
.timer-container {  
 display: flex;  
 align-items: center;  
 justify-content: center;  
 gap: 20px;  
}  
  
.timer-icon {  
 font-size: 2.5rem;  
}  
  
.timer-content {  
 flex: 1;  
 max-width: 600px;  
}  
  
.timer-title {  
 font-size: 1.375rem;  
 margin-bottom: 16px;  
 font-weight: 600;  
}  
  
.countdown-timer {  
 display: flex;  
 justify-content: center;  
 align-items: center;  
 gap: 8px;  
 margin-bottom: 16px;  
}  
  
.time-unit {  
 background: rgba(255, 255, 255, 0.2);  
 border-radius: var(--radius-small);  
 padding: 12px;  
 min-width: 70px;  
 display: flex;  
 flex-direction: column;  
 align-items: center;  
 backdrop-filter: blur(8px);  
}  
  
.time-value {  
 font-size: 1.75rem;  
 font-weight: 700;  
 line-height: 1;  
}  
  
.time-label {  
 font-size: 0.75rem;  
 opacity: 0.9;  
 margin-top: 4px;  
 font-weight: 500;  
}  
  
.time-separator {  
 font-size: 1.75rem;  
 font-weight: 700;  
 opacity: 0.8;  
}  
  
.timer-message {  
 font-size: 1rem;  
 line-height: 1.5;  
 opacity: 0.95;  
 font-weight: 500;  
}  
  
/\* Actions Section \*/  
.actions-section {  
 margin: 24px;  
 padding: 24px;  
}  
  
.action-buttons {  
 display: flex;  
 gap: 16px;  
 justify-content: center;  
}  
  
.cancel-btn, .pay-btn {  
 padding: 12px 32px;  
 border: none;  
 border-radius: 24px;  
 font-size: 1rem;  
 font-weight: 600;  
 cursor: pointer;  
 transition: var(--transition);  
 min-width: 140px;  
}  
  
.cancel-btn {  
 background: rgba(255, 71, 87, 0.1);  
 color: var(--accent-red);  
 border: 1px solid rgba(255, 71, 87, 0.2);  
}  
  
.cancel-btn:hover {  
 background: var(--accent-red);  
 color: white;  
 transform: translateY(-2px);  
 box-shadow: 0 4px 12px rgba(255, 71, 87, 0.3);  
}  
  
.pay-btn {  
 background: linear-gradient(135deg, var(--accent-green), #66bb6a);  
 color: white;  
 box-shadow: 0 2px 8px rgba(76, 175, 80, 0.3);  
}  
  
.pay-btn:hover {  
 transform: translateY(-2px);  
 box-shadow: 0 4px 12px rgba(76, 175, 80, 0.4);  
 background: linear-gradient(135deg, #43a047, var(--accent-green));  
}  
  
/\* Modal Styles \*/  
.modal-overlay {  
 display: none;  
 position: fixed;  
 top: 0;  
 left: 0;  
 width: 100%;  
 height: 100%;  
 background: rgba(15, 23, 42, 0.8);  
 z-index: 1000;  
 backdrop-filter: blur(8px);  
 opacity: 0;  
 transition: opacity 0.3s ease;  
}  
  
.modal-content {  
 position: absolute;  
 top: 50%;  
 left: 50%;  
 transform: translate(-50%, -50%) scale(0.8);  
 background: var(--card-bg);  
 border-radius: var(--radius);  
 width: 90%;  
 max-width: 480px;  
 max-height: 80vh;  
 overflow-y: auto;  
 box-shadow: 0 20px 60px rgba(0, 0, 0, 0.3);  
 border: 1px solid var(--card-border);  
 transition: transform 0.3s ease;  
}  
  
.modal-close {  
 position: absolute;  
 top: 16px;  
 right: 16px;  
 background: rgba(255, 71, 87, 0.1);  
 color: var(--accent-red);  
 border: 1px solid rgba(255, 71, 87, 0.2);  
 width: 36px;  
 height: 36px;  
 border-radius: 50%;  
 font-size: 1.25rem;  
 cursor: pointer;  
 transition: var(--transition);  
 display: flex;  
 align-items: center;  
 justify-content: center;  
}  
  
.modal-close:hover {  
 background: var(--accent-red);  
 color: white;  
 transform: scale(1.05);  
}  
  
.modal-header {  
 background: linear-gradient(135deg, var(--primary), var(--secondary));  
 color: white;  
 padding: 32px 24px 24px;  
 text-align: center;  
 border-radius: var(--radius) var(--radius) 0 0;  
}  
  
.modal-player-image {  
 width: 80px;  
 height: 80px;  
 border-radius: 50%;  
 object-fit: cover;  
 margin-bottom: 16px;  
 border: 3px solid rgba(255, 255, 255, 0.3);  
}  
  
.modal-player-name {  
 font-size: 1.5rem;  
 margin-bottom: 4px;  
 font-weight: 600;  
}  
  
.modal-player-role {  
 font-size: 0.875rem;  
 opacity: 0.9;  
 font-weight: 500;  
}  
  
.modal-body {  
 padding: 24px;  
}  
  
.profile-section h3 {  
 font-size: 1.125rem;  
 margin-bottom: 16px;  
 color: var(--text-main);  
 font-weight: 600;  
}  
  
.profile-details {  
 display: flex;  
 flex-direction: column;  
 gap: 12px;  
}  
  
.detail-item {  
 display: flex;  
 justify-content: space-between;  
 align-items: center;  
 padding: 10px 0;  
 border-bottom: 1px solid var(--border-subtle);  
}  
  
.detail-item:last-child {  
 border-bottom: none;  
}  
  
.detail-label {  
 font-weight: 500;  
 color: var(--text-muted);  
 font-size: 0.875rem;  
}  
  
.detail-value {  
 font-weight: 600;  
 color: var(--text-main);  
 font-size: 0.875rem;  
}  
  
/\* Animations \*/  
@keyframes pulse {  
 0% { transform: scale(1); }  
 50% { transform: scale(1.02); }  
 100% { transform: scale(1); }  
}  
  
/\* Notification Styles \*/  
.notification {  
 font-family: 'Inter', sans-serif;  
 border-radius: var(--radius-small);  
 backdrop-filter: blur(8px);  
}  
  
/\* Status Banner \*/  
.status-banner {  
 border-radius: 0;  
 backdrop-filter: blur(8px);  
}  
/\* Place this in your CSS file \*/  
  
  
/\* Responsive Design \*/  
@media (max-width: 768px) {  
 .venue-section, .players-section, .payment-section {  
 margin: 16px;  
 border-radius: var(--radius-medium);  
 }  
   
 .venue-name {  
 font-size: 1.875rem;  
 }  
   
 .section-header {  
 flex-direction: column;  
 align-items: stretch;  
 }  
   
 .host-controls {  
 justify-content: center;  
 }  
   
 .players-scroll-container {  
 gap: 12px;  
 }  
   
 .players-horizontal-scroll {  
 max-width: calc(100vw - 120px);  
 }  
   
 .scroll-arrow {  
 width: 45px;  
 height: 45px;  
 font-size: 1.3rem;  
 }  
   
 .player-card {  
 min-width: 180px;  
 max-width: 180px;  
 height: 240px;  
 padding: 20px;  
 }  
   
 .player-image {  
 width: 70px;  
 height: 70px;  
 }  
   
 .player-name {  
 font-size: 1rem;  
 margin-bottom: 14px;  
 }  
   
 .switch-host-btn,  
 .friend-btn {  
 padding: 6px 12px;  
 font-size: 0.8rem;  
 }  
   
 .payment-amount {  
 padding: 6px 12px;  
 font-size: 0.9rem;  
 }  
   
 .payment-input {  
 width: 70px;  
 padding: 6px 10px;  
 font-size: 0.9rem;  
 }  
   
 .timer-container {  
 flex-direction: column;  
 gap: 12px;  
 }  
   
 .action-buttons {  
 flex-direction: column;  
 align-items: center;  
 }  
   
 .cancel-btn, .pay-btn {  
 width: 100%;  
 max-width: 280px;  
 }  
   
 .modal-content {  
 width: 95%;  
 margin: 16px;  
 }  
   
 .countdown-timer {  
 flex-wrap: wrap;  
 gap: 6px;  
 }  
   
 .time-unit {  
 min-width: 60px;  
 padding: 8px;  
 }  
   
 .time-value {  
 font-size: 1.5rem;  
 }  
   
 .info-grid {  
 grid-template-columns: 1fr;  
 }  
   
 .password-display {  
 flex-direction: column;  
 gap: 8px;  
 }  
}

————————————————————————————————————————

## BookingDetails\BookingDetails.html

<!DOCTYPE html>  
<html lang="en">  
<head>  
 <meta charset="UTF-8" />  
 <meta name="viewport" content="width=device-width, initial-scale=1.0"/>  
 <title>Booking Details - Book&Play</title>  
 <link rel="stylesheet" href="BookingDetails.css" />  
</head>  
<body>  
  
 <!-- Venue Information Section -->  
 <section class="venue-section" >  
 <div class="venue-banner">  
 <button class="back-btn" onclick="goBack()">← Back</button>  
 <img src="" alt="Venue Image" class="venue-image" id="venueImage" />  
 <div class="venue-overlay">  
 <h1 class="venue-name" id="venueName"></h1>  
 <p class="venue-location" id="venueLocation"></p>  
 </div>  
 </div>  
  
 <div class="booking-info">  
 <div class="info-grid">  
 <div class="info-card">  
 <span class="info-label">Booking Date</span>  
 <span class="info-value" id="bookingDate">Loading...</span>  
 </div>  
 <div class="info-card">  
 <span class="info-label">Time</span>  
 <span class="info-value" id="bookingTime">Loading...</span>  
 </div>  
 <div class="info-card">  
 <span class="info-label">Booking ID</span>  
 <span class="info-value" id="bookingId"></span>  
 </div>  
 <div class="info-card total-price">  
 <span class="info-label">Total Venue Price</span>  
 <span class="info-value">$<span id="totalPrice"></span></span>  
 </div>  
 <div class="info-card privacy-card">  
 <span class="info-label">Privacy</span>  
 <button class="privacy-toggle" id="privacyToggle">  
 <span class="privacy-status">Public</span>  
 <span class="privacy-icon">🌐</span>  
 </button>  
 </div>  
 </div>  
  
 <!-- Private Room Password -->  
 <div class="password-section" id="passwordSection" style="display: none;">  
 <div class="password-card">  
 <h3>🔒 Private Room Password</h3>  
 <div class="password-display">  
 <code id="roomPassword">BP2025-ELITE-9847</code>  
 <button class="copy-btn" id="copyPassword">📋 Copy</button>  
 </div>  
 <p class="password-note">Share this password with players to join the private room</p>  
 </div>  
 </div>  
 </div>  
 </section>  
  
 <!-- Player List Section -->  
 <section class="players-section">  
 <div class="section-header">  
 <h2 class="section-title">Players</h2>  
 <div class="host-controls" id="hostControls">  
 <button class="edit-prices-btn" id="editPricesBtn">✏ Edit Prices</button>  
 <button class="save-prices-btn" id="savePricesBtn" style="display: none;">💾 Save Changes</button>  
 <button class="cancel-edit-btn" id="cancelEditBtn" style="display: none;">❌ Cancel</button>  
 </div>  
 </div>  
  
 <div class="players-scroll-container">  
 <button class="scroll-arrow scroll-left" id="scrollLeft"><span>‹</span></button>  
  
 <!-- Player Cards will be injected here -->  
 <div class="players-horizontal-scroll" id="playersScroll"></div>  
  
 <button class="scroll-arrow scroll-right" id="scrollRight"><span>›</span></button>  
 </div>  
 </section>  
  
 <!-- Payment Reminder Section -->  
 <section class="payment-section">  
 <div class="payment-reminder">  
 <div class="timer-container">  
 <div class="timer-icon">⏰</div>  
 <div class="timer-content">  
 <h3 class="timer-title">Payment Deadline</h3>  
 <div class="countdown-timer" id="countdown">  
 <div class="time-unit">  
 <span class="time-value" id="hours">01</span>  
 <span class="time-label">Hours</span>  
 </div>  
 <div class="time-separator">:</div>  
 <div class="time-unit">  
 <span class="time-value" id="minutes">45</span>  
 <span class="time-label">Minutes</span>  
 </div>  
 <div class="time-separator">:</div>  
 <div class="time-unit">  
 <span class="time-value" id="seconds">30</span>  
 <span class="time-label">Seconds</span>  
 </div>  
 </div>  
 <p class="timer-message">  
 You must pay at least <strong>20% ($<span id="minimumAmount">48.00</span>)</strong> of the total booking price before the timer expires, otherwise the booking will be automatically canceled.  
 </p>  
 </div>  
 </div>  
 </div>  
 </section>  
  
 <!-- Action Buttons -->  
 <section class="actions-section">  
 <div class="action-buttons">  
 <button class="cancel-btn">Cancel Booking</button>  
 <button class="pay-btn">Pay Now</button>  
 </div>  
 </section>  
  
 <!-- Player Details Modal -->  
 <div class="modal-overlay" id="playerModal">  
 <div class="modal-content">  
 <button class="modal-close" id="modalClose">&times;</button>  
 <div class="modal-header">  
 <img src="" alt="Player" class="modal-player-image" id="modalPlayerImage">  
 <div class="modal-player-info">  
 <h2 class="modal-player-name" id="modalPlayerName"></h2>  
 <p class="modal-player-role" id="modalPlayerRole"></p>  
 </div>  
 </div>  
 <div class="modal-body">  
 <div class="profile-section">  
 <h3>Profile Information</h3>  
 <div class="profile-details">  
 <div class="detail-item">  
 <span class="detail-label">Member Since:</span>  
 <span class="detail-value" id="memberSince"></span>  
 </div>  
 <div class="detail-item">  
 <span class="detail-label">Games Played:</span>  
 <span class="detail-value" id="gamesPlayed"></span>  
 </div>  
 <div class="detail-item">  
 <span class="detail-label">Rating:</span>  
 <span class="detail-value" id="playerRating"></span>  
 </div>  
 <div class="detail-item">  
 <span class="detail-label">Favorite Sport:</span>  
 <span class="detail-value" id="favoriteSport"></span>  
 </div>  
 <div class="detail-item">  
 <span class="detail-label">Location:</span>  
 <span class="detail-value" id="playerLocation"></span>  
 </div>  
 </div>  
 </div>  
 </div>  
 </div>  
 </div>  
  
 <script src="BookingDetails.js"></script>  
</body>  
</html>

————————————————————————————————————————

## BookingDetails\BookingDetails.js

fetchBookingDetails();  
  
function fetchBookingDetails() {  
 const urlParams = new URLSearchParams(window.location.search);  
 const bookingId = urlParams.get("booking\_id");  
  
 if (!bookingId) {  
 showNotification("Missing booking ID in URL", "error");  
 return;  
 }  
  
 fetch(`getBookingDetails.php?booking\_id=${bookingId}`)  
 .then(res => res.json())  
 .then(data => {  
 if (data.error) {  
 showNotification(data.error, "error");  
 return;  
 }  
  
 window.players = data.players; // ✅ Set players globally  
 window.currentGroupId = data.booking.group\_id; // ✅ Set group ID globally  
  
 populateBookingDetails(data.booking);  
 populatePlayerList(data.players);  
   
 // ✅ Initialize privacy toggle after data is loaded  
 initializePrivacyToggle();  
   
 // ✅ Test if privacy toggle can be manually enabled  
 setTimeout(() => {  
 testPrivacyToggle();  
 }, 1000);  
 })  
 .catch(err => {  
 console.error(err);  
 showNotification("Failed to fetch booking details", "error");  
 });  
}  
  
  
  
function populateBookingDetails(booking) {  
 document.getElementById("venueName").textContent = booking.place\_name;  
 document.getElementById("venueLocation").textContent = booking.venue\_location;  
 document.getElementById("venueImage").src = booking.venue\_image;  
 document.getElementById("bookingDate").textContent = booking.booking\_date;  
 document.getElementById("bookingTime").textContent = booking.booking\_time;  
 document.getElementById("totalPrice").textContent = booking.total\_price + " ₪";  
 document.getElementById("bookingId").textContent = booking.booking\_id;  
  
 // store group id globally for later updates  
 window.currentGroupId = booking.group\_id;  
  
 // ✅ Display privacy  
 const privacyToggle = document.getElementById("privacyToggle");  
 const passwordSection = document.getElementById("passwordSection");  
 const roomPassword = document.getElementById("roomPassword");  
  
 if (booking.privacy === "private") {  
 window.isPrivate = true;  
 privacyToggle.innerHTML = `<span class="privacy-status">Private</span><span class="privacy-icon">🔒</span>`;  
 privacyToggle.classList.add("private");  
 passwordSection.style.display = "block";  
 roomPassword.textContent = booking.group\_password;  
 } else {  
 window.isPrivate = false;  
 privacyToggle.innerHTML = `<span class="privacy-status">Public</span><span class="privacy-icon">🌐</span>`;  
 privacyToggle.classList.remove("private");  
 passwordSection.style.display = "none";  
 }  
}  
  
  
function populatePlayerList(players) {  
 const container = document.getElementById("playersScroll");  
 container.innerHTML = ""; // Clear old list  
  
 // Remove duplicate players based on username  
 const uniquePlayers = players.filter((player, index, self) =>   
 index === self.findIndex(p => p.username === player.username)  
 );  
  
 uniquePlayers.forEach(p => {  
 console.log("🎯 Adding player card", p.username);  
 const isHost = p.is\_host == "1";  
 const isCurrentUser = p.username === currentUsername;  
  
 const playerCard = document.createElement("div");  
 playerCard.className = "player-card";  
  
 // Add style class  
 if (isHost) {  
 playerCard.classList.add("host-player"); // Orange background for host  
 } else if (isCurrentUser) {  
 playerCard.classList.add("me-player"); // Green background for "me"  
 }  
  
 // Badge  
 let badgeHTML = "";  
 if (isHost && isCurrentUser) {  
 badgeHTML = `<div class="host-badge">HOST (YOU)</div>`;  
 } else if (isHost) {  
 badgeHTML = `<div class="host-badge">HOST</div>`;  
 } else if (isCurrentUser) {  
 badgeHTML = `<div class="me-badge">ME</div>`;  
 }  
  
 // Buttons  
 let buttonsHTML = "";  
 if (!isCurrentUser && currentUsername === getHostUsername(uniquePlayers)) {  
 buttonsHTML += `<button class="switch-host-btn">👑 Make Host</button>`;  
 }  
 if (!isCurrentUser) {  
 buttonsHTML += `<button class="friend-btn">👥 Add Friend</button>`;  
 }  
  
 playerCard.innerHTML = `  
 ${badgeHTML}  
 <img class="player-image" src="${p.user\_image}" alt="${p.username}" data-player="${p.username}">  
 <div class="player-name">${p.username}</div>  
 <div class="player-status">${p.payment\_status === "paid" ? "✅ Paid" : "💸 Not Paid"}</div>  
 <div class="payment-amount" data-player="${p.username}">  
 <span class="amount-value">${p.price}</span>   
 </div>  
 <input class="payment-input" data-player="${p.username}" style="display: none;" value="${p.price}" />  
 <div class="player-actions">${buttonsHTML}</div>  
 `;  
  
 container.appendChild(playerCard);  
 });  
  
 // Show or hide host controls  
 const hostControls = document.getElementById("hostControls");  
 const currentHost = getHostUsername(uniquePlayers);  
   
 console.log("🎯 Host controls check:", { currentUsername, currentHost, isHost: currentUsername === currentHost });  
   
 if (currentHost === currentUsername) {  
 hostControls.style.display = "flex";  
 } else {  
 hostControls.style.display = "none";  
 }  
  
 initializePlayerActions(); // Re-bind actions  
}  
function getHostUsername(players) {  
 const host = players.find(p => p.is\_host == "1");  
 return host ? host.username : null;  
}  
  
  
document.addEventListener("DOMContentLoaded", function () {  
 console.log("✅ BookingDetails.js is loaded");  
   
 // Initialize variables (move to global scope)  
 window.isPrivate = false;  
 window.isEditingPrices = false;  
 window.timeRemaining = 6330; // 1:45:30  
 window.originalPrices = {};  
 window.currentHost = "1";  
   
 // Initialize all functions (except privacy toggle - called after data load)  
 initializeScrolling();  
 initializeEditPrices();  
 initializePlayerActions();  
 initializeActionButtons();  
 startCountdown();  
});  
  
function initializePrivacyToggle() {  
 const privacyToggle = document.getElementById('privacyToggle');  
 const passwordSection = document.getElementById('passwordSection');  
 const copyPasswordBtn = document.getElementById('copyPassword');  
  
 if (!privacyToggle) {  
 console.log("❌ Privacy toggle element not found");  
 return;  
 }  
  
 // ✅ Check if current user is the host  
 const isHost = window.players && window.players.find(p =>   
 p.username === currentUsername && p.is\_host == "1"  
 );  
  
 console.log("🔍 Host check:", {   
 currentUsername,   
 players: window.players,   
 isHost,  
 playersLength: window.players ? window.players.length : 0,  
 privacyToggleFound: !!privacyToggle  
 });  
  
 // If not host, disable  
 if (!isHost) {  
 console.log("❌ Not host - disabling privacy toggle");  
 privacyToggle.disabled = true;  
 privacyToggle.style.cursor = "not-allowed";  
 privacyToggle.style.opacity = "0.6";  
 privacyToggle.style.pointerEvents = "none";  
 return;  
 }  
  
 console.log("✅ Is host - enabling privacy toggle");  
  
 // ✅ Enable the toggle for host  
 privacyToggle.disabled = false;  
 privacyToggle.style.cursor = "pointer";  
 privacyToggle.style.opacity = "1";  
 privacyToggle.style.pointerEvents = "auto";  
  
 // ✅ Remove any existing event listeners to prevent duplicates  
 const newToggle = privacyToggle.cloneNode(true);  
 privacyToggle.parentNode.replaceChild(newToggle, privacyToggle);  
   
 // ✅ Get the new reference  
 const freshToggle = document.getElementById('privacyToggle');  
  
 // ✅ Host can toggle  
 freshToggle.addEventListener('click', function (e) {  
 e.preventDefault();  
 e.stopPropagation();  
 console.log("🔒 Privacy toggle clicked");  
   
 window.isPrivate = !window.isPrivate;  
 const newPrivacy = window.isPrivate ? 'private' : 'public';  
  
 if (window.isPrivate) {  
 this.innerHTML = '<span class="privacy-status">Private</span><span class="privacy-icon">🔒</span>';  
 this.classList.add('private');  
 passwordSection.style.display = 'block';  
 showNotification('Room is now private. Password generated!', 'success');  
 } else {  
 this.innerHTML = '<span class="privacy-status">Public</span><span class="privacy-icon">🌐</span>';  
 this.classList.remove('private');  
 passwordSection.style.display = 'none';  
 showNotification('Room is now public. Anyone can join!', 'info');  
 }  
  
 // notify backend about the change  
 if (window.currentGroupId) {  
 const formData = new URLSearchParams();  
 formData.append('group\_id', window.currentGroupId);  
 formData.append('privacy', newPrivacy);  
  
 fetch('updatePrivacy.php', {  
 method: 'POST',  
 headers: { 'Content-Type': 'application/x-www-form-urlencoded' },  
 body: formData.toString()  
 })  
 .then(res => res.json())  
 .then(data => {  
 if (!data.success) {  
 showNotification(data.error || 'Failed to update privacy', 'error');  
 } else {  
 console.log("✅ Privacy updated successfully");  
 }  
 })  
 .catch((error) => {  
 console.error("❌ Privacy update failed:", error);  
 showNotification('Failed to update privacy', 'error');  
 });  
 }  
 });  
  
 copyPasswordBtn?.addEventListener('click', function () {  
 const password = document.getElementById('roomPassword').textContent;  
 navigator.clipboard.writeText(password).then(() => {  
 showNotification('Password copied to clipboard!', 'success');  
 this.textContent = '✅ Copied';  
 setTimeout(() => {  
 this.innerHTML = '📋 Copy';  
 }, 2000);  
 });  
 });  
}  
  
  
  
function initializeScrolling() {  
 const scrollContainer = document.getElementById('playersScroll');  
 const leftArrow = document.getElementById('scrollLeft');  
 const rightArrow = document.getElementById('scrollRight');  
 const scrollAmount = 200;  
  
 leftArrow?.addEventListener('click', () => {  
 scrollContainer.scrollBy({ left: -scrollAmount, behavior: 'smooth' });  
 });  
  
 rightArrow?.addEventListener('click', () => {  
 scrollContainer.scrollBy({ left: scrollAmount, behavior: 'smooth' });  
 });  
  
 function updateArrowStates() {  
 if (!scrollContainer) return;  
 const maxScrollLeft = scrollContainer.scrollWidth - scrollContainer.clientWidth;  
 const isAtStart = scrollContainer.scrollLeft <= 0;  
 const isAtEnd = scrollContainer.scrollLeft >= maxScrollLeft - 5;  
   
 if (leftArrow) {  
 leftArrow.disabled = isAtStart;  
 leftArrow.style.opacity = isAtStart ? '0.3' : '1';  
 }  
 if (rightArrow) {  
 rightArrow.disabled = isAtEnd;  
 rightArrow.style.opacity = isAtEnd ? '0.3' : '1';  
 }  
 }  
  
 scrollContainer?.addEventListener('scroll', updateArrowStates);  
 setTimeout(updateArrowStates, 100);  
}  
  
function initializeEditPrices() {  
 const editBtn = document.getElementById('editPricesBtn');  
 const saveBtn = document.getElementById('savePricesBtn');  
 const cancelBtn = document.getElementById('cancelEditBtn');  
  
 editBtn?.addEventListener('click', enterEditMode);  
 saveBtn?.addEventListener('click', saveChanges);  
 cancelBtn?.addEventListener('click', cancelChanges);  
}  
  
function initializePlayerActions() {  
 document.querySelectorAll('.switch-host-btn').forEach((btn, index) => {  
 btn.addEventListener('click', function() {  
 // Since host player (index 0) has no switch-host-btn, we need to adjust  
 const playerId = this.closest('.player-card').querySelector('.player-image').dataset.player;  
 switchHost(playerId);  
 });  
 });  
  
 document.querySelectorAll('.friend-btn').forEach((btn, index) => {  
 if (!btn.disabled) { // Skip disabled buttons (host player)  
 btn.addEventListener('click', function() {  
 const playerId = this.closest('.player-card').querySelector('.player-image').dataset.player;  
 toggleFriend(playerId);  
 });  
 }  
 });  
}  
  
function initializeActionButtons() {  
 // Hide action buttons if in view-only mode (from JoinGroup image click)  
 if (window.viewOnly) {  
 const actionButtons = document.querySelector('.action-buttons');  
 if (actionButtons) {  
 actionButtons.style.display = 'none';  
 }  
   
 // Also hide payment reminder section in view-only mode  
 const paymentSection = document.querySelector('.payment-section');  
 if (paymentSection) {  
 paymentSection.style.display = 'none';  
 }  
   
 // Update page title to indicate view-only mode  
 document.title = 'View Booking Details - Book&Play';  
   
 return;  
 }  
   
 document.querySelector('.pay-btn')?.addEventListener('click', handlePayNow);  
 document.querySelector('.cancel-btn')?.addEventListener('click', handleCancelBooking);  
}  
  
function switchHost(newHostId) {  
 const currentHostCard = document.querySelector('.host-player');  
 if (currentHostCard) {  
 currentHostCard.classList.remove('host-player');  
 currentHostCard.querySelector('.host-badge')?.remove();  
   
 // Add switch-host button back to old host  
 const playerActions = currentHostCard.querySelector('.player-actions');  
 const friendBtn = playerActions.querySelector('.friend-btn');  
   
 if (!playerActions.querySelector('.switch-host-btn')) {  
 const switchBtn = document.createElement('button');  
 switchBtn.className = 'switch-host-btn';  
 switchBtn.innerHTML = '👑 Make Host';  
 switchBtn.addEventListener('click', function() {  
 const playerId = this.closest('.player-card').querySelector('.player-image').dataset.player;  
 switchHost(playerId);  
 });  
 playerActions.insertBefore(switchBtn, friendBtn);  
 }  
   
 friendBtn.disabled = false;  
 friendBtn.innerHTML = '👥 Add Friend';  
 }  
  
 const newHostCard = document.querySelector(`[data-player="${newHostId}"]`).closest('.player-card');  
   
 if (newHostCard) {  
 newHostCard.classList.add('host-player');  
  
 const hostBadge = document.createElement('div');  
 hostBadge.className = 'host-badge';  
 hostBadge.textContent = 'HOST (YOU)';  
 newHostCard.appendChild(hostBadge);  
  
 // Remove switch-host button and disable friend button  
 const switchBtn = newHostCard.querySelector('.switch-host-btn');  
 if (switchBtn) switchBtn.remove();  
   
 const friendBtn = newHostCard.querySelector('.friend-btn');  
 friendBtn.disabled = true;  
 friendBtn.innerHTML = '👤 You';  
  
 showNotification('You are now the host!', 'success');  
 }  
}  
  
function toggleFriend(playerId) {  
 const friendBtn = document.querySelector(`[data-player="${playerId}"]`).closest('.player-card').querySelector('.friend-btn');  
 const isFriend = friendBtn.classList.contains('friends');  
  
 if (isFriend) {  
 friendBtn.classList.remove('friends');  
 friendBtn.innerHTML = '👥 Add Friend';  
 showNotification('Removed from friends', 'warning');  
 } else {  
 friendBtn.classList.add('friends');  
 friendBtn.innerHTML = '✅ Friends';  
 showNotification('Added as friend!', 'success');  
 }  
}  
  
function enterEditMode() {  
 document.querySelectorAll('.payment-amount').forEach(amount => {  
 window.originalPrices[amount.dataset.player] = amount.querySelector('.amount-value').textContent;  
 amount.style.display = 'none';  
 });  
   
 document.querySelectorAll('.payment-input').forEach(input => input.style.display = 'inline-block');  
   
 document.getElementById('editPricesBtn').style.display = 'none';  
 document.getElementById('savePricesBtn').style.display = 'inline-block';  
 document.getElementById('cancelEditBtn').style.display = 'inline-block';  
}  
  
function saveChanges() {  
 document.querySelectorAll('.payment-input').forEach(input => {  
 const playerId = input.dataset.player;  
 const value = parseFloat(input.value).toFixed(2);  
 const amountDisplay = document.querySelector(`.payment-amount[data-player="${playerId}"] .amount-value`);  
 amountDisplay.textContent = value;  
 });  
 exitEditMode();  
 showNotification('Prices saved!', 'success');  
}  
  
function cancelChanges() {  
 document.querySelectorAll('.payment-input').forEach(input => {  
 input.value = window.originalPrices[input.dataset.player];  
 });  
 exitEditMode();  
 showNotification('Changes canceled', 'warning');  
}  
  
function exitEditMode() {  
 document.querySelectorAll('.payment-amount').forEach(amount => amount.style.display = 'inline-block');  
 document.querySelectorAll('.payment-input').forEach(input => input.style.display = 'none');  
   
 document.getElementById('editPricesBtn').style.display = 'inline-block';  
 document.getElementById('savePricesBtn').style.display = 'none';  
 document.getElementById('cancelEditBtn').style.display = 'none';  
}  
  
function startCountdown() {  
 setInterval(() => {  
 if (window.timeRemaining > 0) {  
 window.timeRemaining--;  
 updateCountdownDisplay();  
 }  
 }, 1000);  
}  
  
function updateCountdownDisplay() {  
 const hours = Math.floor(window.timeRemaining / 3600);  
 const minutes = Math.floor((window.timeRemaining % 3600) / 60);  
 const seconds = window.timeRemaining % 60;  
  
 document.getElementById('hours').textContent = hours.toString().padStart(2, '0');  
 document.getElementById('minutes').textContent = minutes.toString().padStart(2, '0');  
 document.getElementById('seconds').textContent = seconds.toString().padStart(2, '0');  
}  
  
function handlePayNow() {  
 showNotification('Redirecting to payment...', 'info');  
 setTimeout(() => alert('Payment gateway would be here'), 1500);  
}  
  
function handleCancelBooking() {  
 if (confirm('Cancel this booking?')) {  
 showNotification('Booking canceled', 'warning');  
 }  
}  
  
function showNotification(message, type = 'info') {  
 const existing = document.querySelector('.notification');  
 if (existing) existing.remove();  
  
 const notification = document.createElement('div');  
 notification.className = `notification notification-${type}`;  
 notification.textContent = message;  
  
 const colors = {  
 success: '#4CAF50',  
 warning: '#ff9800',  
 error: '#f44336',  
 info: '#2196F3'  
 };  
  
 notification.style.cssText = `  
 position: fixed;  
 top: 20px;  
 right: 20px;  
 background: ${colors[type]};  
 color: white;  
 padding: 15px 25px;  
 border-radius: 8px;  
 font-weight: 600;  
 z-index: 1002;  
 box-shadow: 0 4px 12px rgba(0, 0, 0, 0.2);  
 transform: translateX(100%);  
 transition: transform 0.3s ease;  
 `;  
  
 document.body.appendChild(notification);  
 setTimeout(() => notification.style.transform = 'translateX(0)', 10);  
 setTimeout(() => {  
 notification.style.transform = 'translateX(100%)';  
 setTimeout(() => notification.remove(), 300);  
 }, 4000);  
}  
  
// ✅ View Booking Details (for image clicks)  
function viewBookingDetails(booking\_id) {  
 const url = `../BookingDetails/BookingDetails.php?booking\_id=${encodeURIComponent(booking\_id)}&view\_only=true`;  
 window.location.href = url;  
}  
  
// ✅ Test function to manually test privacy toggle  
function testPrivacyToggle() {  
 console.log("🧪 Testing privacy toggle manually");  
 const privacyToggle = document.getElementById('privacyToggle');  
 if (privacyToggle) {  
 privacyToggle.disabled = false;  
 privacyToggle.style.cursor = "pointer";  
 privacyToggle.style.opacity = "1";  
 console.log("✅ Privacy toggle enabled for testing");  
 } else {  
 console.log("❌ Privacy toggle element not found");  
 }  
}  
  
// ✅ Go back function  
function goBack() {  
 if (window.viewOnly) {  
 // If in view-only mode, go back to JoinGroup  
 window.location.href = '../JoinGroup/JoinGroup.php';  
 } else {  
 // Otherwise, go back to previous page  
 window.history.back();  
 }  
}  
  
window.onload = function () {  
 console.log("✅ BookingDetails.js is loaded");  
   
 // Initialize variables (move to global scope)  
 window.isPrivate = false;  
 window.isEditingPrices = false;  
 window.timeRemaining = 6330; // 1:45:30  
 window.originalPrices = {};  
 window.currentHost = "1";  
   
 // Initialize all functions  
 initializePrivacyToggle();  
 initializeScrolling();  
 initializeEditPrices();  
 initializePlayerActions();  
 initializeActionButtons();  
 startCountdown();  
};

————————————————————————————————————————

## BookingDetails\BookingDetails.php

<?php  
session\_start();  
require\_once '../../../db.php';  
$currentUsername = $\_SESSION['user\_id'] ?? '';  
  
// Check if this is a view-only mode (from JoinGroup image click)  
$viewOnly = isset($\_GET['view\_only']) && $\_GET['view\_only'] === 'true';  
  
include 'BookingDetails.html';  
  
// expose the logged in user and view\_only status to JavaScript  
echo "<script>window.currentUsername = " . json\_encode($currentUsername) . ";</script>";  
echo "<script>window.viewOnly = " . json\_encode($viewOnly) . ";</script>";  
  
exit();  
?>

————————————————————————————————————————

## BookingDetails\getBookingDetails.php

<?php  
require\_once '../../../db.php'; // adjust path as needed  
header('Content-Type: application/json');  
  
$bookingId = $\_GET['booking\_id'] ?? null;  
  
if (!$bookingId) {  
 echo json\_encode(['error' => 'Missing booking\_id']);  
 exit;  
}  
  
// ✅ Get booking data + group + venue  
$bookingSql = "  
 SELECT   
 b.\*,   
 g.group\_id,  
 g.group\_name,   
 g.privacy,  
 g.group\_password,  
 v.location AS venue\_location,   
 v.image\_url,   
 v.price,   
 v.SportCategory  
 FROM bookings b  
 LEFT JOIN groups g ON b.booking\_id = g.booking\_id  
 JOIN sportfacilities v ON b.facilities\_id = v.facilities\_id  
 WHERE b.booking\_id = ?  
";  
  
$stmt = $conn->prepare($bookingSql);  
if (!$stmt) {  
 echo json\_encode([  
 "error" => "Booking query failed: " . $conn->error,  
 "sql" => $bookingSql  
 ]);  
 exit;  
}  
$stmt->bind\_param("i", $bookingId);  
$stmt->execute();  
$bookingResult = $stmt->get\_result();  
$booking = $bookingResult->fetch\_assoc();  
  
if (!$booking) {  
 echo json\_encode([  
 "error" => "Booking not found with ID $bookingId"  
 ]);  
 exit;  
}  
  
// ✅ Get players  
$playersSql = "  
 SELECT   
 u.username,   
 u.user\_image,  
 gm.payment\_amount AS price,  
 CASE   
 WHEN u.username = g.created\_by THEN 1   
 ELSE 0   
 END AS is\_host  
 FROM groups g  
 JOIN group\_members gm ON g.group\_id = gm.group\_id  
 JOIN users u ON gm.username = u.username  
 WHERE g.booking\_id = ?  
 ORDER BY is\_host DESC, u.username ASC  
";  
  
$stmt2 = $conn->prepare($playersSql);  
if (!$stmt2) {  
 echo json\_encode([  
 "error" => "Players query failed: " . $conn->error,  
 "sql" => $playersSql  
 ]);  
 exit;  
}  
$stmt2->bind\_param("i", $bookingId);  
$stmt2->execute();  
$playersResult = $stmt2->get\_result();  
$players = [];  
  
// ✅ Ensure only one host and fix image paths  
$hostFound = false;  
while ($row = $playersResult->fetch\_assoc()) {  
 // Fix player image path  
 if ($row['user\_image'] && !empty($row['user\_image'])) {  
 $row['user\_image'] = '../../../uploads/users/' . $row['user\_image'];  
 } else {  
 $row['user\_image'] = '../../../uploads/users/default.jpg';  
 }  
   
 // Ensure only one host  
 if ($row['is\_host'] == 1) {  
 if ($hostFound) {  
 $row['is\_host'] = 0; // Convert additional hosts to regular players  
 } else {  
 $hostFound = true;  
 }  
 }  
   
 $players[] = $row;  
}  
  
// ✅ Calculate total price  
$start = new DateTime($booking['start\_time']);  
$end = new DateTime($booking['end\_time']);  
$intervalInSeconds = ($end->getTimestamp() - $start->getTimestamp());  
$hours = $intervalInSeconds / 3600;  
$totalPrice = round($booking['price'] \* $hours, 2);  
  
// ✅ Fix venue image path  
$venueImage = '';  
if ($booking['image\_url'] && !empty($booking['image\_url'])) {  
 $venueImage = '../../../uploads/venues/' . str\_replace('\\', '/', $booking['image\_url']);  
} else {  
 $venueImage = '../../../Images/staduim\_icon.png'; // Default image  
}  
  
// ✅ Send complete data  
echo json\_encode([  
 'booking' => [  
 'place\_name' => $booking['group\_name'],  
 'venue\_location' => $booking['venue\_location'],  
 'venue\_image' => $venueImage,  
 'booking\_date' => $booking['booking\_date'],  
 'booking\_time' => $booking['start\_time'] . ' - ' . $booking['end\_time'],  
 'booking\_id' => $booking['booking\_id'],  
 'total\_price' => $totalPrice,  
 'privacy' => $booking['privacy'],  
 'group\_password' => $booking['group\_password'],  
 'group\_id' => $booking['group\_id']  
 ],  
 'players' => $players  
]);  
?>

————————————————————————————————————————

## BookingDetails\makeHost.php

<?php  
require\_once '../../../db.php';  
  
$groupId = $\_POST['group\_id'] ?? null;  
$newHost = $\_POST['username'] ?? null;  
  
if (!$groupId || !$newHost) {  
 echo json\_encode(['success' => false, 'error' => 'Missing data']);  
 exit;  
}  
  
$conn->query("UPDATE group\_members SET is\_host = 0 WHERE group\_id = $groupId");  
  
$stmt = $conn->prepare("UPDATE group\_members SET is\_host = 1 WHERE group\_id = ? AND username = ?");  
$stmt->bind\_param("is", $groupId, $newHost);  
$success = $stmt->execute();  
  
echo json\_encode(['success' => $success]);  
?>

————————————————————————————————————————

## BookingDetails\updatePlayerPrice.php

<?php  
require\_once '../../../db.php';  
  
$groupId = $\_POST['group\_id'] ?? null;  
$username = $\_POST['username'] ?? null;  
$price = $\_POST['price'] ?? null;  
  
if (!$groupId || !$username || !is\_numeric($price)) {  
 echo json\_encode(['error' => 'Invalid input']);  
 exit;  
}  
  
$stmt = $conn->prepare("UPDATE group\_members SET payment\_amount = ? WHERE group\_id = ? AND username = ?");  
$stmt->bind\_param("dis", $price, $groupId, $username);  
$success = $stmt->execute();  
  
echo json\_encode(['success' => $success]);  
?>

————————————————————————————————————————

## BookingDetails\updatePrivacy.php

<?php  
session\_start();  
require\_once '../../../db.php';  
  
header('Content-Type: application/json');  
  
$username = $\_SESSION['user\_id'] ?? '';  
$groupId = $\_POST['group\_id'] ?? null;  
$privacy = $\_POST['privacy'] ?? null;  
  
if (!$username) {  
 echo json\_encode(['success' => false, 'error' => 'Not logged in']);  
 exit;  
}  
  
if (!$groupId || ($privacy !== 'public' && $privacy !== 'private')) {  
 echo json\_encode(['success' => false, 'error' => 'Invalid input']);  
 exit;  
}  
  
// Verify that the current user is the group admin  
$stmt = $conn->prepare('SELECT created\_by, group\_password FROM groups WHERE group\_id = ?');  
$stmt->bind\_param('i', $groupId);  
$stmt->execute();  
$result = $stmt->get\_result();  
$group = $result->fetch\_assoc();  
$stmt->close();  
  
if (!$group || $group['created\_by'] !== $username) {  
 echo json\_encode(['success' => false, 'error' => 'Unauthorized']);  
 exit;  
}  
  
$groupPassword = null;  
if ($privacy === 'private') {  
 // keep existing password if already set  
 $groupPassword = $group['group\_password'] ?? null;  
}  
  
$stmt = $conn->prepare('UPDATE groups SET privacy = ?, group\_password = ? WHERE group\_id = ?');  
$stmt->bind\_param('ssi', $privacy, $groupPassword, $groupId);  
$success = $stmt->execute();  
$stmt->close();  
  
echo json\_encode(['success' => $success]);  
?>

————————————————————————————————————————

## BookVenue\BookVenue.css

input[type="radio"], input[type="checkbox"] {  
 display: none;  
}  
  
  
  
  
  
.search-section {  
 padding: 16px 24px;  
 background: linear-gradient(135deg, rgba(224, 247, 250, 0.3) 0%, rgba(208, 248, 206, 0.3) 100%);  
 border-bottom: 1px solid var(--border-subtle);  
}  
  
.search-container {  
 max-width: 1400px;  
 margin: 0 auto;  
}  
  
.search-box-main {  
 display: flex;  
 align-items: center;  
 background: white;  
 border-radius: var(--radius);  
 padding: 8px;  
 box-shadow: 0 4px 6px var(--shadow-light), 0 1px 3px var(--shadow-medium);  
 border: 1px solid var(--card-border);  
 max-width: 600px;  
 margin: 0 auto;  
}  
  
.player-search {  
 flex: 1;  
 border: none;  
 outline: none;  
 padding: 12px 16px;  
 font-size: 16px;  
 font-family: inherit;  
 background: transparent;  
 color: var(--text-main);  
}  
  
.player-search::placeholder {  
 color: var(--text-muted);  
}  
  
.search-btn {  
 padding: 12px 24px;  
 background: linear-gradient(135deg, var(--primary) 0%, var(--secondary) 100%);  
 color: white;  
 border: none;  
 border-radius: var(--radius-small);  
 font-weight: 600;  
 font-size: 14px;  
 cursor: pointer;  
 transition: var(--transition);  
}  
  
.search-btn:hover {  
 transform: translateY(-1px);  
 box-shadow: 0 4px 12px rgba(30, 144, 255, 0.3);  
}  
  
/\* Layout \*/  
.layout {  
 display: flex;  
 margin: 24px;  
 position: relative;  
}  
/\* SIDEBAR - STICKY POSITION \*/  
.sidebar {  
 position: sticky;  
 top: 20px;  
 z-index: 999;  
 width: 200px;  
 background: var(--card-bg);  
 border: 1px solid var(--card-border);  
 border-radius: var(--radius);  
 padding: 20px;  
 box-shadow: 0 4px 6px var(--shadow-light), 0 1px 3px var(--shadow-medium);  
 backdrop-filter: blur(12px);  
 height: fit-content;  
 max-height: calc(100vh - 40px);  
 overflow-y: auto;  
 scrollbar-width: thin;  
 scrollbar-color: var(--primary) transparent;  
 flex-shrink: 0;  
}  
  
.sidebar::-webkit-scrollbar {  
 width: 6px;  
}  
  
.sidebar::-webkit-scrollbar-track {  
 background: transparent;  
}  
  
.sidebar::-webkit-scrollbar-thumb {  
 background: var(--primary);  
 border-radius: 3px;  
}  
  
.filter-section {  
 margin-bottom: 20px;  
}  
  
.filter-section:last-child {  
 margin-bottom: 0;  
}  
  
.filter-section h3 {  
 font-size: 14px;  
 font-weight: 600;  
 margin-bottom: 12px;  
 color: var(--text-main);  
 letter-spacing: -0.025em;  
 padding-bottom: 6px;  
 border-bottom: 1px solid var(--border-subtle);  
}  
  
.sort-option {  
 display: flex;  
 align-items: center;  
 justify-content: space-between;  
 padding: 8px 12px;  
 margin-bottom: 6px;  
 border-radius: var(--radius-small);  
 background: white;  
 color: var(--text-main);  
 font-size: 12px;  
 font-weight: 500;  
 cursor: pointer;  
 border: 1px solid var(--border-subtle);  
 transition: var(--transition);  
 user-select: none;  
}  
  
.sort-option:hover {  
 background: rgba(30, 144, 255, 0.05);  
 border-color: rgba(30, 144, 255, 0.2);  
 transform: translateY(-1px);  
 box-shadow: 0 2px 8px var(--shadow-light);  
}  
  
.sort-option input:checked + span {  
 background: linear-gradient(135deg, var(--primary) 0%, var(--secondary) 100%);  
 color: white;  
 font-weight: 600;  
 border-color: transparent;  
 box-shadow: 0 2px 8px rgba(30, 144, 255, 0.25);  
 border-radius: var(--radius-small);  
 padding: 8px 12px;  
 margin: -8px -12px;  
 display: flex;  
 align-items: center;  
 justify-content: space-between;  
 width: calc(100% + 24px);  
}  
  
.sort-option input:checked + span::after {  
 content: "✓";  
 font-size: 14px;  
 font-weight: bold;  
 opacity: 0.9;  
}  
  
/\* Location Filter \*/  
.location-filter {  
 display: flex;  
 flex-direction: column;  
 gap: 10px;  
}  
  
.search-box {  
 position: relative;  
}  
  
.search-input {  
 width: 100%;  
 padding: 8px 12px;  
 border-radius: var(--radius-small);  
 border: 1px solid var(--border-subtle);  
 font-size: 12px;  
 font-family: inherit;  
 background: white;  
 transition: var(--transition);  
}  
  
.search-input:focus {  
 outline: none;  
 border-color: var(--primary);  
 box-shadow: 0 0 0 3px rgba(30, 144, 255, 0.1);  
}  
  
.city-list {  
 max-height: 160px;  
 overflow-y: auto;  
 border: 1px solid var(--border-subtle);  
 border-radius: var(--radius-small);  
 background: white;  
 padding: 6px;  
 display: flex;  
 flex-direction: column;  
 gap: 2px;  
 scrollbar-width: thin;  
 scrollbar-color: var(--primary) transparent;  
}  
  
.city-list::-webkit-scrollbar {  
 width: 6px;  
}  
  
.city-list::-webkit-scrollbar-track {  
 background: transparent;  
}  
  
.city-list::-webkit-scrollbar-thumb {  
 background: var(--primary);  
 border-radius: 3px;  
}  
  
.city-option {  
 padding: 6px 8px;  
 border-radius: var(--radius-small);  
 cursor: pointer;  
 transition: var(--transition);  
 font-size: 12px;  
 user-select: none;  
 border: 1px solid transparent;  
}  
  
.city-option:hover {  
 background: rgba(30, 144, 255, 0.05);  
 border-color: rgba(30, 144, 255, 0.2);  
}  
  
.city-option input:checked + span {  
 background: linear-gradient(135deg, var(--primary) 0%, var(--secondary) 100%);  
 color: white;  
 font-weight: 600;  
 border-radius: var(--radius-small);  
 padding: 6px 8px;  
 margin: -6px -8px;  
 border: 1px solid transparent;  
 display: block;  
 position: relative;  
}  
  
.city-option input:checked + span::after {  
 content: "✓";  
 font-size: 12px;  
 font-weight: bold;  
 opacity: 0.9;  
 float: right;  
}  
  
/\* Main Content \*/  
.content {  
 flex: 1;  
 margin-left: 24px;  
 padding: 24px;  
 border-radius: var(--radius);  
 box-shadow: 0 4px 6px var(--shadow-light);  
}  
  
.venues-header {  
 display: flex;  
 justify-content: space-between;  
 align-items: center;  
 margin-bottom: 24px;  
 padding: 0 4px;  
}  
  
.venues-header h2 {  
 font-size: 24px;  
 font-weight: 700;  
 color: var(--text-main);  
 letter-spacing: -0.025em;  
}  
.venue-card.unavailable {  
 opacity: 0.6;  
 pointer-events: none;  
 position: relative;  
}  
  
.venue-card.unavailable::after {  
 content: "🔒 Unavailable";  
 position: absolute;  
 top: 50%;  
 left: 50%;  
 transform: translate(-50%, -50%);  
 background: rgba(220, 38, 38, 0.9); /\* red background \*/  
 color: white;  
 padding: 10px 20px;  
 border-radius: 12px;  
 font-weight: 600;  
 font-size: 18px;  
 box-shadow: 0 4px 12px rgba(220, 38, 38, 0.5);  
 text-shadow: 1px 1px 3px rgba(0,0,0,0.3);  
}  
  
.results-count {  
 color: var(--text-muted);  
 font-size: 14px;  
 font-weight: 500;  
}  
  
/\* Venues Grid \*/  
.venues-grid {  
 display: grid;  
 grid-template-columns: repeat(auto-fit, minmax(300px, 400px));  
 justify-content:start;  
 gap: 20px;  
}  
  
.venue-card {  
 background: var(--card-bg);  
 border: 1px solid var(--card-border);  
 border-radius: var(--radius);  
 overflow: hidden;  
 box-shadow: 0 4px 6px var(--shadow-light), 0 1px 3px var(--shadow-medium);  
 transition: var(--transition);  
 cursor: pointer;  
}  
  
.venue-card:hover {  
 transform: translateY(-4px);  
 box-shadow: 0 8px 25px var(--shadow-light), 0 3px 6px var(--shadow-medium);  
 border-color: rgba(30, 144, 255, 0.2);  
}  
  
.venue-image {  
 position: relative;  
 height: 200px;  
 background: #f8fafc;  
 overflow: hidden;  
}  
  
.venue-image img {  
 width: 100%;  
 height: 100%;  
 object-fit: cover;  
 transition: transform 0.3s ease;  
}  
  
.venue-card:hover .venue-image img {  
 transform: scale(1.05);  
}  
  
.venue-content {  
 padding: 20px;  
}  
  
.venue-title {  
 font-size: 18px;  
 font-weight: 700;  
 color: var(--text-main);  
 margin-bottom: 8px;  
 letter-spacing: -0.025em;  
 word-wrap: break-word;  
 hyphens: auto;  
 overflow-wrap: break-word;  
}  
  
.venue-location {  
 color: var(--text-muted);  
 font-size: 14px;  
 margin-bottom: 8px;  
 display: flex;  
 align-items: center;  
 gap: 4px;  
}  
  
.venue-footer {  
 display: flex;  
 justify-content: space-between;  
 align-items: center;  
 padding-top: 16px;  
 border-top: 1px solid var(--border-subtle);  
}  
  
.price {  
 font-size: 20px;  
 font-weight: 700;  
 background: linear-gradient(135deg, var(--primary) 0%, var(--secondary) 100%);  
 -webkit-background-clip: text;  
 -webkit-text-fill-color: transparent;  
 background-clip: text;  
}  
  
.per {  
 font-size: 14px;  
 font-weight: 500;  
 color: var(--text-muted);  
}  
  
.book-btn {  
 background: linear-gradient(135deg, var(--primary) 0%, var(--secondary) 100%);  
 color: white;  
 border: none;  
 padding: 10px 20px;  
 border-radius: var(--radius-small);  
 font-weight: 600;  
 font-size: 14px;  
 cursor: pointer;  
 transition: var(--transition);  
}  
  
.book-btn:hover {  
 transform: translateY(-1px);  
 box-shadow: 0 4px 12px rgba(30, 144, 255, 0.3);  
}  
  
/\* Responsive Design \*/  
@media (max-width: 1024px) {  
 .layout {  
 margin: 20px;  
 }  
   
 .sidebar {  
 width: 240px;  
 }  
   
 .venues-grid {  
 grid-template-columns: repeat(auto-fit, minmax(260px, 1fr));  
 gap: 18px;  
 }  
}  
  
@media (max-width: 768px) {  
 body {  
 padding-top: calc(var(--navbar-height) + 20px);  
 }  
   
 .fresh-navbar {  
 padding: 14px 20px;  
 flex-wrap: wrap;  
 gap: 12px;  
 height: auto;  
 min-height: var(--navbar-height);  
 }  
   
 .nav-links {  
 gap: 6px;  
 flex-wrap: wrap;  
 }  
   
 .nav-links li a {  
 padding: 8px 16px;  
 font-size: 13px;  
 }  
   
 .search-section {  
 padding: 20px;  
 }  
   
   
   
   
   
 .layout {  
 flex-direction: column;  
 margin: 16px;  
 }  
   
 .sidebar {  
 position: static;  
 width: 100%;  
 margin-left: 0;  
 margin-bottom: 20px;  
 max-height: none;  
 }  
   
 .content {  
 margin-left: 0;  
 }  
   
 .venues-grid {  
 grid-template-columns: 1fr;  
 gap: 16px;  
 }  
}  
  
@media (max-width: 480px) {  
 .fresh-navbar {  
 padding: 12px 16px;  
 }  
   
 .logo {  
 font-size: 18px;  
 }  
   
 .search-section {  
 padding: 16px;  
 }  
   
 .search-box-main {  
 flex-direction: column;  
 gap: 8px;  
 }  
   
 .search-btn {  
 width: 100%;  
 }  
   
 .sports-scroll {  
 padding: 16px;  
 gap: 10px;  
 }  
   
 .sport-circle {  
 width: 70px;  
 height: 70px;  
 font-size: 10px;  
 }  
   
 .sport-emoji {  
 font-size: 16px;  
 }  
   
 .venues-header {  
 flex-direction: column;  
 align-items: flex-start;  
 gap: 8px;  
 }  
   
 .venues-header h2 {  
 font-size: 20px;  
 }  
   
 .venue-card {  
 margin-bottom: 16px;  
 }  
   
 .venue-image {  
 height: 180px;  
 }  
   
 .venue-content {  
 padding: 16px;  
 }  
   
 .venue-title {  
 font-size: 16px;  
 }  
   
 .venue-footer {  
 flex-direction: column;  
 gap: 12px;  
 align-items: stretch;  
 }  
   
 .book-btn {  
 width: 100%;  
 padding: 12px;  
 }  
}  
  
/\* Focus states for accessibility \*/  
.sort-option:focus-within,  
.city-option:focus-within,  
.sport-circle:focus-within,  
.venue-card:focus-within {  
 outline: 2px solid var(--primary);  
 outline-offset: 2px;  
}  
  
/\* Animation for loading \*/  
@keyframes fadeInUp {  
 from {  
 opacity: 0;  
 transform: translateY(20px);  
 }  
 to {  
 opacity: 1;  
 transform: translateY(0);  
 }  
}  
  
.venue-card {  
 animation: fadeInUp 0.5s ease-out;  
}  
  
.venue-card:nth-child(even) {  
 animation-delay: 0.1s;  
}  
  
.venue-card:nth-child(3n) {  
 animation-delay: 0.2s;  
}  
  
  
  
.favorite-icon {  
 top: 12px;  
 right: 12px;  
 position: absolute;  
 bottom: 10px; /\* now it goes to the bottom of image \*/  
 right: 15px;  
 font-size: 28px;  
 color: rgba(255, 255, 255, 0.8); /\* always visible on every image \*/  
 text-shadow: 0 0 5px rgba(0, 0, 0, 0.6); /\* glow effect for better contrast \*/  
 cursor: pointer;  
 transition: 0.3s ease;  
 z-index: 2;  
}  
  
.favorite-icon:hover {  
 transform: scale(1.15);  
}  
  
.favorite-icon.active {  
 color: red;  
 text-shadow: 0 0 8px rgba(0, 0, 0, 0.6);  
}  
.venue-rating {  
 font-size: 14px;  
 color: #FFD700; /\* golden \*/  
 margin-bottom: 5px;  
}  
.favorite-icon.active {  
 color: red;  
}  
.favorite-icon {  
 color: white;  
 cursor: pointer;  
 font-size: 20px;  
}

————————————————————————————————————————

## BookVenue\BookVenue.html

<!DOCTYPE html>  
<html lang="en">  
<head>  
 <meta charset="UTF-8">  
 <meta name="viewport" content="width=device-width, initial-scale=1.0">  
 <title>Book&Play - Book Venue</title>  
 <link rel="stylesheet" href="../global.css">  
 <link rel="stylesheet" href="BookVenue.css">  
 <link href="https://fonts.googleapis.com/css2?family=Inter:wght@300;400;500;600;700&display=swap" rel="stylesheet">  
 <script async defer src="https://maps.googleapis.com/maps/api/js?key=AIzaSyCdDZbnpmM7I-vaVXq6N0\_hiJQmHNL5hpk&libraries=geometry"></script>  
</head>  
<body>  
  
 <!-- Player Search Section -->  
 <section class="search-section">  
 <div class="search-container">  
 <div class="search-box-main">  
 <input type="text" class="player-search" placeholder="🔍 Search for venues by name..." id="playerSearch">  
 <button class="search-btn">Search</button>  
 </div>  
 </div>  
 </section>  
 <!-- Layout -->  
 <div class="layout">  
 <!-- Sidebar -->  
 <aside class="sidebar">  
<div class="filter-section">  
 <h3>Sort by Available </h3>  
 <label class="sort-option">  
 <input type="radio" name="Available-sort" id="All-default" checked>  
 <span>All </span>  
 </label>  
 <label class="sort-option">  
 <input type="radio" name="Available-sort" id="Available">  
 <span>Available </span>  
 </label>  
   
 </div>  
 <div class="filter-section">  
 <h3>Sort by Price</h3>  
 <label class="sort-option">  
 <input type="radio" name="price-sort" id="price-low">  
 <span>Low to High</span>  
 </label>  
 <label class="sort-option">  
 <input type="radio" name="price-sort" id="price-high">  
 <span>High to Low</span>  
 </label>  
 </div>  
 <div class="filter-section">  
 <h3>Sort by Rating</h3>  
 <label class="sort-option">  
 <input type="radio" name="rating-sort" id="rating-low">  
 <span>Low to High</span>  
 </label>  
 <label class="sort-option">  
 <input type="radio" name="rating-sort" id="rating-high">  
 <span>High to Low</span>  
 </label>  
 </div>  
  
 <div class="filter-section">  
 <h3>Sort by Distance</h3>  
 <label class="sort-option">  
 <input type="radio" name="distance-sort" id="distance-near" checked>  
 <span>Nearest First</span>  
 </label>  
 <label class="sort-option">  
 <input type="radio" name="distance-sort" id="distance-far">  
 <span>Farthest First</span>  
 </label>  
 </div>  
  
   
 </aside>  
  
 <!-- Main content area -->  
 <main class="content">  
 <div class="venues-header">  
 <h2>Available Venues</h2>  
 </div>  
<div class="venues-grid" id="venuesContainer">  
 <!-- Venue 1 -->  
 <div class="venue-card" onclick="window.location.href='../VenueDetails/VenueDetails.php'">  
 <div class="venue-image">  
 <img src="https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcQ-a\_4m4HqcMGhZt7FG7n1nGijKzrERPqAGEA&s " alt="Venue Image">  
 <div class="favorite-icon" onclick="event.stopPropagation(); toggleFavorite(this)">  
   
 &#10084;  
 </div>  
 </div>  
 <div class="venue-content">  
 <h3 class="venue-title">Elite Basketball Arena</h3>  
 <div class="venue-location">  
 📍   
 <a href="https://www.google.com/maps/search/?api=1&query=Downtown Sports Complex" target="\_blank" onclick="event.stopPropagation();">  
 Downtown Sports Complex, City Center  
 </a>  
 </div>  
 <div class="venue-footer">  
 <div class="price">₪120<span class="per">/hour</span></div>  
 <button class="book-btn" onclick="event.stopPropagation(); window.location.href='VenueBookings.php';">Book Now</button>  
 </div>  
 </div>  
</div>  
  
   
 </main>  
 </div>  
<script src="BookVenue.js" defer></script>  
  
  
</body>  
</html>

————————————————————————————————————————

## BookVenue\BookVenue.js

function waitForGoogleMaps(callback) {  
 if (typeof google !== "undefined" && google.maps && google.maps.geometry) {  
 callback();  
 } else {  
 setTimeout(() => waitForGoogleMaps(callback), 100);  
 }  
}  
  
function filterVenuesBySports(sports = [], searchTerm = "") {  
 const container = document.getElementById("venuesContainer");  
 container.innerHTML = "<p>Loading venues...</p>";  
  
 const params = new URLSearchParams();  
 sports.forEach(sport => params.append("sports[]", sport));  
 if (searchTerm) params.append("search", searchTerm);  
  
 fetch("fetch\_venues.php?" + params.toString())  
 .then(res => res.json())  
 .then(data => {  
 if (!data.success) {  
 container.innerHTML = "<p>No venues found.</p>";  
 return;  
 }  
  
 const sortOptions = {  
 available: document.querySelector("input[name='Available-sort']:checked")?.id,  
 price: document.querySelector("input[name='price-sort']:checked")?.id,  
 rating: document.querySelector("input[name='rating-sort']:checked")?.id,  
 distance: document.querySelector("input[name='distance-sort']:checked")?.id  
 };  
  
 navigator.geolocation.getCurrentPosition(position => {  
 const userLatLng = new google.maps.LatLng(position.coords.latitude, position.coords.longitude);  
 data.venues.forEach(v => {  
 const venueLatLng = new google.maps.LatLng(v.latitude, v.longitude);  
 v.distance = google.maps.geometry.spherical.computeDistanceBetween(userLatLng, venueLatLng) / 1000;  
 });  
  
 const sortedVenues = sortVenues(data.venues, sortOptions);  
 renderVenues(sortedVenues);  
 }, () => {  
 const sortedVenues = sortVenues(data.venues, sortOptions);  
 renderVenues(sortedVenues);  
 });  
 })  
 .catch(err => {  
 container.innerHTML = "<p>Error loading venues.</p>";  
 console.error(err);  
 });  
}  
  
function sortVenues(venues, sortOptions) {  
 // First filter by availability if needed  
 if (sortOptions.available === "Available") {  
 venues = venues.filter(v => v.available === 1 || v.available === true);  
 }  
  
 // Sort venues based on selected options  
 venues.sort((a, b) => {  
 // Distance sorting (highest priority)  
 if (sortOptions.distance === "distance-near") {  
 if (a.distance !== undefined && b.distance !== undefined) {  
 return a.distance - b.distance;  
 }  
 } else if (sortOptions.distance === "distance-far") {  
 if (a.distance !== undefined && b.distance !== undefined) {  
 return b.distance - a.distance;  
 }  
 }  
  
 // Rating sorting  
 if (sortOptions.rating === "rating-high") {  
 const ratingA = parseFloat(a.avg\_rating) || 0;  
 const ratingB = parseFloat(b.avg\_rating) || 0;  
 if (ratingA !== ratingB) {  
 return ratingB - ratingA;  
 }  
 } else if (sortOptions.rating === "rating-low") {  
 const ratingA = parseFloat(a.avg\_rating) || 0;  
 const ratingB = parseFloat(b.avg\_rating) || 0;  
 if (ratingA !== ratingB) {  
 return ratingA - ratingB;  
 }  
 }  
  
 // Price sorting  
 if (sortOptions.price === "price-low") {  
 const priceA = parseFloat(a.price) || 0;  
 const priceB = parseFloat(b.price) || 0;  
 if (priceA !== priceB) {  
 return priceA - priceB;  
 }  
 } else if (sortOptions.price === "price-high") {  
 const priceA = parseFloat(a.price) || 0;  
 const priceB = parseFloat(b.price) || 0;  
 if (priceA !== priceB) {  
 return priceB - priceA;  
 }  
 }  
  
 return 0;  
 });  
  
 return venues;  
}  
  
function toggleFavorite(iconElement, facilityId) {  
 fetch('toggle\_favorite.php', {  
 method: 'POST',  
 headers: { 'Content-Type': 'application/x-www-form-urlencoded' },  
 body: new URLSearchParams({ facility\_id: facilityId })  
 })  
 .then(res => res.json())  
 .then(data => {  
 if (data.success) {  
 iconElement.classList.toggle('active', data.favorited);  
 }  
 });  
}  
  
function renderVenues(venues) {  
 const container = document.getElementById("venuesContainer");  
 container.innerHTML = "";  
  
 if (venues.length === 0) {  
 container.innerHTML = "<p>No venues available.</p>";  
 return;  
 }  
  
 venues.forEach(venue => {  
 const card = document.createElement("div");  
 card.className = "venue-card";  
 card.onclick = () => {  
 window.location.href = `../VenueDetails/VenueDetails.php?facility\_id=${venue.facilities\_id}`;  
 };  
  
 const ratingValue = parseFloat(venue.avg\_rating) || 0;  
 const fullStars = Math.floor(ratingValue);  
 const halfStar = (ratingValue - fullStars) >= 0.5 ? 1 : 0;  
 const emptyStars = 5 - fullStars - halfStar;  
 const starsHTML = '★'.repeat(fullStars) + (halfStar ? '½' : '') + '✩'.repeat(emptyStars);  
  
 const isFavorited = venue.is\_favorite ? 'active' : '';  
  
 card.innerHTML = `  
 <div class="venue-image">  
 <img src="../../../uploads/venues/${venue.image\_url}" alt="Venue Image">  
 <div class="favorite-icon ${isFavorited}" onclick="event.stopPropagation(); toggleFavorite(this, ${venue.facilities\_id})">&#10084;</div>  
 </div>  
 <div class="venue-content">  
 <h3 class="venue-title">${venue.place\_name}</h3>  
 <div class="venue-location">  
 📍   
 <a href="https://www.google.com/maps/search/?api=1&query=${venue.latitude},${venue.longitude}"   
 target="\_blank"   
 onclick="event.stopPropagation();">  
 ${venue.location}  
 </a>  
 </div>  
 <div class="venue-rating">  
 ${starsHTML} (${Number(venue.avg\_rating ?? 0).toFixed(1)})  
 </div>  
 <div class="venue-sport">  
 🏅 Sport: ${venue.SportCategory || "N/A"}  
 </div>  
 <div class="venue-footer">  
 <div class="price">₪${venue.price}<span class="per">/hour</span></div>  
 <div class="distance">${venue.distance?.toFixed(2) || '–'} km away</div>  
 <button class="book-btn" onclick="event.stopPropagation(); window.location.href='../CreateBooking/CreateBooking.php?facility\_id=${venue.facilities\_id}'">Book Now</button>  
 </div>  
 </div>  
 `;  
 container.appendChild(card);  
 });  
}  
  
document.addEventListener("DOMContentLoaded", () => {  
 waitForGoogleMaps(() => {  
 filterVenuesBySports([]);  
 });  
  
 const sortGroups = ["Available-sort", "price-sort", "rating-sort", "distance-sort"];  
 sortGroups.forEach(group => {  
 document.querySelectorAll(`input[name='${group}']`).forEach(radio => {  
 radio.addEventListener("change", () => {  
 const searchValue = document.getElementById("playerSearch").value.trim();  
 filterVenuesBySports([], searchValue);  
 });  
 });  
 });  
  
 const searchInput = document.getElementById("playerSearch");  
 let searchTimeout;  
  
 searchInput.addEventListener("input", () => {  
 clearTimeout(searchTimeout);  
 searchTimeout = setTimeout(() => {  
 const searchValue = searchInput.value.trim();  
 filterVenuesBySports([], searchValue);  
 }, 300);  
 });  
});

————————————————————————————————————————

## BookVenue\BookVenue.php

<?php  
session\_start();  
require\_once '../../../db.php';  
  
// Check if user is logged in  
if (!isset($\_SESSION['user\_id'])) {  
 header('Location: ../../auth/Login\_Page/Login.php');  
 exit();  
}  
require\_once '../../../components/sports-scroll.php';  
  
include '../navbar.php';  
include 'BookVenue.html';  
?>

————————————————————————————————————————

## BookVenue\fetch\_sports.php

<?php  
require\_once '../../../db.php';  
  
header('Content-Type: application/json');  
  
if ($conn->connect\_error) {  
 echo json\_encode(["success" => false, "message" => "Connection failed"]);  
 exit();  
}  
  
$sql = "SELECT sport\_name FROM sports WHERE is\_Accepted = 1";  
$result = $conn->query($sql);  
  
$sports = [];  
while ($row = $result->fetch\_assoc()) {  
 $sports[] = $row['sport\_name'];  
}  
  
echo json\_encode(["success" => true, "sports" => $sports]);  
  
$conn->close();  
?>

————————————————————————————————————————

## BookVenue\fetch\_venues.php

<?php  
require\_once '../../../db.php';  
header('Content-Type: application/json');  
  
  
if ($conn->connect\_error) {  
 echo json\_encode(["success" => false, "message" => "DB connection failed"]);  
 exit();  
}  
  
// Collect filters  
$sports = isset($\_GET['sports']) ? $\_GET['sports'] : [];  
$search = isset($\_GET['search']) ? trim($\_GET['search']) : "";  
  
$conditions = ["is\_Accepted = 1"];  
$params = [];  
$types = "";  
  
// Filter by sports  
if (!empty($sports)) {  
 $placeholders = implode(',', array\_fill(0, count($sports), '?'));  
 $conditions[] = "SportCategory IN ($placeholders)";  
 $params = array\_merge($params, $sports);  
 $types .= str\_repeat('s', count($sports));  
}  
  
// Filter by venue name start  
if (!empty($search)) {  
 $conditions[] = "place\_name LIKE ?";  
 $params[] = $search . '%';  
 $types .= 's';  
}  
  
$whereClause = implode(" AND ", $conditions);  
$sql = "  
 SELECT f.\*, ROUND(AVG(r.rating\_value), 1) AS avg\_rating  
 FROM sportfacilities f  
 LEFT JOIN ratings r ON f.facilities\_id = r.facilities\_id   
 WHERE $whereClause  
 GROUP BY f.facilities\_id  
";  
  
  
$stmt = $conn->prepare($sql);  
if ($params) {  
 $stmt->bind\_param($types, ...$params);  
}  
  
$stmt->execute();  
$result = $stmt->get\_result();  
  
$venues = [];  
while ($row = $result->fetch\_assoc()) {  
 $venues[] = $row;  
}  
  
echo json\_encode(["success" => true, "venues" => $venues]);  
?>

————————————————————————————————————————

## BookVenue\toggle\_favorite.php

<?php  
session\_start();  
require\_once '../../../db.php';  
  
header('Content-Type: application/json');  
  
$user\_id = $\_SESSION['user\_id'] ?? null;  
$facility\_id = $\_POST['facility\_id'] ?? null;  
  
if (!$user\_id || !$facility\_id) {  
 echo json\_encode(['success' => false, 'message' => 'Missing user or facility ID']);  
 exit;  
}  
  
// Check if already exists  
$stmt = $conn->prepare("SELECT \* FROM user\_favorite\_facilities WHERE user\_id = ? AND facility\_id = ?");  
$stmt->bind\_param("si", $user\_id, $facility\_id);  
$stmt->execute();  
$result = $stmt->get\_result();  
  
if ($result->num\_rows > 0) {  
 // Remove favorite  
 $delete = $conn->prepare("DELETE FROM user\_favorite\_facilities WHERE user\_id = ? AND facility\_id = ?");  
 $delete->bind\_param("si", $user\_id, $facility\_id);  
 $delete->execute();  
 echo json\_encode(['success' => true, 'favorited' => false]);  
} else {  
 // Add favorite  
 $insert = $conn->prepare("INSERT INTO user\_favorite\_facilities (user\_id, facility\_id) VALUES (?, ?)");  
 $insert->bind\_param("si", $user\_id, $facility\_id);  
 $insert->execute();  
 echo json\_encode(['success' => true, 'favorited' => true]);  
}  
?>

————————————————————————————————————————

## Chats\Chats.css

:root {  
 --primary: #1e90ff;  
 --secondary: #22c55e;  
 --bg-light: #e0f7fa;  
 --bg-secondary: #d0f8ce;  
 --card-bg: rgba(255, 255, 255, 0.95);  
 --text-main: #0f172a;  
 --text-light: #475569;  
 --radius-sm: 8px;  
 --radius-md: 14px;  
 --radius-lg: 24px;  
 --shadow-sm: 0 4px 6px rgba(0, 0, 0, 0.05);  
 --shadow-md: 0 6px 16px rgba(0, 0, 0, 0.08);  
 --shadow-lg: 0 10px 25px rgba(0, 0, 0, 0.12);  
 --transition: all 0.3s cubic-bezier(0.4, 0, 0.2, 1);  
}  
  
body {  
 margin: 0;  
 font-family: 'Outfit', sans-serif;  
 background: linear-gradient(135deg, var(--bg-light), var(--bg-secondary));  
 color: var(--text-main);  
 min-height: 100vh;  
 padding: 0;  
 box-sizing: border-box;  
}  
  
\*, \*:before, \*:after {  
 box-sizing: inherit;  
}  
  
/\* Navbar \*/  
.navbar {  
 display: flex;  
 justify-content: space-between;  
 align-items: center;  
 padding: 16px 30px;  
 background: linear-gradient(to right, var(--bg-light), var(--bg-secondary));  
 box-shadow: var(--shadow-sm);  
 position: sticky;  
 top: 0;  
 z-index: 1000;  
 flex-wrap: wrap;  
 gap: 10px;  
}  
  
.fresh-navbar {  
 display: flex;  
 justify-content: space-between;  
 align-items: center;  
 padding: 14px 5%;  
 background: linear-gradient(to right, #e0f7fa, #d0f8ce);  
 box-shadow: 0 2px 12px rgba(0, 0, 0, 0.05);  
 position: sticky;  
 top: 0;  
 z-index: 1000;  
 border-bottom: 1px solid rgba(0, 0, 0, 0.04);  
 flex-wrap: wrap;  
 gap: 10px;  
}  
  
.logo {  
 font-size: 22px;  
 font-weight: bold;  
 background: linear-gradient(to right, #0ea5e9, #22c55e);  
 -webkit-background-clip: text;  
 -webkit-text-fill-color: transparent;  
 background-clip: text;  
 margin-right: auto;  
}  
  
.nav-links {  
 list-style: none;  
 display: flex;  
 gap: 14px;  
 align-items: center;  
 margin: 0;  
 padding: 0;  
 flex-wrap: wrap;  
 justify-content: center;  
}  
  
.nav-links li a {  
 display: block;  
 padding: 10px 18px;  
 border-radius: 20px;  
 font-size: 14px;  
 font-weight: 500;  
 text-decoration: none;  
 color: var(--text-main);  
 background: white;  
 transition: all 0.3s ease;  
 box-shadow: 0 2px 5px rgba(0, 0, 0, 0.04);  
 text-align: center;  
}  
  
.nav-links li a:hover {  
 transform: translateY(-1px);  
 box-shadow: 0 3px 10px rgba(0, 0, 0, 0.06);  
 background: linear-gradient(to right, #4fc3f7, #81c784);  
 color: white;  
}  
  
.nav-links li a.active {  
 background: linear-gradient(to right, #4fc3f7, #81c784);  
 color: white;  
 font-weight: bold;  
 box-shadow: 0 0 10px rgba(79, 195, 247, 0.3);  
}  
  
.nav-links li a.logout {  
 background: rgba(244, 67, 54, 0.15);  
 color: #f44336;  
 font-weight: 600;  
}  
  
.top-buttons {  
 display: flex;  
 justify-content: center;  
 margin: 20px 0;  
 flex-wrap: wrap;  
 gap: 10px;  
}  
  
.top-buttons button {  
 background: white;  
 border: none;  
 padding: 10px 22px;  
 margin: 0;  
 border-radius: var(--radius-md);  
 cursor: pointer;  
 box-shadow: var(--shadow-sm);  
 font-weight: 600;  
 color: var(--text-main);  
 transition: var(--transition);  
 min-width: 120px;  
}  
  
.top-buttons .active {  
 background: linear-gradient(to right, #4fc3f7, #81c784);  
 color: white;  
 box-shadow: var(--shadow-md);  
}  
  
/\* Layout \*/  
.main-container {  
 backdrop-filter: blur(8px);  
 display: flex;  
 height: 75vh;  
 margin: 0 5%;  
 border-radius: var(--radius-lg);  
 overflow: hidden;  
 box-shadow: var(--shadow-md);  
 background: var(--card-bg);  
}  
  
.sidebar {  
 width: 30%;  
 padding: 20px;  
 overflow-y: auto;  
 border-top-left-radius: var(--radius-lg);  
 border-bottom-left-radius: var(--radius-lg);  
 box-shadow: var(--shadow-md);  
 backdrop-filter: blur(8px);  
 border-right: none;  
}  
  
.chat-section {  
 width: 70%;  
 display: flex;  
 flex-direction: column;  
 justify-content: space-between;  
 background: linear-gradient(135deg, #e0f7fa 0%, #d0f8ce 100%);  
 backdrop-filter: blur(10px);  
}  
  
.chat-header {  
 background: linear-gradient(to right, #4fc3f7, #81c784);  
 padding: 20px;  
 color: white;  
 font-weight: 600;  
 font-size: 1.1rem;  
 border-top-right-radius: var(--radius-lg);  
 box-shadow: var(--shadow-md);  
}  
  
.chat-body {  
 flex-grow: 1;  
 padding: 20px;  
 overflow-y: auto;  
 display: flex;  
 flex-direction: column;  
}  
  
.message {  
 max-width: 65%;  
 padding: 12px 18px;  
 border-radius: var(--radius-lg);  
 margin-bottom: 12px;  
 font-size: 1rem;  
 box-shadow: var(--shadow-sm);  
 background: linear-gradient(to right, #4fc3f7, #1e90ff);  
 color: white;  
 align-self: flex-start;  
 backdrop-filter: blur(8px);  
 transition: var(--transition);  
 word-wrap: break-word;  
 overflow-wrap: break-word;  
}  
  
.message.right {  
 background: linear-gradient(to right, #34d399, #22c55e);  
 color: white;  
 align-self: flex-end;  
 text-align: right;  
}  
  
.message:hover {  
 transform: translateY(-3px);  
 box-shadow: var(--shadow-md);  
 opacity: 0.95;  
}  
  
.chat-input {  
 display: flex;  
 padding: 14px 20px;  
 border-top: 1px solid #ccc;  
 background: linear-gradient(to right, #e0f7fa, #d0f8ce);  
 backdrop-filter: blur(10px);  
 box-shadow: var(--shadow-sm);  
 border-bottom-left-radius: var(--radius-lg);  
 border-bottom-right-radius: var(--radius-lg);  
 box-shadow: 0 8px 16px rgba(30, 144, 255, 0.1);  
 flex-wrap: wrap;  
 gap: 10px;  
}  
  
.chat-input input {  
 flex: 1;  
 padding: 12px 16px;  
 border-radius: var(--radius-md);  
 border: none;  
 font-size: 1rem;  
 margin-right: 10px;  
 box-shadow: var(--shadow-sm);  
 background: linear-gradient(to right, #ccf1ff, #c5f2d3);  
 color: var(--text-main);  
 outline: none;  
 min-width: 150px;  
}  
  
.chat-input button {  
 background: linear-gradient(to right, var(--primary), var(--secondary));  
 color: white;  
 border: none;  
 padding: 10px 18px;  
 border-radius: var(--radius-md);  
 font-weight: 600;  
 cursor: pointer;  
 transition: var(--transition);  
 box-shadow: var(--shadow-md);  
 white-space: nowrap;  
}  
  
.chat-input input::placeholder {  
 color: #475569;  
 opacity: 1;  
}  
  
.chat-input button:hover {  
 background: linear-gradient(to right, var(--primary), var(--secondary));  
 transform: scale(1.05);  
 opacity: 0.9;  
}  
  
.user-entry {  
 background: white;  
 padding: 14px 18px;  
 margin-bottom: 12px;  
 border-radius: var(--radius-md);  
 cursor: pointer;  
 box-shadow: var(--shadow-sm);  
 transition: var(--transition);  
 position: relative;  
 word-wrap: break-word;  
}  
  
.user-entry:hover {  
 background: linear-gradient(to right, #bae6fd, #bbf7d0);  
 color: var(--text-main);  
 transform: scale(1.02);  
 box-shadow: var(--shadow-md);  
}  
  
.user-entry.active {  
 background: linear-gradient(to right, #4fc3f7, #81c784);  
 color: white;  
 font-weight: 600;  
 box-shadow: 0 4px 12px rgba(30, 144, 255, 0.15);  
}  
  
  
  
.my-freinds-title {  
 font-size: 20px;  
 text-align: center;  
 font-weight: 700;  
 margin-right: 1000px;  
 background: linear-gradient(to right, #000000, #000000);  
 -webkit-background-clip: text;  
 -webkit-text-fill-color: transparent;  
 text-shadow: 2px 2px 8px rgba(0, 0, 0, 0.05);  
 font-family: 'Outfit', sans-serif;  
 letter-spacing: 0.5px;  
}  
  
  
  
  
  
  
/\* Friends Section \*/  
.friends-section {  
 display: flex;  
 flex-wrap: wrap;  
 gap: 20px;  
 margin: 0 5%;  
 padding: 25px;  
 background: linear-gradient(135deg, #e0f7fa, #d0f8ce);  
 border-radius: var(--radius-lg);  
 box-shadow: var(--shadow-md);  
 justify-content: center;  
 backdrop-filter: blur(8px);  
}  
  
.friend-card {  
 width: 160px;  
 text-align: center;  
 background: rgba(255, 255, 255, 0.85);  
 border-radius: var(--radius-md);  
 padding: 16px;  
 box-shadow: var(--shadow-sm);  
 transition: var(--transition);  
 cursor: pointer;  
 backdrop-filter: blur(6px);  
}  
  
.friend-card img {  
 width: 60px;  
 height: 60px;  
 border-radius: 50%;  
 margin-bottom: 10px;  
 object-fit: cover;  
 box-shadow: var(--shadow-sm);  
}  
  
.friend-search {  
 display: flex;  
 justify-content: center;  
 margin: 20px 0;  
 padding: 0 5%;  
}  
  
  
.my-friends-title {  
 font-size: 20px;  
 text-align: center;  
 font-weight: 700;  
 margin-right: 1000px;  
 background: linear-gradient(to right, #000000, #000000);  
 -webkit-background-clip: text;  
 -webkit-text-fill-color: transparent;  
 text-shadow: 2px 2px 8px rgba(0, 0, 0, 0.05);  
 font-family: 'Outfit', sans-serif;  
 letter-spacing: 0.5px;  
}  
  
  
/\* Friends Table Style \*/  
.friends-table {  
 display: flex;  
 flex-direction: column;  
 gap: 16px;  
 padding: 0 5% 40px 5%;  
}  
  
.friend-row {  
 display: flex;  
 align-items: center;  
 justify-content: space-between;  
 background: rgba(255, 255, 255, 0.85);  
 border-radius: var(--radius-md);  
 padding: 14px 20px;  
 box-shadow: var(--shadow-sm);  
 backdrop-filter: blur(6px);  
 transition: var(--transition);  
 flex-wrap: wrap;  
 gap: 10px;  
}  
  
.friend-row:hover {  
 box-shadow: var(--shadow-md);  
 transform: scale(1.01);  
}  
  
.friend-img {  
 border-radius: 50%;  
 width: 50px;  
 height: 50px;  
 object-fit: cover;  
 margin-right: 16px;  
 box-shadow: var(--shadow-sm);  
 flex-shrink: 0;  
}  
  
.friend-name {  
 flex: 1;  
 font-weight: 600;  
 font-size: 1.1rem;  
 color: var(--text-main);  
 min-width: 120px;  
}  
  
.friend-actions {  
 display: flex;  
 flex-wrap: wrap;  
 gap: 10px;  
}  
  
.friend-actions button {  
 padding: 8px 14px;  
 border: none;  
 border-radius: var(--radius-sm);  
 cursor: pointer;  
 font-weight: 500;  
 color: white;  
 background: linear-gradient(to right, #4fc3f7, #1e90ff);  
 box-shadow: var(--shadow-sm);  
 transition: var(--transition);  
 white-space: nowrap;  
}  
  
.friend-actions button:hover {  
 opacity: 0.9;  
}  
  
/\* Responsive Media Queries \*/  
@media (max-width: 1200px) {  
 .main-container {  
 margin: 0 3%;  
 }  
   
 .friends-section,   
 .friends-table {  
 padding: 20px 3%;  
 margin: 0 3%;  
 }  
   
 .friend-card {  
 width: 140px;  
 }  
}  
  
@media (max-width: 992px) {  
 .sidebar {  
 width: 40%;  
 }  
   
 .chat-section {  
 width: 60%;  
 }  
   
 .message {  
 max-width: 75%;  
 }  
   
 .friend-card {  
 width: 130px;  
 padding: 12px;  
 }  
}  
  
@media (max-width: 768px) {  
 .navbar,  
 .fresh-navbar {  
 padding: 14px 20px;  
 justify-content: center;  
 }  
   
 .logo {  
 margin-right: 0;  
 margin-bottom: 10px;  
 width: 100%;  
 text-align: center;  
 }  
   
 .main-container {  
 flex-direction: column;  
 height: auto;  
 margin: 0 20px 20px;  
 }  
   
 .sidebar,   
 .chat-section {  
 width: 100%;  
 border-radius: 0;  
 }  
   
 .sidebar {  
 border-top-left-radius: var(--radius-lg);  
 border-top-right-radius: var(--radius-lg);  
 max-height: 300px;  
 }  
   
 .chat-section {  
 border-bottom-left-radius: var(--radius-lg);  
 border-bottom-right-radius: var(--radius-lg);  
 }  
   
 .chat-header {  
 border-radius: 0;  
 }  
   
 .message {  
 max-width: 85%;  
 }  
   
 .friends-section {  
 margin: 20px;  
 }  
   
 .friend-card {  
 width: calc(50% - 20px);  
 min-width: 120px;  
 }  
   
 .friend-row {  
 padding: 12px 15px;  
 }  
   
 .friend-name {  
 font-size: 1rem;  
 }  
   
 .friend-actions button {  
 padding: 6px 12px;  
 font-size: 0.9rem;  
 }  
}  
  
@media (max-width: 576px) {  
 .nav-links {  
 gap: 8px;  
 width: 100%;  
 justify-content: center;  
 }  
   
 .nav-links li a {  
 padding: 8px 14px;  
 font-size: 0.9rem;  
 }  
   
 .main-container {  
 margin: 0 10px 20px;  
 }  
   
 .chat-input {  
 padding: 10px 15px;  
 }  
   
 .chat-input input {  
 padding: 10px;  
 margin-right: 0;  
 width: 100%;  
 }  
   
 .chat-input button {  
 width: 100%;  
 margin-top: 10px;  
 }  
   
 .message {  
 max-width: 90%;  
 font-size: 0.95rem;  
 }  
   
 .friend-card {  
 width: 100%;  
 }  
   
 .friend-img {  
 width: 40px;  
 height: 40px;  
 margin-right: 10px;  
 }  
   
 .friend-row {  
 justify-content: center;  
 text-align: center;  
 }  
   
 .friend-name {  
 width: 100%;  
 text-align: center;  
 margin: 8px 0;  
 }  
   
 .friend-actions {  
 width: 100%;  
 justify-content: center;  
 }  
}  
  
@media (max-width: 375px) {  
 .navbar,  
 .fresh-navbar {  
 padding: 10px;  
 }  
   
 .top-buttons button {  
 width: 100%;  
 }  
   
 .search-box {  
 padding: 8px 12px;  
 }  
}

————————————————————————————————————————

## Chats\Chats.html

<!DOCTYPE html>  
<html lang="en">  
<head>  
 <meta charset="UTF-8">  
 <title>Messages | Book&Play</title>  
 <link rel="stylesheet" href="Chats.css">  
 <link href="https://fonts.googleapis.com/css2?family=Outfit:wght@400;600&display=swap" rel="stylesheet">  
</head>  
<body>  
  
  
<div>  
 <h1 class="my-friends-title">My Friends :</h1>  
</div>  
  
<div class="main-container" id="chatSection">  
 <!-- Sidebar -->  
 <div class="sidebar" id="sidebar">  
   
 </div>  
  
 <!-- Chat Area -->  
 <div class="chat-section">  
 <div class="chat-header" id="chatHeader">  
 Select a user  
 </div>  
 <div class="chat-body" id="chatBody">  
 <p class="message left" style="display: none;"></p>  
 <p class="message right" style="display: none;"></p>  
 </div>  
 <div class="chat-input">  
 <input type="text" id="messageInput" placeholder="Type a message...">  
 <button id="sendButton">Send</button>  
</div>  
 </div>  
</div>  
  
<script>  
 const sessionUsername = "<?php echo $\_SESSION['user\_id']; ?>";  
</script>  
<script src="Chats.js" defer></script>  
</body>  
</html>

————————————————————————————————————————

## Chats\Chats.js

let currentChatUser = null; // Current user in chat  
  
// When page loads, fetch conversations  
window.addEventListener('DOMContentLoaded', () => {  
 fetch("fetch\_conversations.php")  
 .then(response => response.json())  
 .then(data => {  
 if (data.success) {  
 const sidebar = document.getElementById("sidebar");  
 sidebar.innerHTML = "";  
  
 data.users.forEach(username => {  
 const userDiv = document.createElement("div");  
 userDiv.className = "user-entry";  
 userDiv.textContent = username;  
 userDiv.onclick = () => loadChat(username);  
 sidebar.appendChild(userDiv);  
 });  
 }  
 });  
});  
  
// Load conversation when clicked  
function loadChat(username) {  
 currentChatUser = username;  
 document.getElementById("chatHeader").textContent = username;  
  
 fetch(`fetch\_messages.php?chat\_with=${encodeURIComponent(username)}`)  
 .then(res => res.json())  
 .then(data => {  
 const chatBody = document.getElementById("chatBody");  
 chatBody.innerHTML = "";  
  
 if (data.success) {  
 data.messages.forEach(msg => {  
 const messageElement = document.createElement("p");  
 messageElement.classList.add("message");  
  
 if (msg.sender\_username === sessionUsername) {  
 messageElement.classList.add("right");  
 }  
  
 messageElement.textContent = msg.message\_text;  
 chatBody.appendChild(messageElement);  
 });  
 }  
 });  
}  
  
// Send message when pressing "Send" button or Enter  
document.addEventListener("DOMContentLoaded", () => {  
 const sendButton = document.getElementById("sendButton");  
 const messageInput = document.getElementById("messageInput");  
  
 function sendMessage() {  
 const message = messageInput.value.trim();  
 if (!message || !currentChatUser) return;  
  
 fetch("send\_message.php", {  
 method: "POST",  
 headers: { "Content-Type": "application/x-www-form-urlencoded" },  
 body: new URLSearchParams({  
 receiver: currentChatUser,  
 message: message  
 })  
 })  
 .then(res => res.json())  
 .then(data => {  
 if (data.success) {  
 const chatBody = document.getElementById("chatBody");  
 const msg = document.createElement("p");  
 msg.className = "message right";  
 msg.textContent = message;  
 chatBody.appendChild(msg);  
 messageInput.value = "";  
 } else {  
 alert("❌ Failed to send message.");  
 }  
 });  
 }  
  
 // When pressing Send button  
 sendButton.addEventListener("click", sendMessage);  
  
 // When pressing Enter  
 messageInput.addEventListener("keypress", (e) => {  
 if (e.key === "Enter") {  
 e.preventDefault();  
 sendMessage();  
 }  
 });  
});

————————————————————————————————————————

## Chats\Chats.php

<?php  
session\_start();  
require\_once '../../../db.php';  
if (!isset($\_SESSION['user\_id'])) {  
 // Unset all session variables  
 session\_unset();  
 // Destroy the session completely  
 session\_destroy();  
 header('Location: ../../auth/Login\_Page/Login.php');  
 exit();  
}  
include '../navbar.php';  
include 'Chats.html';  
exit();  
?>

————————————————————————————————————————

## Chats\fetch\_conversations.php

<?php  
session\_start();  
require\_once '../../db.php'; // change path according to your project  
  
header('Content-Type: application/json');  
  
if (!isset($\_SESSION['user\_id'])) {  
 echo json\_encode(["success" => false, "message" => "Not logged in"]);  
 exit;  
}  
  
$currentUser = $\_SESSION['user\_id'];  
  
$sql = "SELECT DISTINCT   
 CASE   
 WHEN sender\_username = ? THEN receiver\_username   
 ELSE sender\_username   
 END AS chat\_partner  
 FROM messages  
 WHERE sender\_username = ? OR receiver\_username = ?";  
  
$stmt = $conn->prepare($sql);  
$stmt->bind\_param("sss", $currentUser, $currentUser, $currentUser);  
$stmt->execute();  
$result = $stmt->get\_result();  
  
$chatUsers = [];  
while ($row = $result->fetch\_assoc()) {  
 $chatUsers[] = $row['chat\_partner'];  
}  
  
echo json\_encode(["success" => true, "users" => $chatUsers]);

————————————————————————————————————————

## Chats\fetch\_messages.php

<?php  
session\_start();  
require\_once '../../db.php'; // adjust according to your project  
  
header('Content-Type: application/json');  
  
if (!isset($\_SESSION['user\_id'])) {  
 echo json\_encode(["success" => false, "message" => "Not logged in"]);  
 exit;  
}  
  
$currentUser = $\_SESSION['user\_id'];  
$chatPartner = $\_GET['chat\_with'] ?? '';  
  
if (!$chatPartner) {  
 echo json\_encode(["success" => false, "message" => "No user selected"]);  
 exit;  
}  
  
$sql = "SELECT sender\_username, receiver\_username, message\_text, sent\_at   
 FROM messages  
 WHERE   
 (sender\_username = ? AND receiver\_username = ?)  
 OR (sender\_username = ? AND receiver\_username = ?)  
 ORDER BY sent\_at ASC";  
  
$stmt = $conn->prepare($sql);  
$stmt->bind\_param("ssss", $currentUser, $chatPartner, $chatPartner, $currentUser);  
$stmt->execute();  
$result = $stmt->get\_result();  
  
$messages = [];  
while ($row = $result->fetch\_assoc()) {  
 $messages[] = $row;  
}  
  
echo json\_encode(["success" => true, "messages" => $messages]);  
?>

————————————————————————————————————————

## Chats\send\_message.php

<?php  
session\_start();  
require\_once '../../db.php';  
  
header('Content-Type: application/json');  
  
$sender = $\_SESSION['user\_id'] ?? '';  
$receiver = $\_POST['receiver'] ?? '';  
$message = trim($\_POST['message'] ?? '');  
  
if (!$sender || !$receiver || !$message) {  
 echo json\_encode(["success" => false, "message" => "Invalid input"]);  
 exit;  
}  
  
$stmt = $conn->prepare("INSERT INTO messages (sender\_username, receiver\_username, message\_text) VALUES (?, ?, ?)");  
$stmt->bind\_param("sss", $sender, $receiver, $message);  
  
if ($stmt->execute()) {  
 echo json\_encode(["success" => true]);  
} else {  
 echo json\_encode(["success" => false, "message" => "Database error"]);  
}  
?>

————————————————————————————————————————

## CreateBooking\CreateBooking.css

/\* CSS Variables for themes \*/  
  
:root {  
 --primary: #1e90ff;  
 --secondary: #4fc3f7;  
 --success: #10b981;  
 --warning: #f59e0b;  
 --danger: #ef4444;  
 --text-main: #1f2937;  
 --text-muted: #6b7280;  
 --card-bg: rgba(255, 255, 255, 0.9);  
 --card-border: rgba(0, 0, 0, 0.1);  
 --border-subtle: rgba(0, 0, 0, 0.06);  
 --shadow-light: rgba(0, 0, 0, 0.1);  
 --radius: 12px;  
 --radius-small: 8px;  
 --transition: all 0.2s ease;  
}  
  
.back-btn {  
 display: inline-flex;  
 align-items: center;  
 gap: 8px;  
 padding: 10px 20px;  
 border-radius: 24px;  
 font-size: 14px;  
 font-weight: 600;  
 text-decoration: none;  
 color: var(--text-main);  
 background: var(--card-bg);  
 border: 1px solid var(--border-subtle);  
 transition: var(--transition);  
 box-shadow: 0 2px 5px var(--shadow-light);  
}  
  
.back-btn:hover {  
 transform: translateY(-2px);  
 background: linear-gradient(135deg, var(--primary), var(--secondary));  
 color: white;  
 border-color: transparent;  
 box-shadow: 0 4px 12px var(--shadow-medium);  
}  
  
  
/\* Main Container \*/  
.container {  
 max-width: 1200px;  
 margin: 0 auto;  
 padding: 0px 24px;  
}  
  
.booking-layout {  
 display: grid;  
 grid-template-columns: 1fr 400px;  
 gap: 32px;  
}  
  
/\* Venue Info Card \*/  
.venue-info {  
 background: var(--card-bg);  
 border: 1px solid var(--card-border);  
 border-radius: var(--radius);  
 overflow: hidden;  
 margin-bottom: 24px;  
 box-shadow: 0 4px 20px var(--shadow-light);  
 backdrop-filter: blur(10px);  
}  
  
.venue-image {  
 height: 220px;  
 display: flex;  
 align-items: center;  
 justify-content: center;  
 position: relative;  
 overflow: hidden;  
 border-radius: var(--radius);  
 padding: 0;  
 background: none;  
}  
  
  
.venue-image img {  
 max-width: 100%;  
 max-height: 100%;  
 object-fit: cover;  
 border-radius: 12px;  
}  
  
  
.venue-image::before {  
 content: '';  
 position: absolute;  
 top: 0;  
 left: 0;  
 right: 0;  
 bottom: 0;  
 background: url('data:image/svg+xml,<svg xmlns="http://www.w3.org/2000/svg" viewBox="0 0 100 100"><circle cx="50" cy="50" r="40" fill="none" stroke="rgba(255,255,255,0.1)" stroke-width="2"/><circle cx="30" cy="30" r="15" fill="rgba(255,255,255,0.05)"/><circle cx="70" cy="70" r="20" fill="rgba(255,255,255,0.05)"/></svg>');  
}  
  
.venue-details {  
 padding: 24px;  
}  
  
.venue-title {  
 font-size: 24px;  
 font-weight: 700;  
 margin-bottom: 8px;  
 color: var(--text-main);  
}  
  
.venue-location {  
 color: var(--text-muted);  
 margin-bottom: 16px;  
 display: flex;  
 align-items: center;  
 gap: 8px;  
 font-weight: 500;  
}  
  
.venue-price {  
 font-size: 28px;  
 font-weight: 700;  
 background: linear-gradient(135deg, var(--primary), var(--secondary));  
 -webkit-background-clip: text;  
 -webkit-text-fill-color: transparent;  
}  
  
.venue-price .per {  
 font-size: 16px;  
 color: var(--text-muted);  
 font-weight: 500;  
}  
  
/\* Booking Form \*/  
.booking-form {  
 background: var(--card-bg);  
 border: 1px solid var(--card-border);  
 border-radius: var(--radius);  
 padding: 24px;  
 box-shadow: 0 4px 20px var(--shadow-light);  
 backdrop-filter: blur(10px);  
}  
  
.form-section {  
 margin-bottom: 24px;  
}  
  
.form-section:last-child {  
 margin-bottom: 0;  
}  
  
.section-title {  
 font-size: 18px;  
 font-weight: 600;  
 margin-bottom: 16px;  
 color: var(--text-main);  
}  
  
.form-group {  
 margin-bottom: 16px;  
}  
  
.form-label {  
 display: block;  
 font-weight: 500;  
 margin-bottom: 8px;  
 color: var(--text-main);  
}  
  
.form-input {  
 width: 100%;  
 padding: 12px 16px;  
 border: 1px solid var(--card-border);  
 border-radius: var(--radius-small);  
 font-size: 14px;  
 transition: var(--transition);  
 background: rgba(255, 255, 255, 0.8);  
 color: var(--text-main);  
}  
  
.form-input:focus {  
 outline: none;  
 border-color: var(--primary);  
 box-shadow: 0 0 0 3px rgba(30, 144, 255, 0.1);  
 background: rgba(255, 255, 255, 0.95);  
}  
  
.form-row {  
 display: grid;  
 grid-template-columns: 1fr 1fr;  
 gap: 16px;  
}  
  
.form-row.three-columns {  
 grid-template-columns: 1fr 1fr 1fr;  
}  
  
/\* Player Count Input \*/  
.player-count-group {  
 position: relative;  
}  
  
.player-count-note {  
 font-size: 12px;  
 color: var(--text-muted);  
 margin-top: 4px;  
 font-style: italic;  
}  
  
/\* Group Type Button \*/  
.group-type-controls {  
 display: flex;  
 align-items: center;  
 gap: 16px;  
 flex-wrap: wrap;  
}  
  
.group-type-btn {  
 display: flex;  
 align-items: center;  
 gap: 8px;  
 padding: 12px 20px;  
 border: 2px solid var(--primary);  
 border-radius: var(--radius);  
 background: linear-gradient(#1e90ff, #4fc3f7);  
 color: white;  
 font-size: 14px;  
 font-weight: 600;  
 cursor: pointer;  
 transition: var(--transition);  
 box-shadow: 0 2px 8px rgba(30, 144, 255, 0.25);  
}  
  
.group-type-btn:hover {  
 transform: translateY(-1px);  
   
 box-shadow: 0 4px 12px rgba(30, 144, 255, 0.35);  
}  
  
.group-type-btn.private {  
 background: linear-gradient(135deg, #64748b, #475569);  
 border-color: #64748b;  
 box-shadow: 0 2px 8px rgba(100, 116, 139, 0.25);  
}  
  
.group-type-btn.private:hover {  
 box-shadow: 0 4px 12px rgba(100, 116, 139, 0.35);  
}  
  
.password-input {  
 flex: 1;  
 min-width: 200px;  
 display: none;  
 animation: slideIn 0.3s ease;  
}  
  
.password-input.show {  
 display: block;  
}  
  
@keyframes slideIn {  
 from {  
 opacity: 0;  
 transform: translateX(-20px);  
 }  
 to {  
 opacity: 1;  
 transform: translateX(0);  
 }  
}  
  
/\* Flatpickr Custom Styles \*/  
.flatpickr-input {  
 width: 100%;  
 padding: 12px 16px;  
 border: 1px solid var(--card-border);  
 border-radius: var(--radius-small);  
 font-size: 14px;  
 transition: var(--transition);  
 background: rgba(255, 255, 255, 0.8);  
 color: var(--text-main);  
}  
  
.flatpickr-input:focus {  
 outline: none;  
 border-color: var(--primary);  
 box-shadow: 0 0 0 3px rgba(30, 144, 255, 0.1);  
 background: rgba(255, 255, 255, 0.95);  
}  
  
/\* Unavailable Slots Section \*/  
.unavailable-section {  
 background: rgba(239, 68, 68, 0.05);  
 border: 1px solid rgba(239, 68, 68, 0.2);  
 border-radius: var(--radius);  
 padding: 20px;  
 margin-top: 20px;  
 backdrop-filter: blur(5px);  
}  
  
.unavailable-header {  
 display: flex;  
 align-items: center;  
 gap: 12px;  
 margin-bottom: 16px;  
}  
  
.unavailable-icon {  
 width: 32px;  
 height: 32px;  
 background: rgba(239, 68, 68, 0.1);  
 border-radius: 50%;  
 display: flex;  
 align-items: center;  
 justify-content: center;  
 font-size: 16px;  
}  
  
.unavailable-title {  
 font-size: 16px;  
 font-weight: 600;  
 color: var(--danger);  
}  
  
.unavailable-slots {  
 display: grid;  
 grid-template-columns: repeat(auto-fit, minmax(100px, 1fr));  
 gap: 8px;  
}  
  
.unavailable-slot {  
 padding: 8px 12px;  
 background: rgba(239, 68, 68, 0.1);  
 border: 1px solid rgba(239, 68, 68, 0.2);  
 border-radius: var(--radius-small);  
 text-align: center;  
 font-size: 13px;  
 font-weight: 500;  
 color: var(--danger);  
}  
  
/\* Warning Section \*/  
.warning-section {  
 background: rgba(245, 158, 11, 0.1);  
 border: 1px solid rgba(245, 158, 11, 0.3);  
 border-radius: var(--radius);  
 padding: 20px;  
 margin-top: 20px;  
 backdrop-filter: blur(5px);  
}  
  
.warning-header {  
 display: flex;  
 align-items: center;  
 gap: 12px;  
 margin-bottom: 12px;  
}  
  
.warning-icon {  
 width: 32px;  
 height: 32px;  
 background: rgba(245, 158, 11, 0.2);  
 border-radius: 50%;  
 display: flex;  
 align-items: center;  
 justify-content: center;  
 font-size: 16px;  
}  
  
.warning-title {  
 font-size: 16px;  
 font-weight: 600;  
 color: var(--warning);  
}  
  
.warning-text {  
 color: var(--text-main);  
 font-size: 14px;  
 line-height: 1.5;  
}  
  
/\* Booking Summary \*/  
.booking-summary {  
 background: var(--card-bg);  
 border: 1px solid var(--card-border);  
 border-radius: var(--radius);  
 padding: 24px;  
 position: sticky;  
 top: 100px;  
 box-shadow: 0 4px 20px var(--shadow-light);  
 backdrop-filter: blur(10px);  
}  
  
.summary-title {  
 font-size: 20px;  
 font-weight: 600;  
 margin-bottom: 20px;  
 color: var(--text-main);  
}  
  
.summary-item {  
 display: flex;  
 justify-content: space-between;  
 align-items: center;  
 padding: 12px 0;  
 border-bottom: 1px solid rgba(0, 0, 0, 0.05);  
}  
  
.summary-item:last-child {  
 border-bottom: none;  
 font-weight: 600;  
 font-size: 18px;  
 padding-top: 16px;  
 margin-top: 8px;  
 border-top: 1px solid var(--card-border);  
}  
  
.summary-label {  
 color: var(--text-muted);  
 font-weight: 500;  
}  
  
.summary-value {  
 font-weight: 600;  
 color: var(--text-main);  
}  
  
.book-button {  
 width: 100%;  
 padding: 16px;  
 background: linear-gradient(135deg, var(--primary), var(--secondary));  
 color: white;  
 border: none;  
 border-radius: var(--radius);  
 font-size: 16px;  
 font-weight: 600;  
 cursor: pointer;  
 transition: var(--transition);  
 margin-top: 20px;  
 box-shadow: 0 4px 12px rgba(30, 144, 255, 0.25);  
}  
  
.book-button:hover {  
 transform: translateY(-2px);  
 box-shadow: 0 8px 25px rgba(30, 144, 255, 0.3);  
}  
  
.book-button:disabled {  
 opacity: 0.6;  
 cursor: not-allowed;  
 transform: none;  
 box-shadow: 0 4px 12px rgba(30, 144, 255, 0.15);  
}  
  
/\* Responsive Design \*/  
@media (max-width: 768px) {  
 .booking-layout {  
 grid-template-columns: 1fr;  
 gap: 24px;  
 }  
  
 .booking-summary {  
 position: relative;  
 top: 0;  
 order: -1;  
 }  
  
 .container {  
 padding: 24px 16px;  
 }  
  
 .form-row {  
 grid-template-columns: 1fr;  
 }  
  
 .form-row.three-columns {  
 grid-template-columns: 1fr;  
 }  
  
 .group-type-controls {  
 flex-direction: column;  
 align-items: stretch;  
 gap: 12px;  
 }  
  
 .password-input {  
 min-width: auto;  
 }  
  
 .fresh-navbar {  
 padding: 12px 16px;  
 }  
  
 .unavailable-slots {  
 grid-template-columns: repeat(2, 1fr);  
 }  
}  
  
@media (max-width: 480px) {  
 .venue-image {  
 height: 180px;  
 font-size: 36px;  
 }  
  
 .unavailable-slots {  
 grid-template-columns: 1fr;  
 }  
}

————————————————————————————————————————

## CreateBooking\CreateBooking.html

<!DOCTYPE html>  
<html lang="en">  
  
<head>  
 <meta charset="UTF-8">  
 <meta name="viewport" content="width=device-width, initial-scale=1.0">  
 <title>Book Venue - Book&Play</title>  
 <link href="https://fonts.googleapis.com/css2?family=Inter:wght@300;400;500;600;700&display=swap" rel="stylesheet">  
 <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/flatpickr/4.6.13/flatpickr.min.css">  
 <link rel="stylesheet" href="CreateBooking.css">  
 <link rel="stylesheet" href="../global.css">  
  
</head>  
  
<body>  
 <!-- Back Button -->  
 <div style="padding: 20px;">  
 <a href="../BookVenue/BookVenue.php" class="back-btn">  
 ← Back to Venues  
 </a>  
 </div>  
 <!-- Main Container -->  
 <div class="container">  
 <div class="booking-layout">  
 <!-- Left Column - Venue Info & Form -->  
 <div class="main-content">  
 <!-- Venue Info -->  
 <div class="venue-info">  
 <div class="venue-image">  
 <img src="images/default.jpg" alt="Venue Image">  
 </div>  
 <div class="venue-details">  
 <h2 class="venue-title">Elite Basketball Arena</h2>  
 <div class="venue-location">  
 📍 Downtown Sports Complex, City Center  
 </div>  
 <div class="venue-price">  
 <span class="per">per hour</span>  
 </div>  
 </div>  
 </div>  
  
 <!-- Booking Form -->  
 <div class="booking-form">  
 <form id="bookingForm">  
 <!-- Date & Time Selection -->  
 <div class="form-section">  
 <h3 class="section-title">Select Date & Time</h3>  
 <div class="form-row three-columns">  
 <div class="form-group">  
 <label class="form-label">Start Date</label>  
 <input type="text" class="flatpickr-input" id="startDate" placeholder="Select start date" required>  
 </div>  
 <div class="form-group">  
 <label class="form-label">Start Time</label>  
 <input type="text" class="flatpickr-input" id="startTime" placeholder="Select start time" required>  
 </div>  
 <div class="form-group">  
 <label class="form-label">End Time</label>  
 <input type="text" class="flatpickr-input" id="endTime" placeholder="Select end time" required>  
 </div>  
 </div>  
  
 <!-- Unavailable Time Slots -->  
 <div class="unavailable-section">  
 <div class="unavailable-header">  
 <div class="unavailable-icon">⚠️</div>  
 <div class="unavailable-title">Unavailable Time Slots</div>  
 </div>  
 <div class="unavailable-slots"></div>  
 </div>  
 </div>  
  
 <!-- Group Settings -->  
 <div class="form-section">  
 <h3 class="section-title">Group Settings</h3>  
  
 <div class="form-group">  
 <label class="form-label">Number of Players</label>  
 <div class="player-count-group">  
 <input type="number" class="form-input" id="playerCount" min="0" max="50"  
 placeholder="How many players will join? (Max 50)" required  
 oninput="this.value = this.value.slice(0,2)" />  
 <div class="player-count-note">Maximum 50 players</div>  
 </div>  
 </div>  
  
 <div class="form-group">  
 <label class="form-label">Group Type</label>  
 <div class="group-type-controls">  
 <button type="button" class="group-type-btn" id="groupTypeBtn" onclick="toggleGroupType()">  
 <span id="groupTypeIcon">🌐</span>  
 <span id="groupTypeText">Public Group</span>  
 </button>  
 <input type="password" class="form-input password-input" id="groupPassword"  
 placeholder="Enter group password" style="display: none;">  
  
 </div>  
 </div>  
 </div>  
  
 <!-- Payment Warning -->  
 <div class="warning-section">  
 <div class="warning-header">  
 <div class="warning-icon">⏰</div>  
 <div class="warning-title">Payment Reminder</div>  
 </div>  
 <div class="warning-text">  
 <strong>Important:</strong> At least 20% of the total payment must be completed within one hour of  
 booking confirmation. Failure to complete payment within this timeframe will result in automatic booking  
 cancellation.  
 </div>  
 </div>  
 </form>  
 </div>  
 </div>  
  
 <!-- Right Column - Booking Summary -->  
 <div class="booking-summary">  
 <h3 class="summary-title">Booking Summary</h3>  
  
 <div class="summary-content">  
 <div class="summary-item">  
 <span class="summary-label">Venue</span>  
 <span class="summary-value" id="summaryVenue">Elite Basketball Arena</span>  
 </div>  
 <div class="summary-item">  
 <span class="summary-label">Date</span>  
 <span class="summary-value" id="summaryDate">-</span>  
 </div>  
 <div class="summary-item">  
 <span class="summary-label">Start Time</span>  
 <span class="summary-value" id="summaryStartTime">-</span>  
 </div>  
 <div class="summary-item">  
 <span class="summary-label">End Time</span>  
 <span class="summary-value" id="summaryEndTime">-</span>  
 </div>  
 <div class="summary-item">  
 <span class="summary-label">Duration</span>  
 <span class="summary-value" id="summaryDuration">-</span>  
 </div>  
 <div class="summary-item">  
 <span class="summary-label">Players</span>  
 <span class="summary-value" id="summaryPlayers">50</span>  
 </div>  
 <div class="summary-item">  
 <span class="summary-label">Group Type</span>  
 <span class="summary-value" id="summaryGroupType">Public</span>  
 </div>  
 <div class="summary-item">  
 <span class="summary-label">Rate</span>  
 <span class="summary-value" id="summaryRate">₪120/hour</span>  
 </div>  
 <div class="summary-item">  
 <span class="summary-label">Total</span>  
 <span class="summary-value" id="summaryTotal">₪0</span>  
 </div>  
 </div>  
  
 <button type="submit" class="book-button" form="bookingForm">  
 Confirm Booking  
 </button>  
 </div>  
 </div>  
 </div>  
  
 <script src="https://cdnjs.cloudflare.com/ajax/libs/flatpickr/4.6.13/flatpickr.min.js"></script>  
 <script src="CreateBooking.js"></script>  
</body>  
  
</html>

————————————————————————————————————————

## CreateBooking\CreateBooking.js

let isPublic = true;  
let startTimePicker, endTimePicker, startDatePicker;  
let unavailableRanges = [];  
  
document.addEventListener('DOMContentLoaded', function () {  
 initializeDateTimePickers();  
 setupEventListeners();  
 updateSummary();  
});  
  
function initializeDateTimePickers() {  
 startDatePicker = flatpickr("#startDate", {  
 minDate: "today",  
 dateFormat: "Y-m-d",  
 onChange: ([selectedDate]) => {  
 updateEndTimeMinTime();  
 updateSummary();  
  
 if (facilityId && selectedDate) {  
 const formattedDate = selectedDate.getFullYear() + '-' +  
 String(selectedDate.getMonth() + 1).padStart(2, '0') + '-' +  
 String(selectedDate.getDate()).padStart(2, '0');  
 loadUnavailableRanges(facilityId, formattedDate);  
 }  
 }  
 });  
  
 startTimePicker = flatpickr("#startTime", {  
 enableTime: true,  
 noCalendar: true,  
 dateFormat: "H:i",  
 time\_24hr: true,  
 minuteIncrement: 30,  
 onChange: () => {  
 updateEndTimeMinTime();  
 updateSummary();  
 }  
 });  
  
 endTimePicker = flatpickr("#endTime", {  
 enableTime: true,  
 noCalendar: true,  
 dateFormat: "H:i",  
 time\_24hr: true,  
 minuteIncrement: 30,  
 onChange: updateSummary  
 });  
}  
  
function setupEventListeners() {  
 document.getElementById('playerCount').addEventListener('input', updateSummary);  
 document.getElementById('bookingForm').addEventListener('submit', handleFormSubmission);  
}  
  
const facilityId = new URLSearchParams(window.location.search).get("facility\_id");  
  
if (facilityId) {  
 fetch(`CreateBookingAPI.php?facility\_id=${facilityId}`)  
 .then(res => res.json())  
 .then(data => {  
 if (data.success) {  
 const facility = data.facility;  
 window.selectedFacilityPrice = facility.price;  
  
 document.querySelector(".venue-image img").src = `../../../uploads/venues/${facility.image\_url}` || "images/default.jpg";  
 document.querySelector(".venue-title").textContent = facility.place\_name;  
 document.querySelector(".venue-location").textContent = `📍 ${facility.location}`;  
 document.querySelector(".venue-price").innerHTML = `₪${facility.price} <span class="per">per hour</span>`;  
  
 document.getElementById("summaryVenue").textContent = facility.place\_name;  
 document.getElementById("summaryRate").textContent = `₪${facility.price}/hour`;  
 } else {  
 console.warn("Facility not found.");  
 }  
 })  
 .catch(err => console.error("Failed to fetch facility", err));  
}  
  
function toggleGroupType() {  
 const btn = document.getElementById('groupTypeBtn');  
 const icon = document.getElementById('groupTypeIcon');  
 const text = document.getElementById('groupTypeText');  
 const passwordInput = document.getElementById('groupPassword');  
  
 isPublic = !isPublic;  
  
 if (isPublic) {  
 btn.classList.remove('private');  
 icon.textContent = '🌐';  
 text.textContent = 'Public Group';  
 passwordInput.style.display = 'none';  
 } else {  
 btn.classList.add('private');  
 icon.textContent = '🔒';  
 text.textContent = 'Private Group';  
 passwordInput.style.display = 'block';  
 }  
 updateSummary();  
}  
  
function updateEndTimeMinTime() {  
 const startDate = document.getElementById('startDate').value;  
 const startTime = document.getElementById('startTime').value;  
  
 const now = new Date();  
 const todayStr = now.toISOString().split('T')[0];  
  
 if (startDate === todayStr) {  
 const currentTimeStr = now.getHours().toString().padStart(2, '0') + ':' +  
 Math.floor(now.getMinutes() / 30) \* 30  
 .toString()  
 .padStart(2, '0');  
 startTimePicker.set('minTime', currentTimeStr);  
 if (startTime < currentTimeStr) {  
 startTimePicker.setDate(currentTimeStr, true);  
 }  
 } else {  
 startTimePicker.set('minTime', '00:00');  
 }  
  
 if (startTime) {  
 endTimePicker.set('minTime', startTime);  
 }  
}  
  
function validateForm() {  
 const startDate = document.getElementById('startDate').value;  
 const startTime = document.getElementById('startTime').value;  
 const endTime = document.getElementById('endTime').value;  
  
 const now = new Date();  
 const selectedStartFull = new Date(`${startDate}T${startTime}`);  
 if (selectedStartFull < now) {  
 alert("Start time cannot be in the past.");  
 return false;  
 }  
  
 const playerCount = parseInt(document.getElementById('playerCount').value);  
  
 if (!startDate || !startTime || !endTime || isNaN(playerCount)) {  
 alert('Please fill in all required fields.');  
 return false;  
 }  
  
 if (playerCount > 50 || playerCount < 0) {  
 alert('Number of players must be between 0 and 50.');  
 return false;  
 }  
  
 if (startTime >= endTime) {  
 alert('End time must be after start time.');  
 return false;  
 }  
  
 const start = new Date(`2000-01-01T${startTime}`);  
 const end = new Date(`2000-01-01T${endTime}`);  
 const diffHours = (end - start) / (1000 \* 60 \* 60);  
  
 if (diffHours < 1) {  
 alert('The booking duration must be at least 1 hour.');  
 return false;  
 }  
  
 if (!isPublic && !document.getElementById('groupPassword').value) {  
 alert('Please enter a password for the private group.');  
 return false;  
 }  
  
 for (const range of unavailableRanges) {  
 const rangeStart = new Date(`2000-01-01T${range.from}`);  
 const rangeEnd = new Date(`2000-01-01T${range.to}`);  
  
 if (start < rangeEnd && end > rangeStart) {  
 alert(`Selected time ${startTime} → ${endTime} overlaps with unavailable slot ${range.from} → ${range.to}`);  
 return false;  
 }  
 }  
  
 return true;  
}  
  
  
function handleFormSubmission(e) {  
 e.preventDefault();  
  
 const button = document.querySelector('.book-button');  
 const originalText = button.textContent;  
  
 if (!validateForm()) return;  
  
 button.textContent = 'Processing...';  
 button.disabled = true;  
  
 const formData = new FormData();  
formData.append("facility\_id", facilityId);  
formData.append("start\_date", document.getElementById("startDate").value);  
formData.append("start\_time", document.getElementById("startTime").value);  
formData.append("end\_time", document.getElementById("endTime").value);  
formData.append("player\_count", document.getElementById("playerCount").value);  
formData.append("group\_type", isPublic ? "public" : "private");  
formData.append("group\_password", isPublic ? "" : document.getElementById("groupPassword").value);  
  
fetch("CreateBookingAPI.php", {  
 method: "POST",  
 body: formData  
})  
.then(res => res.json())  
.then(data => {  
 if (data.success) {  
 alert(`Booking confirmed successfully!\nYour Booking ID is: ${data.booking\_id}`);  
 setTimeout(() => {  
 window.location.href = "../MyBookings/MyBookings.php";  
 }, 2000);  
 } else {  
 alert("Error: " + data.message);  
 }  
})  
.catch(err => {  
 console.error("Booking error", err);  
 alert("An error occurred. Try again.");  
})  
.finally(() => {  
 button.textContent = originalText;  
 button.disabled = false;  
 button.style.background = 'linear-gradient(135deg, var(--primary), var(--secondary))';  
});  
  
}  
  
function updateSummary() {  
 const startDate = document.getElementById('startDate').value;  
 const startTime = document.getElementById('startTime').value;  
 const endTime = document.getElementById('endTime').value;  
 const playerCount = document.getElementById('playerCount').value;  
  
 if (startDate) {  
 const dateObj = new Date(startDate);  
 document.getElementById('summaryDate').textContent =  
 dateObj.toLocaleDateString('en-US', {  
 weekday: 'short',  
 month: 'short',  
 day: 'numeric',  
 year: 'numeric'  
 });  
 }  
  
 document.getElementById('summaryStartTime').textContent = startTime || '-';  
 document.getElementById('summaryEndTime').textContent = endTime || '-';  
  
 if (startTime && endTime) {  
 const start = new Date(`2000-01-01T${startTime}`);  
 const end = new Date(`2000-01-01T${endTime}`);  
 const diffHours = (end - start) / (1000 \* 60 \* 60);  
  
 if (diffHours > 0) {  
 document.getElementById('summaryDuration').textContent = `${diffHours} hour${diffHours !== 1 ? 's' : ''}`;  
 const price = window.selectedFacilityPrice || 0;  
document.getElementById('summaryTotal').textContent = `₪${diffHours \* price}`;  
 } else {  
 document.getElementById('summaryDuration').textContent = '-';  
 document.getElementById('summaryTotal').textContent = '₪0';  
 }  
 }  
  
 document.getElementById('summaryPlayers').textContent = playerCount || '0';  
 document.getElementById('summaryGroupType').textContent = isPublic ? 'Public' : 'Private';  
}  
  
function loadUnavailableRanges(facilityId, bookingDate) {  
 fetch(`CreateBookingAPI.php?facility\_id=${facilityId}&booking\_date=${bookingDate}`)  
 .then(res => res.json())  
 .then(data => {  
 const container = document.querySelector('.unavailable-slots');  
 container.innerHTML = '';  
 unavailableRanges = [];  
  
 if (data.success && Array.isArray(data.unavailable\_ranges)) {  
 unavailableRanges = mergeTimeRanges(data.unavailable\_ranges);  
 unavailableRanges.forEach(range => {  
 const slot = document.createElement('div');  
 slot.className = 'unavailable-slot';  
 slot.textContent = `${range.from} → ${range.to}`;  
 container.appendChild(slot);  
 });  
 updateTimePickersWithUnavailableTimes(unavailableRanges);  
 } else {  
 container.innerHTML = '<div class="unavailable-slot">No unavailable slots</div>';  
 updateTimePickersWithUnavailableTimes([]);  
 }  
 })  
 .catch(err => {  
 console.error("Failed to load unavailable ranges", err);  
 });  
}  
  
function mergeTimeRanges(ranges) {  
 if (!ranges.length) return [];  
  
 const sorted = ranges.slice().sort((a, b) => a.from.localeCompare(b.from));  
 const merged = [sorted[0]];  
  
 for (let i = 1; i < sorted.length; i++) {  
 const last = merged[merged.length - 1];  
 const current = sorted[i];  
  
 if (current.from <= last.to) {  
 last.to = current.to > last.to ? current.to : last.to;  
 } else {  
 merged.push(current);  
 }  
 }  
  
 return merged;  
}  
  
function updateTimePickersWithUnavailableTimes(ranges) {  
 const disabled = ranges.map(range => {  
 return {  
 from: `2000-01-01T${range.from}`,  
 to: `2000-01-01T${range.to}`  
 };  
 });  
  
 startTimePicker.set('disable', disabled);  
 endTimePicker.set('disable', disabled);  
}

————————————————————————————————————————

## CreateBooking\CreateBooking.php

<?php  
require\_once '../../../db.php';  
include 'CreateBooking.html'  
?>

————————————————————————————————————————

## CreateBooking\CreateBookingAPI.php

<?php  
session\_start();  
require\_once '../../../db.php';  
header('Content-Type: application/json');  
  
// Add debugging  
error\_log("=== CREATEBOOKINGAPI.PHP DEBUG START ===");  
error\_log("Request method: " . $\_SERVER['REQUEST\_METHOD']);  
error\_log("Session user\_id: " . ($\_SESSION['user\_id'] ?? 'null'));  
  
// Check database connection  
if ($conn->connect\_error) {  
 error\_log("❌ Database connection failed: " . $conn->connect\_error);  
 echo json\_encode(['success' => false, 'message' => 'Database connection failed']);  
 exit;  
}  
  
error\_log("✅ Database connection successful");  
  
if ($\_SERVER['REQUEST\_METHOD'] === 'POST') {  
 if (!isset($\_SESSION['user\_id'])) {  
 echo json\_encode(["success" => false, "message" => "User not logged in"]);  
 exit;  
 }  
  
 $username = $\_SESSION['user\_id'];  
 $facility\_id = $\_POST['facility\_id'] ?? null;  
 $start\_date = $\_POST['start\_date'] ?? null;  
 $start\_time = $\_POST['start\_time'] ?? null;  
 $end\_time = $\_POST['end\_time'] ?? null;  
 $player\_count = (int) ($\_POST['player\_count'] ?? 0);  
 $group\_type = $\_POST['group\_type'] ?? 'public';  
 $group\_password = trim($\_POST['group\_password'] ?? '');  
  
 if (!$facility\_id || !$start\_date || !$start\_time || !$end\_time) {  
 echo json\_encode(["success" => false, "message" => "Missing required fields"]);  
 exit;  
 }  
  
 // Get price from sportfacilities table  
 $priceQuery = $conn->prepare("SELECT price FROM sportfacilities WHERE facilities\_id = ?");  
 $priceQuery->bind\_param("i", $facility\_id);  
 $priceQuery->execute();  
 $priceResult = $priceQuery->get\_result();  
 $facility = $priceResult->fetch\_assoc();  
 $pricePerHour = $facility ? (int)$facility['price'] : 0;  
  
 // Calculate total price  
 $start = new DateTime($start\_time);  
 $end = new DateTime($end\_time);  
 $interval = $start->diff($end);  
 $hours = $interval->h + ($interval->i / 60);  
 $total\_price = $pricePerHour \* $hours;  
  
 // Create booking  
 $stmt = $conn->prepare("INSERT INTO bookings (facilities\_id, username, booking\_date, start\_time, end\_time, Total\_Price) VALUES (?, ?, ?, ?, ?, ?)");  
 $stmt->bind\_param("issssi", $facility\_id, $username, $start\_date, $start\_time, $end\_time, $total\_price);  
  
 if (!$stmt->execute()) {  
 echo json\_encode(["success" => false, "message" => "Failed to insert booking"]);  
 exit;  
 }  
  
 $booking\_id = $stmt->insert\_id;  
  
 if ($player\_count > 1) {  
 $group\_name = $username . "'s Group";  
 $privacy = $group\_type === 'public' ? 'public' : 'private';  
 $password\_to\_save = ($privacy === 'private') ? $group\_password : null;  
  
 $group\_stmt = $conn->prepare("INSERT INTO `groups` (group\_name, facilities\_id, created\_by, max\_members, booking\_id, privacy, group\_password) VALUES (?, ?, ?, ?, ?, ?, ?)");  
 $group\_stmt->bind\_param("sisiiss", $group\_name, $facility\_id, $username, $player\_count, $booking\_id, $privacy, $password\_to\_save);  
  
 if (!$group\_stmt->execute()) {  
 echo json\_encode(["success" => false, "message" => "Failed to insert group"]);  
 exit;  
 }  
  
 $group\_id = $group\_stmt->insert\_id;  
  
 // ⬇️ Add user to group  
 $add\_host\_stmt = $conn->prepare("INSERT INTO group\_members (group\_id, username, payment\_amount) VALUES (?, ?, 0)");  
 $add\_host\_stmt->bind\_param("is", $group\_id, $username);  
 $add\_host\_stmt->execute();  
}  
  
  
 echo json\_encode([  
 "success" => true,  
 "booking\_id" => $booking\_id  
]);  
  
 exit;  
}  
  
// ======================= GET =======================  
  
$facility\_id = $\_GET['facility\_id'] ?? null;  
$booking\_date = $\_GET['booking\_date'] ?? null;  
  
error\_log("GET request - facility\_id: " . $facility\_id . ", booking\_date: " . $booking\_date);  
  
if (!$facility\_id) {  
 error\_log("❌ Missing facility\_id in GET request");  
 echo json\_encode(["success" => false, "message" => "Missing facility\_id"]);  
 exit;  
}  
  
if ($booking\_date) {  
 error\_log("Fetching unavailable ranges for date: " . $booking\_date);  
 $stmt = $conn->prepare("SELECT start\_time, end\_time FROM bookings WHERE facilities\_id = ? AND booking\_date = ?");  
 $stmt->bind\_param("is", $facility\_id, $booking\_date);  
 $stmt->execute();  
 $result = $stmt->get\_result();  
  
 $unavailable = [];  
 while ($row = $result->fetch\_assoc()) {  
 $unavailable[] = [  
 'from' => substr($row['start\_time'], 0, 5),  
 'to' => substr($row['end\_time'], 0, 5)  
 ];  
 }  
  
 error\_log("✅ Found " . count($unavailable) . " unavailable ranges");  
 echo json\_encode(["success" => true, "unavailable\_ranges" => $unavailable]);  
 exit;  
}  
  
error\_log("Fetching facility details for ID: " . $facility\_id);  
$stmt = $conn->prepare("SELECT \* FROM sportfacilities WHERE facilities\_id = ?");  
$stmt->bind\_param("i", $facility\_id);  
$stmt->execute();  
$result = $stmt->get\_result();  
  
if ($facility = $result->fetch\_assoc()) {  
 error\_log("✅ Facility found: " . $facility['place\_name']);  
 echo json\_encode(["success" => true, "facility" => $facility]);  
} else {  
 error\_log("❌ Facility not found for ID: " . $facility\_id);  
 echo json\_encode(["success" => false, "message" => "Facility not found"]);  
}  
  
error\_log("=== CREATEBOOKINGAPI.PHP DEBUG END ===");  
?>

————————————————————————————————————————

## Favorites\Favorites.css

/\* Color Variables \*/  
:root {  
 --primary: #1e90ff;  
 --secondary: #22c55e;  
 --bg-light: #e0f7fa;  
 --bg-secondary: #d0f8ce;  
 --card-bg: rgba(255, 255, 255, 0.95);  
 --card-border: rgba(209, 213, 219, 0.6);  
 --text-main: #0f172a;  
 --text-muted: #64748b;  
 --text-light: #94a3b8;  
 --shadow-light: rgba(30, 144, 255, 0.08);  
 --shadow-medium: rgba(0, 0, 0, 0.08);  
 --shadow-strong: rgba(0, 0, 0, 0.15);  
 --radius: 12px;  
 --radius-medium: 10px;  
 --radius-small: 8px;  
 --transition: all 0.3s cubic-bezier(0.4, 0, 0.2, 1);  
 --border-subtle: rgba(0, 0, 0, 0.06);  
 --navbar-height: 80px;  
}  
  
/\* Reset \*/  
\* {  
 margin: 0;  
 padding: 0;  
 box-sizing: border-box;  
}  
  
body {  
 font-family: 'Inter', sans-serif;  
 background: linear-gradient(135deg, var(--bg-light), var(--bg-secondary));  
 color: var(--text-main);  
 min-height: 100vh;  
 line-height: 1.6;  
 -webkit-font-smoothing: antialiased;  
 padding-top: var(--navbar-height);  
}  
  
.container {  
 max-width: 1200px;  
 margin: 0 auto;  
 padding: 24px;  
}  
  
.header {  
 text-align: center;  
 margin-bottom: 48px;  
 color: var(--text-main);  
}  
  
.title {  
 font-size: 2.5rem;  
 font-weight: 700;  
 margin-bottom: 12px;  
 background: linear-gradient(135deg, var(--primary), var(--secondary));  
 -webkit-background-clip: text;  
 -webkit-text-fill-color: transparent;  
 text-shadow: none;  
}  
  
.subtitle {  
 font-size: 1.1rem;  
 color: var(--text-muted);  
 opacity: 0.9;  
}  
  
.favorites-grid {  
 display: grid;  
 grid-template-columns: repeat(auto-fit, minmax(320px, 1fr));  
 gap: 24px;  
 margin-bottom: 40px;  
}  
  
.venue-card {  
 background: var(--card-bg);  
 border-radius: var(--radius);  
 box-shadow: 0 2px 8px var(--shadow-light), 0 1px 3px var(--shadow-medium);  
 border: 1px solid var(--card-border);  
 overflow: hidden;  
 transition: var(--transition);  
 position: relative;  
 backdrop-filter: blur(12px);  
}  
  
.venue-card:hover {  
 transform: translateY(-4px);  
 box-shadow: 0 8px 25px var(--shadow-light), 0 4px 12px var(--shadow-medium);  
 border-color: rgba(30, 144, 255, 0.2);  
}  
  
.venue-image {  
 width: 100%;  
 height: 200px;  
 object-fit: cover;  
 background: linear-gradient(45deg, var(--bg-light), var(--bg-secondary));  
}  
  
.favorite-icon {  
 position: absolute;  
 top: 15px;  
 right: 15px;  
 background: var(--card-bg);  
 border-radius: 50%;  
 width: 36px;  
 height: 36px;  
 display: flex;  
 align-items: center;  
 justify-content: center;  
 font-size: 18px;  
 box-shadow: 0 2px 8px var(--shadow-light);  
 border: 1px solid var(--border-subtle);  
 cursor: pointer;  
 transition: var(--transition);  
 backdrop-filter: blur(8px);  
}  
  
.favorite-icon:hover {  
 transform: translateY(-1px) scale(1.05);  
 background: linear-gradient(135deg, var(--primary), var(--secondary));  
 color: white;  
 border-color: transparent;  
 box-shadow: 0 4px 12px rgba(30, 144, 255, 0.3);  
}  
  
.card-content {  
 padding: 24px;  
}  
  
.venue-name {  
 font-size: 1.4rem;  
 font-weight: 600;  
 margin-bottom: 8px;  
 color: var(--text-main);  
}  
  
.sport-type {  
 display: inline-block;  
 background: linear-gradient(135deg, var(--primary), var(--secondary));  
 color: white;  
 padding: 6px 14px;  
 border-radius: 20px;  
 font-size: 0.85rem;  
 font-weight: 500;  
 margin-bottom: 12px;  
 box-shadow: 0 2px 4px rgba(30, 144, 255, 0.2);  
}  
  
.location {  
 color: var(--text-muted);  
 font-size: 0.95rem;  
 margin-bottom: 12px;  
 display: flex;  
 align-items: center;  
 gap: 6px;  
}  
  
.rating {  
 display: flex;  
 align-items: center;  
 gap: 8px;  
 margin-bottom: 16px;  
}  
  
.stars {  
 color: #f59e0b;  
 font-size: 1.1rem;  
}  
  
.rating-text {  
 color: var(--text-light);  
 font-size: 0.9rem;  
}  
  
.view-details-btn {  
 background: linear-gradient(135deg, var(--primary), var(--secondary));  
 color: white;  
 border: none;  
 padding: 12px 24px;  
 border-radius: 24px;  
 font-weight: 600;  
 cursor: pointer;  
 transition: var(--transition);  
 width: 100%;  
 font-size: 1rem;  
 box-shadow: 0 2px 8px rgba(30, 144, 255, 0.2);  
}  
  
.view-details-btn:hover {  
 transform: translateY(-2px);  
 box-shadow: 0 8px 25px rgba(30, 144, 255, 0.4), 0 4px 12px var(--shadow-medium);  
}  
  
.empty-state {  
 text-align: center;  
 padding: 80px 20px;  
 color: var(--text-main);  
 background: var(--card-bg);  
 border-radius: var(--radius);  
 box-shadow: 0 2px 8px var(--shadow-light);  
 border: 1px solid var(--card-border);  
 backdrop-filter: blur(12px);  
}  
  
.empty-icon {  
 font-size: 4rem;  
 margin-bottom: 24px;  
 background: linear-gradient(135deg, var(--primary), var(--secondary));  
 -webkit-background-clip: text;  
 -webkit-text-fill-color: transparent;  
}  
  
.empty-state h3 {  
 font-size: 1.8rem;  
 margin-bottom: 12px;  
 font-weight: 600;  
 color: var(--text-main);  
}  
  
.empty-state p {  
 font-size: 1.1rem;  
 color: var(--text-muted);  
 opacity: 0.8;  
}  
  
/\* NAVBAR \*/  
.fresh-navbar {  
 position: fixed;  
 top: 0;  
 left: 0;  
 right: 0;  
 height: var(--navbar-height);  
 z-index: 1000;  
 display: flex;  
 justify-content: space-between;  
 align-items: center;  
 padding: 12px 24px;  
 background: linear-gradient(135deg, rgba(224, 247, 250, 0.95), rgba(208, 248, 206, 0.95));  
 backdrop-filter: blur(12px);  
 border-bottom: 1px solid var(--border-subtle);  
 box-shadow: 0 1px 3px var(--shadow-light), 0 1px 2px var(--shadow-medium);  
}  
  
.logo {  
 font-weight: bold;  
 font-size: 20px;  
 background: linear-gradient(135deg, var(--primary), var(--secondary));  
 -webkit-background-clip: text;  
 -webkit-text-fill-color: transparent;  
}  
  
.nav-wrapper {  
 display: flex;  
 align-items: center;  
}  
  
.nav-links {  
 list-style: none;  
 display: flex;  
 gap: 8px;  
 align-items: center;  
}  
  
.nav-links li a {  
 padding: 10px 20px;  
 border-radius: 24px;  
 font-size: 14px;  
 font-weight: 500;  
 text-decoration: none;  
 color: var(--text-main);  
 background: var(--card-bg);  
 border: 1px solid var(--border-subtle);  
 transition: var(--transition);  
 box-shadow: 0 1px 3px var(--shadow-light);  
}  
  
.nav-links li a:hover {  
 transform: translateY(-1px);  
 box-shadow: 0 4px 12px var(--shadow-light);  
 background: linear-gradient(to right, #4fc3f7, #81c784);  
 color: white;  
 border-color: transparent;  
}  
  
.nav-links li a.active {  
 color: white;  
 background: linear-gradient(to right, #4fc3f7, #81c784);  
 font-weight: 600;  
 border-color: transparent;  
 box-shadow: 0 2px 8px rgba(30, 144, 255, 0.25);  
}  
  
.nav-links li a.logout {  
 background: rgba(239, 68, 68, 0.1);  
 color: #dc2626;  
 font-weight: 600;  
 border-color: rgba(239, 68, 68, 0.2);  
}  
  
.nav-links li a.logout:hover {  
 background: #dc2626;  
 color: white;  
 border-color: #dc2626;  
}  
  
/\* SPORTS SCROLL \*/  
.sports-scroll {  
 display: flex;  
 gap: 12px;  
 overflow-x: auto;  
 padding: 16px 24px;  
 background: linear-gradient(135deg, rgba(224, 247, 250, 0.4), rgba(208, 248, 206, 0.4));  
 scrollbar-width: thin;  
 scrollbar-color: var(--primary) transparent;  
}  
  
.sports-scroll::-webkit-scrollbar {  
 height: 6px;  
}  
  
.sports-scroll::-webkit-scrollbar-thumb {  
 background: var(--primary);  
 border-radius: 3px;  
}  
  
.sport-item {  
 flex-shrink: 0;  
 position: relative;  
 display: inline-block;  
 cursor: pointer;  
}  
  
.sport-checkbox {  
 display: none;  
}  
  
.sport-circle {  
 width: 70px;  
 height: 70px;  
 border-radius: 50%;  
 background: var(--card-bg);  
 font-size: 11px;  
 font-weight: 500;  
 text-align: center;  
 display: flex;  
 flex-direction: column;  
 align-items: center;  
 justify-content: center;  
 gap: 3px;  
 box-shadow: 0 2px 8px rgba(0,0,0,0.1);  
 transition: var(--transition);  
 border: 2px solid transparent;  
}  
  
.sport-circle:hover {  
 transform: translateY(-2px);  
 background: rgba(30, 144, 255, 0.05);  
 border-color: rgba(30, 144, 255, 0.2);  
}  
  
.sport-checkbox:checked + .sport-circle {  
 background: linear-gradient(135deg, var(--primary), var(--secondary));  
 color: white;  
 box-shadow: 0 4px 12px rgba(30, 144, 255, 0.3);  
 border-color: rgba(30, 144, 255, 0.4);  
 transform: translateY(-2px);  
 position: relative;  
}  
  
.sport-checkbox:checked + .sport-circle::after {  
 content: "✓";  
 position: absolute;  
 top: 1px;  
 right: 25px;  
 font-size: 16px;  
 font-weight: bold;  
 color: white;  
}  
  
@media (max-width: 768px) {  
 .container {  
 padding: 16px;  
 }  
   
 .title {  
 font-size: 2rem;  
 }  
   
 .favorites-grid {  
 grid-template-columns: 1fr;  
 gap: 20px;  
 }  
   
 .venue-card {  
 margin: 0 auto;  
 max-width: 400px;  
 }  
   
 .fresh-navbar {  
 padding: 8px 16px;  
 }  
   
 .sports-scroll {  
 padding: 12px 16px;  
 }  
}

————————————————————————————————————————

## Favorites\Favorites.html

<!DOCTYPE html>  
<html lang="en">  
<head>  
 <meta charset="UTF-8">  
 <meta name="viewport" content="width=device-width, initial-scale=1.0">  
 <title>My Favorites - Book&Play</title>  
 <link rel="stylesheet" href="Favorites.css">  
</head>  
<body>  
 <div class="container">  
 <header class="header">  
 <h1 class="title">My Favorites ❤️</h1>  
 <p class="subtitle">Your most loved venues</p>  
 </header>  
  
 <div class="favorites-grid" id="favoritesGrid">  
 <!-- Venue cards will be loaded here dynamically -->  
 </div>  
  
 <div class="empty-state" id="emptyState" style="display: none;">  
 <div class="empty-icon">💔</div>  
 <h3>No favorites yet</h3>  
 <p>Start exploring venues and add them to your favorites!</p>  
 </div>  
 </div>  
  
 <script src="Favorites.js"></script>  
</body>  
</html>

————————————————————————————————————————

## Favorites\Favorites.js

// Sample favorite venues data  
const favoriteVenues = [  
 {  
 id: 1,  
 name: "Elite Sports Complex",  
 sport: "Basketball",  
 location: "Downtown Plaza",  
 rating: 4.8,  
 image: "https://images.unsplash.com/photo-1546519638-68e109498ffc?w=400&h=200&fit=crop",  
 isFavorite: true  
 },  
 {  
 id: 2,  
 name: "Ocean View Tennis Club",  
 sport: "Tennis",  
 location: "Marina District",  
 rating: 4.6,  
 image: "https://images.unsplash.com/photo-1622279457486-62dcc4a431d6?w=400&h=200&fit=crop",  
 isFavorite: true  
 },  
 {  
 id: 3,  
 name: "City Football Arena",  
 sport: "Football",  
 location: "Sports Center",  
 rating: 4.9,  
 image: "https://images.unsplash.com/photo-1431324155629-1a6deb1dec8d?w=400&h=200&fit=crop",  
 isFavorite: true  
 },  
 {  
 id: 4,  
 name: "Aqua Fitness Center",  
 sport: "Swimming",  
 location: "Health District",  
 rating: 4.7,  
 image: "https://images.unsplash.com/photo-1530549387789-4c1017266635?w=400&h=200&fit=crop",  
 isFavorite: true  
 }  
];  
  
// Generate star rating HTML  
function generateStars(rating) {  
 const fullStars = Math.floor(rating);  
 const hasHalfStar = rating % 1 !== 0;  
 let starsHtml = '';  
   
 for (let i = 0; i < fullStars; i++) {  
 starsHtml += '★';  
 }  
   
 if (hasHalfStar) {  
 starsHtml += '☆';  
 }  
   
 return starsHtml;  
}  
  
// Create venue card HTML  
function createVenueCard(venue) {  
 return `  
 <div class="venue-card">  
 <img src="${venue.image}" alt="${venue.name}" class="venue-image">  
 <div class="favorite-icon" onclick="toggleFavorite(${venue.id})">  
 ❤️  
 </div>  
 <div class="card-content">  
 <h3 class="venue-name">${venue.name}</h3>  
 <span class="sport-type">${venue.sport}</span>  
 <div class="location">📍 ${venue.location}</div>  
 <div class="rating">  
 <span class="stars">${generateStars(venue.rating)}</span>  
 <span class="rating-text">${venue.rating}/5.0</span>  
 </div>  
 <button class="view-details-btn" onclick="window.location.href='../VenueDetails/VenueDetails.php'">  
 View Details  
</button>  
  
 </div>  
 </div>  
 `;  
}  
  
// Load and display favorite venues  
function loadFavorites() {  
 const grid = document.getElementById('favoritesGrid');  
 const emptyState = document.getElementById('emptyState');  
   
 if (favoriteVenues.length === 0) {  
 grid.style.display = 'none';  
 emptyState.style.display = 'block';  
 } else {  
 grid.style.display = 'grid';  
 emptyState.style.display = 'none';  
   
 grid.innerHTML = favoriteVenues  
 .map(venue => createVenueCard(venue))  
 .join('');  
 }  
}  
  
// Toggle favorite status  
function toggleFavorite(venueId) {  
 const venueIndex = favoriteVenues.findIndex(v => v.id === venueId);  
   
 if (venueIndex !== -1) {  
 // Remove from favorites with smooth animation  
 const card = document.querySelector(`[onclick="toggleFavorite(${venueId})"]`).closest('.venue-card');  
 card.style.opacity = '0';  
 card.style.transform = 'scale(0.8)';  
   
 setTimeout(() => {  
 favoriteVenues.splice(venueIndex, 1);  
 loadFavorites();  
 }, 300);  
 }  
}  
  
// View venue details  
function viewDetails(venueId) {  
 const venue = favoriteVenues.find(v => v.id === venueId);  
 if (venue) {  
 alert(`Viewing details for: ${venue.name}\n\nThis would normally navigate to the venue details page.`);  
 // In a real app, you would navigate to: window.location.href = `venue-details.html?id=${venueId}`;  
 }  
}  
  
// Initialize the page  
document.addEventListener('DOMContentLoaded', function() {  
 loadFavorites();  
});  
  
// Add some interactive effects  
document.addEventListener('DOMContentLoaded', function() {  
 // Add entrance animation to cards  
 setTimeout(() => {  
 const cards = document.querySelectorAll('.venue-card');  
 cards.forEach((card, index) => {  
 card.style.opacity = '0';  
 card.style.transform = 'translateY(20px)';  
   
 setTimeout(() => {  
 card.style.transition = 'all 0.5s ease';  
 card.style.opacity = '1';  
 card.style.transform = 'translateY(0)';  
 }, index \* 100);  
 });  
 }, 100);  
});

————————————————————————————————————————

## Favorites\Favorites.php

<?php  
session\_start();  
$\_SESSION['previous\_page'] = $\_SERVER['PHP\_SELF'];  
include '../navbar.php';  
include 'Favorites.html';  
?>

————————————————————————————————————————

## FindPlayer\FindPlayer.css

/\* FindPlayer.css - Matching Global Style \*/  
  
/\* Import global variables \*/  
:root {  
 --primary: #1e90ff;  
 --secondary: #22c55e;  
 --bg-light: #e0f7fa;  
 --bg-secondary: #d0f8ce;  
 --card-bg: rgba(255, 255, 255, 0.95);  
 --card-border: rgba(209, 213, 219, 0.6);  
 --text-main: #0f172a;  
 --text-muted: #64748b;  
 --text-light: #94a3b8;  
 --shadow-light: rgba(30, 144, 255, 0.08);  
 --shadow-medium: rgba(0, 0, 0, 0.08);  
 --shadow-strong: rgba(0, 0, 0, 0.15);  
 --radius: 12px;  
 --radius-medium: 10px;  
 --radius-small: 8px;  
 --transition: all 0.3s cubic-bezier(0.4, 0, 0.2, 1);  
 --border-subtle: rgba(0, 0, 0, 0.06);  
 --navbar-height: 80px;  
}  
  
\* {  
 margin: 0;  
 padding: 0;  
 box-sizing: border-box;  
}  
  
body {  
 font-family: 'Inter', sans-serif;  
 background: linear-gradient(135deg, var(--bg-light), var(--bg-secondary));  
 background-attachment: fixed;  
 background-size: 100% 100%;  
 background-repeat: no-repeat;  
 color: var(--text-main);  
 min-height: 100vh;  
 line-height: 1.6;  
 -webkit-font-smoothing: antialiased;  
 padding-top: var(--navbar-height);  
}  
  
.container {  
 max-width: 1200px;  
 margin: 0 auto;  
 padding: 24px;  
}  
  
/\* Search Section \*/  
.search-section {  
 display: flex;  
 gap: 16px;  
 margin-bottom: 32px;  
 align-items: center;  
}  
  
.search-bar {  
 flex: 1;  
 position: relative;  
 background: var(--card-bg);  
 border-radius: 24px;  
 padding: 4px;  
 box-shadow: 0 2px 8px var(--shadow-light), 0 1px 3px var(--shadow-medium);  
 border: 1px solid var(--border-subtle);  
 backdrop-filter: blur(12px);  
}  
  
.search-bar input {  
 width: calc(100% - 50px);  
 padding: 14px 20px;  
 border: none;  
 outline: none;  
 background: transparent;  
 font-size: 15px;  
 color: var(--text-main);  
 font-weight: 500;  
}  
  
.search-bar input::placeholder {  
 color: var(--text-light);  
}  
  
.search-btn {  
 position: absolute;  
 right: 4px;  
 top: 50%;  
 transform: translateY(-50%);  
 background: linear-gradient(135deg, var(--primary), var(--secondary));  
 border: none;  
 border-radius: 50%;  
 width: 40px;  
 height: 40px;  
 cursor: pointer;  
 font-size: 16px;  
 transition: var(--transition);  
 box-shadow: 0 2px 8px rgba(30, 144, 255, 0.25);  
 color: white;  
}  
  
.search-btn:hover {  
 transform: translateY(-50%) scale(1.05);  
 box-shadow: 0 4px 12px rgba(30, 144, 255, 0.3);  
}  
  
.sort-btn, .friends-btn {  
 background: linear-gradient(135deg, var(--primary), var(--secondary));  
 color: white;  
 border: none;  
 padding: 14px 24px;  
 border-radius: 24px;  
 cursor: pointer;  
 font-weight: 600;  
 font-size: 14px;  
 transition: var(--transition);  
 box-shadow: 0 2px 8px rgba(30, 144, 255, 0.25);  
}  
  
.sort-btn:hover, .friends-btn:hover {  
 transform: translateY(-1px);  
 box-shadow: 0 4px 12px rgba(30, 144, 255, 0.3);  
}  
  
/\* Availability Section \*/  
.availability-section {  
 background: var(--card-bg);  
 border-radius: var(--radius);  
 padding: 24px;  
 margin-bottom: 32px;  
 box-shadow: 0 2px 8px var(--shadow-light), 0 1px 3px var(--shadow-medium);  
 border: 1px solid var(--border-subtle);  
 backdrop-filter: blur(12px);  
}  
  
.availability-header {  
 display: flex;  
 justify-content: space-between;  
 align-items: center;  
 margin-bottom: 24px;  
 flex-wrap: wrap;  
 gap: 16px;  
}  
  
.availability-controls {  
 display: flex;  
 align-items: center;  
 gap: 16px;  
 flex-wrap: wrap;  
}  
  
.select-all-btn {  
 background: linear-gradient(135deg, var(--secondary), #16a34a);  
 color: white;  
 border: none;  
 padding: 10px 16px;  
 border-radius: 20px;  
 cursor: pointer;  
 font-weight: 600;  
 font-size: 13px;  
 transition: var(--transition);  
 box-shadow: 0 2px 8px rgba(34, 197, 94, 0.25);  
}  
  
.select-all-btn:hover {  
 transform: translateY(-1px);  
 box-shadow: 0 4px 12px rgba(34, 197, 94, 0.3);  
}  
  
.availability-header h3 {  
 color: var(--text-main);  
 font-size: 20px;  
 font-weight: 600;  
}  
  
.toggle-container {  
 display: flex;  
 align-items: center;  
 gap: 12px;  
 font-weight: 500;  
 color: var(--text-muted);  
 font-size: 14px;  
}  
  
.toggle {  
 position: relative;  
 display: inline-block;  
 width: 54px;  
 height: 30px;  
}  
  
.toggle input {  
 opacity: 0;  
 width: 0;  
 height: 0;  
}  
  
.slider {  
 position: absolute;  
 cursor: pointer;  
 top: 0;  
 left: 0;  
 right: 0;  
 bottom: 0;  
 background: var(--text-light);  
 transition: var(--transition);  
 border-radius: 30px;  
}  
  
.slider:before {  
 position: absolute;  
 content: "";  
 height: 24px;  
 width: 24px;  
 left: 3px;  
 bottom: 3px;  
 background: white;  
 transition: var(--transition);  
 border-radius: 50%;  
 box-shadow: 0 2px 8px var(--shadow-medium);  
}  
  
input:checked + .slider {  
 background: linear-gradient(135deg, var(--primary), var(--secondary));  
}  
  
input:checked + .slider:before {  
 transform: translateX(24px);  
}  
  
.days-container {  
 display: grid;  
 grid-template-columns: repeat(auto-fit, minmax(300px, 1fr));  
 gap: 16px;  
}  
  
.day-card {  
 background: rgba(30, 144, 255, 0.03);  
 border-radius: var(--radius);  
 padding: 20px;  
 border: 1px solid var(--border-subtle);  
 transition: var(--transition);  
}  
  
.day-card.active {  
 border-color: var(--primary);  
 background: rgba(30, 144, 255, 0.08);  
 box-shadow: 0 2px 8px rgba(30, 144, 255, 0.15);  
}  
  
.day-header {  
 display: flex;  
 align-items: center;  
 gap: 10px;  
 margin-bottom: 16px;  
}  
  
.day-header input[type="checkbox"] {  
 width: 16px;  
 height: 16px;  
 accent-color: var(--primary);  
}  
  
.day-header label {  
 font-weight: 600;  
 cursor: pointer;  
 color: var(--text-main);  
 font-size: 15px;  
}  
  
.time-slots {  
 display: flex;  
 flex-direction: column;  
 gap: 12px;  
}  
  
.time-slot {  
 display: flex;  
 align-items: center;  
 gap: 12px;  
 background: var(--card-bg);  
 padding: 12px 16px;  
 border-radius: var(--radius-small);  
 border: 1px solid var(--border-subtle);  
 box-shadow: 0 1px 3px var(--shadow-light);  
}  
  
.time-slot input {  
 border: none;  
 outline: none;  
 background: transparent;  
 font-size: 14px;  
 color: var(--text-main);  
 flex: 1;  
 font-weight: 500;  
 min-width: 80px;  
}  
  
.time-slot input[type="time"] {  
 -webkit-appearance: none;  
 -moz-appearance: textfield;  
}  
  
.time-slot input[type="time"]::-webkit-calendar-picker-indicator {  
 color: var(--primary);  
 cursor: pointer;  
}  
  
.time-slot .delete-slot {  
 background: rgba(239, 68, 68, 0.1);  
 color: #dc2626;  
 border: none;  
 border-radius: 50%;  
 width: 24px;  
 height: 24px;  
 cursor: pointer;  
 font-size: 14px;  
 font-weight: 600;  
 transition: var(--transition);  
 display: flex;  
 align-items: center;  
 justify-content: center;  
}  
  
.time-slot .delete-slot:hover {  
 background: #dc2626;  
 color: white;  
}  
  
.add-slot-btn {  
 background: linear-gradient(135deg, var(--secondary), #16a34a);  
 color: white;  
 border: none;  
 padding: 10px 16px;  
 border-radius: var(--radius-small);  
 cursor: pointer;  
 font-size: 13px;  
 font-weight: 500;  
 margin-top: 8px;  
 transition: var(--transition);  
 box-shadow: 0 2px 8px rgba(34, 197, 94, 0.25);  
}  
  
.add-slot-btn:hover {  
 transform: translateY(-1px);  
 box-shadow: 0 4px 12px rgba(34, 197, 94, 0.3);  
}  
  
/\* Players Section \*/  
.players-section h3 {  
 color: var(--text-main);  
 font-size: 22px;  
 font-weight: 600;  
 margin-bottom: 24px;  
}  
  
.players-grid {  
 display: grid;  
 grid-template-columns: repeat(auto-fill, minmax(280px, 1fr));  
 gap: 20px;  
}  
  
.player-card {  
 background: var(--card-bg);  
 border-radius: var(--radius);  
 padding: 0;  
 box-shadow: 0 2px 8px var(--shadow-light), 0 1px 3px var(--shadow-medium);  
 border: 1px solid var(--border-subtle);  
 transition: var(--transition);  
 backdrop-filter: blur(12px);  
 overflow: hidden;  
 display: flex;  
 flex-direction: column;  
 min-height: 320px;  
 cursor: pointer;  
}  
  
.player-card:hover {  
 transform: translateY(-4px);  
 box-shadow: 0 8px 20px var(--shadow-light), 0 4px 12px var(--shadow-medium);  
}  
  
.player-image {  
 width: 100%;  
 height: 160px;  
 background: #f8fafc;  
 display: flex;  
 align-items: center;  
 justify-content: center;  
 overflow: hidden;  
 position: relative;  
}  
  
.player-image img {  
 max-width: 100%;  
 max-height: 100%;  
 width: auto;  
 height: auto;  
 object-fit: scale-down;  
 object-position: center;  
 border-radius: 8px;  
 box-shadow: 0 2px 8px rgba(0, 0, 0, 0.1);  
}  
  
.player-details {  
 padding: 20px;  
 text-align: center;  
 flex: 1;  
 display: flex;  
 flex-direction: column;  
 justify-content: space-between;  
}  
  
.player-name {  
 font-size: 18px;  
 font-weight: 600;  
 color: var(--text-main);  
 margin-bottom: 8px;  
}  
  
.player-sport {  
 font-size: 14px;  
 color: var(--text-muted);  
 margin-bottom: 4px;  
}  
  
.player-location {  
 font-size: 13px;  
 color: var(--text-light);  
 margin-bottom: 16px;  
}  
  
.player-actions {  
 display: flex;  
 gap: 8px;  
 margin-top: 12px;  
}  
  
.add-friend-btn {  
 flex: 1;  
 background: linear-gradient(135deg, var(--secondary), #16a34a);  
 color: white;  
 border: none;  
 padding: 10px 16px;  
 border-radius: var(--radius-small);  
 cursor: pointer;  
 font-weight: 600;  
 font-size: 12px;  
 transition: var(--transition);  
 box-shadow: 0 2px 8px rgba(34, 197, 94, 0.25);  
}  
  
.add-friend-btn:hover {  
 transform: translateY(-1px);  
 box-shadow: 0 4px 12px rgba(34, 197, 94, 0.3);  
}  
  
.add-friend-btn.added {  
 background: var(--text-light);  
 color: white;  
 cursor: not-allowed;  
}  
  
.connect-btn {  
 flex: 1;  
 background: linear-gradient(135deg, var(--primary), var(--secondary));  
 color: white;  
 border: none;  
 padding: 10px 16px;  
 border-radius: var(--radius-small);  
 cursor: pointer;  
 font-weight: 600;  
 font-size: 12px;  
 transition: var(--transition);  
 box-shadow: 0 2px 8px rgba(30, 144, 255, 0.25);  
}  
  
.connect-btn:hover {  
 transform: translateY(-1px);  
 box-shadow: 0 4px 12px rgba(30, 144, 255, 0.3);  
}  
  
/\* Modal Styles \*/  
.modal {  
 display: none;  
 position: fixed;  
 z-index: 1000;  
 left: 0;  
 top: 0;  
 width: 100%;  
 height: 100%;  
 background-color: rgba(0, 0, 0, 0.5);  
 backdrop-filter: blur(8px);  
}  
  
.modal-content {  
 background: var(--card-bg);  
 margin: 3% auto;  
 border-radius: var(--radius);  
 width: 90%;  
 max-width: 500px;  
 position: relative;  
 box-shadow: 0 20px 60px var(--shadow-strong);  
 border: 1px solid var(--border-subtle);  
 backdrop-filter: blur(12px);  
 overflow: hidden;  
 max-height: 90vh;  
 overflow-y: auto;  
}  
  
.close {  
 position: absolute;  
 top: 16px;  
 right: 20px;  
 font-size: 24px;  
 font-weight: 600;  
 cursor: pointer;  
 color: white;  
 transition: var(--transition);  
 z-index: 10;  
 background: rgba(0, 0, 0, 0.3);  
 border-radius: 50%;  
 width: 32px;  
 height: 32px;  
 display: flex;  
 align-items: center;  
 justify-content: center;  
}  
  
.close:hover {  
 background: rgba(0, 0, 0, 0.5);  
}  
  
.modal-header {  
 background: linear-gradient(135deg, var(--primary), var(--secondary));  
 padding: 40px 20px 30px;  
 text-align: center;  
 color: white;  
 position: relative;  
}  
  
.modal-avatar {  
 width: 100px;  
 height: 100px;  
 border-radius: 50%;  
 margin: 0 auto 20px;  
 border: 4px solid rgba(255, 255, 255, 0.3);  
 overflow: hidden;  
 background: rgba(255, 255, 255, 0.1);  
}  
  
.modal-avatar img {  
 width: 100%;  
 height: 100%;  
 object-fit: cover;  
}  
  
.modal-header h2 {  
 font-size: 24px;  
 font-weight: 700;  
 margin-bottom: 8px;  
 text-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);  
}  
  
.modal-subtitle {  
 font-size: 16px;  
 font-weight: 500;  
 opacity: 0.9;  
 margin: 0;  
}  
  
.modal-body {  
 padding: 30px;  
}  
  
.profile-info {  
 display: flex;  
 flex-direction: column;  
 gap: 20px;  
}  
  
.info-row {  
 display: flex;  
 justify-content: space-between;  
 align-items: center;  
 padding: 12px 0;  
 border-bottom: 1px solid var(--border-subtle);  
}  
  
.info-row:last-child {  
 border-bottom: none;  
}  
  
.info-label {  
 font-weight: 600;  
 color: var(--text-muted);  
 font-size: 14px;  
 min-width: 120px;  
}  
  
.info-value {  
 font-weight: 600;  
 color: var(--text-main);  
 font-size: 15px;  
 text-align: right;  
}  
  
/\* No Results Styling \*/  
.no-results {  
 grid-column: 1 / -1;  
 text-align: center;  
 padding: 40px 20px;  
 color: var(--text-muted);  
}  
  
.no-results p {  
 font-size: 16px;  
 font-weight: 500;  
 margin: 0;  
}  
  
@media (max-width: 768px) {  
 .search-section {  
 flex-direction: column;  
 gap: 12px;  
 }  
   
 .days-container {  
 grid-template-columns: 1fr;  
 }  
   
 .players-grid {  
 grid-template-columns: 1fr;  
 }  
   
 .container {  
 padding: 16px;  
 }  
   
 .player-card {  
 min-height: auto;  
 }  
   
 .player-image {  
 height: 120px;  
 }  
   
 .player-actions {  
 flex-direction: column;  
 gap: 8px;  
 }  
   
 .add-friend-btn,  
 .connect-btn {  
 width: 100%;  
 }  
}

————————————————————————————————————————

## FindPlayer\FindPlayer.html

<!DOCTYPE html>  
<html lang="en">  
<head>  
 <meta charset="UTF-8" />  
 <meta name="viewport" content="width=device-width, initial-scale=1.0"/>  
 <title>Find Player - Book&Play</title>  
 <link rel="stylesheet" href="FindPlayer.css" />  
</head>  
<body>  
 <div class="container">  
 <!-- 🔍 Search Section -->  
 <div class="search-section">  
 <div class="search-bar">  
 <input type="text" id="searchInput" placeholder="Search players by username..." />  
 <button class="search-btn">🔍</button>  
 </div>  
 <button class="sort-btn" id="sortByMeBtn">Sort By Me</button>  
 </div>  
  
 <!-- 📅 Availability Section -->  
 <div class="availability-section">  
 <div class="availability-header">  
 <h3>Your Availability</h3>  
 <div class="availability-controls">  
 <button class="select-all-btn" id="selectAllBtn">Select Everyday</button>  
 <div class="toggle-container">  
 <span>Available to Play</span>  
 <label class="toggle">  
 <input type="checkbox" id="availableToggle" checked />  
 <span class="slider"></span>  
 </label>  
 </div>  
 </div>  
 </div>  
  
 <div class="days-container">  
 <!-- Day Cards -->  
 <div class="day-card" data-day="sunday">  
 <div class="day-header">  
 <input type="checkbox" id="sun" />  
 <label for="sun">Sunday</label>  
 </div>  
 <div class="time-slots" id="slots-sunday"></div>  
 </div>  
 <div class="day-card" data-day="monday">  
 <div class="day-header">  
 <input type="checkbox" id="mon" />  
 <label for="mon">Monday</label>  
 </div>  
 <div class="time-slots" id="slots-monday"></div>  
 </div>  
 <div class="day-card" data-day="tuesday">  
 <div class="day-header">  
 <input type="checkbox" id="tue" />  
 <label for="tue">Tuesday</label>  
 </div>  
 <div class="time-slots" id="slots-tuesday"></div>  
 </div>  
 <div class="day-card" data-day="wednesday">  
 <div class="day-header">  
 <input type="checkbox" id="wed" />  
 <label for="wed">Wednesday</label>  
 </div>  
 <div class="time-slots" id="slots-wednesday"></div>  
 </div>  
 <div class="day-card" data-day="thursday">  
 <div class="day-header">  
 <input type="checkbox" id="thu" />  
 <label for="thu">Thursday</label>  
 </div>  
 <div class="time-slots" id="slots-thursday"></div>  
 </div>  
 <div class="day-card" data-day="friday">  
 <div class="day-header">  
 <input type="checkbox" id="fri" />  
 <label for="fri">Friday</label>  
 </div>  
 <div class="time-slots" id="slots-friday"></div>  
 </div>  
 <div class="day-card" data-day="saturday">  
 <div class="day-header">  
 <input type="checkbox" id="sat" />  
 <label for="sat">Saturday</label>  
 </div>  
 <div class="time-slots" id="slots-saturday"></div>  
 </div>  
 </div>  
 </div>  
  
 <!-- 👥 Players Display Section -->  
 <div class="players-section">  
 <h3>Suggested Players</h3>  
 <div class="players-grid" id="playersGrid">  
   
 <!-- Player Card 1 -->  
 <div class="player-card" onclick="openPlayerModal('Messi', 'messi@example.com', 'Soccer', '36', 'Male', '+972-50-123-4567', 'Tel Aviv', 'https://fcb-abj-pre.s3.amazonaws.com/img/jugadors/MESSI.jpg')">  
 <div class="player-image">  
 <img src="https://fcb-abj-pre.s3.amazonaws.com/img/jugadors/MESSI.jpg" alt="Messi" />  
 </div>  
 <div class="player-details">  
 <h4 class="player-name">Messi</h4>  
 <p class="player-sport">Sport: Soccer</p>  
 <p class="player-location">2.5 km away</p>  
 <div class="player-actions">  
 <button class="add-friend-btn" onclick="event.stopPropagation(); addFriend('Messi')">Add Friend</button>  
 <button class="connect-btn" onclick="event.stopPropagation()">Connect</button>  
 </div>  
 </div>  
 </div>  
  
 <!-- Player Card 2 -->  
 <div class="player-card" onclick="openPlayerModal('Sarah', 'sarah@example.com', 'Tennis', '28', 'Female', '+972-50-765-4321', 'Tel Aviv', 'https://thumbs.dreamstime.com/b/female-playing-tennis-waiting-ball-38898405.jpg')">  
 <div class="player-image">  
 <img src="https://thumbs.dreamstime.com/b/female-playing-tennis-waiting-ball-38898405.jpg" alt="Sarah" />  
 </div>  
 <div class="player-details">  
 <h4 class="player-name">Sarah</h4>  
 <p class="player-sport">Sport: Tennis</p>  
 <p class="player-location">1.8 km away</p>  
 <div class="player-actions">  
 <button class="add-friend-btn" onclick="event.stopPropagation(); addFriend('Sarah')">Add Friend</button>  
 <button class="connect-btn" onclick="event.stopPropagation()">Connect</button>  
 </div>  
 </div>  
 </div>  
  
 <!-- Player Card 3 -->  
 <div class="player-card" onclick="openPlayerModal('Michael', 'michael@example.com', 'Basketball', '24', 'Male', '+972-50-234-5678', 'Haifa', 'https://images.unsplash.com/photo-1546525848-3ce03ca516f6?ixlib=rb-4.0.3&auto=format&fit=crop&w=400&q=80')">  
 <div class="player-image">  
 <img src="https://images.unsplash.com/photo-1546525848-3ce03ca516f6?ixlib=rb-4.0.3&auto=format&fit=crop&w=400&q=80" alt="Michael" />  
 </div>  
 <div class="player-details">  
 <h4 class="player-name">Michael</h4>  
 <p class="player-sport">Sport: Basketball</p>  
 <p class="player-location">5.2 km away</p>  
 <div class="player-actions">  
 <button class="add-friend-btn" onclick="event.stopPropagation(); addFriend('Michael')">Add Friend</button>  
 <button class="connect-btn" onclick="event.stopPropagation()">Connect</button>  
 </div>  
 </div>  
 </div>  
  
 <!-- Player Card 4 -->  
 <div class="player-card" onclick="openPlayerModal('Emma', 'emma@example.com', 'Swimming', '22', 'Female', '+972-50-345-6789', 'Netanya', 'https://images.unsplash.com/photo-1530549387789-4c1017266635?ixlib=rb-4.0.3&auto=format&fit=crop&w=400&q=80')">  
 <div class="player-image">  
 <img src="https://images.unsplash.com/photo-1530549387789-4c1017266635?ixlib=rb-4.0.3&auto=format&fit=crop&w=400&q=80" alt="Emma" />  
 </div>  
 <div class="player-details">  
 <h4 class="player-name">Emma</h4>  
 <p class="player-sport">Sport: Swimming</p>  
 <p class="player-location">3.7 km away</p>  
 <div class="player-actions">  
 <button class="add-friend-btn" onclick="event.stopPropagation(); addFriend('Emma')">Add Friend</button>  
 <button class="connect-btn" onclick="event.stopPropagation()">Connect</button>  
 </div>  
 </div>  
 </div>  
  
 <!-- Player Card 5 -->  
 <div class="player-card" onclick="openPlayerModal('David', 'david@example.com', 'Running', '30', 'Male', '+972-50-456-7890', 'Jerusalem', 'https://images.unsplash.com/photo-1571019613454-1cb2f99b2d8b?ixlib=rb-4.0.3&auto=format&fit=crop&w=400&q=80')">  
 <div class="player-image">  
 <img src="https://images.unsplash.com/photo-1571019613454-1cb2f99b2d8b?ixlib=rb-4.0.3&auto=format&fit=crop&w=400&q=80" alt="David" />  
 </div>  
 <div class="player-details">  
 <h4 class="player-name">David</h4>  
 <p class="player-sport">Sport: Running</p>  
 <p class="player-location">8.1 km away</p>  
 <div class="player-actions">  
 <button class="add-friend-btn" onclick="event.stopPropagation(); addFriend('David')">Add Friend</button>  
 <button class="connect-btn" onclick="event.stopPropagation()">Connect</button>  
 </div>  
 </div>  
 </div>  
  
 <!-- Player Card 6 -->  
 <div class="player-card" onclick="openPlayerModal('Luna', 'luna@example.com', 'Volleyball', '26', 'Female', '+972-50-567-8901', 'Eilat', 'https://images.unsplash.com/photo-1594736797933-d0fce0ba4d2b?ixlib=rb-4.0.3&auto=format&fit=crop&w=400&q=80')">  
 <div class="player-image">  
 <img src="https://images.unsplash.com/photo-1594736797933-d0fce0ba4d2b?ixlib=rb-4.0.3&auto=format&fit=crop&w=400&q=80" alt="Luna" />  
 </div>  
 <div class="player-details">  
 <h4 class="player-name">Luna</h4>  
 <p class="player-sport">Sport: Volleyball</p>  
 <p class="player-location">4.3 km away</p>  
 <div class="player-actions">  
 <button class="add-friend-btn" onclick="event.stopPropagation(); addFriend('Luna')">Add Friend</button>  
 <button class="connect-btn" onclick="event.stopPropagation()">Connect</button>  
 </div>  
 </div>  
 </div>  
  
 <!-- Player Card 7 -->  
 <div class="player-card" onclick="openPlayerModal('James', 'james@example.com', 'Cycling', '29', 'Male', '+972-50-678-9012', 'Be er Sheva', 'https://images.unsplash.com/photo-1558618666-fcd25c85cd64?ixlib=rb-4.0.3&auto=format&fit=crop&w=400&q=80')">  
 <div class="player-image">  
 <img src="https://images.unsplash.com/photo-1558618666-fcd25c85cd64?ixlib=rb-4.0.3&auto=format&fit=crop&w=400&q=80" alt="James" />  
 </div>  
 <div class="player-details">  
 <h4 class="player-name">James</h4>  
 <p class="player-sport">Sport: Cycling</p>  
 <p class="player-location">6.8 km away</p>  
 <div class="player-actions">  
 <button class="add-friend-btn" onclick="event.stopPropagation(); addFriend('James')">Add Friend</button>  
 <button class="connect-btn" onclick="event.stopPropagation()">Connect</button>  
 </div>  
 </div>  
 </div>  
  
 <!-- Player Card 8 -->  
 <div class="player-card" onclick="openPlayerModal('Sophia', 'sophia@example.com', 'Yoga', '25', 'Female', '+972-50-789-0123', 'Herzliya', 'https://images.unsplash.com/photo-1544367567-0f2fcb009e0b?ixlib=rb-4.0.3&auto=format&fit=crop&w=400&q=80')">  
 <div class="player-image">  
 <img src="https://images.unsplash.com/photo-1544367567-0f2fcb009e0b?ixlib=rb-4.0.3&auto=format&fit=crop&w=400&q=80" alt="Sophia" />  
 </div>  
 <div class="player-details">  
 <h4 class="player-name">Sophia</h4>  
 <p class="player-sport">Sport: Yoga</p>  
 <p class="player-location">2.1 km away</p>  
 <div class="player-actions">  
 <button class="add-friend-btn" onclick="event.stopPropagation(); addFriend('Sophia')">Add Friend</button>  
 <button class="connect-btn" onclick="event.stopPropagation()">Connect</button>  
 </div>  
 </div>  
 </div>  
  
 <!-- Player Card 9 -->  
 <div class="player-card" onclick="openPlayerModal('Alex', 'alex@example.com', 'Boxing', '27', 'Male', '+972-50-890-1234', 'Ashdod', 'https://images.unsplash.com/photo-1549476464-37392f717541?ixlib=rb-4.0.3&auto=format&fit=crop&w=400&q=80')">  
 <div class="player-image">  
 <img src="https://images.unsplash.com/photo-1549476464-37392f717541?ixlib=rb-4.0.3&auto=format&fit=crop&w=400&q=80" alt="Alex" />  
 </div>  
 <div class="player-details">  
 <h4 class="player-name">Alex</h4>  
 <p class="player-sport">Sport: Boxing</p>  
 <p class="player-location">7.5 km away</p>  
 <div class="player-actions">  
 <button class="add-friend-btn" onclick="event.stopPropagation(); addFriend('Alex')">Add Friend</button>  
 <button class="connect-btn" onclick="event.stopPropagation()">Connect</button>  
 </div>  
 </div>  
 </div>  
  
 <!-- Player Card 10 -->  
 <div class="player-card" onclick="openPlayerModal('Maya', 'maya@example.com', 'Pilates', '31', 'Female', '+972-50-901-2345', 'Rishon LeZion', 'https://images.unsplash.com/photo-1518611012118-696072aa579a?ixlib=rb-4.0.3&auto=format&fit=crop&w=400&q=80')">  
 <div class="player-image">  
 <img src="https://images.unsplash.com/photo-1518611012118-696072aa579a?ixlib=rb-4.0.3&auto=format&fit=crop&w=400&q=80" alt="Maya" />  
 </div>  
 <div class="player-details">  
 <h4 class="player-name">Maya</h4>  
 <p class="player-sport">Sport: Pilates</p>  
 <p class="player-location">3.2 km away</p>  
 <div class="player-actions">  
 <button class="add-friend-btn" onclick="event.stopPropagation(); addFriend('Maya')">Add Friend</button>  
 <button class="connect-btn" onclick="event.stopPropagation()">Connect</button>  
 </div>  
 </div>  
 </div>  
  
 <!-- Player Card 11 -->  
 <div class="player-card" onclick="openPlayerModal('Oliver', 'oliver@example.com', 'Golf', '35', 'Male', '+972-50-012-3456', 'Caesarea', 'https://images.unsplash.com/photo-1579952363873-27d3bfad9c0d?ixlib=rb-4.0.3&auto=format&fit=crop&w=400&q=80')">  
 <div class="player-image">  
 <img src="https://images.unsplash.com/photo-1579952363873-27d3bfad9c0d?ixlib=rb-4.0.3&auto=format&fit=crop&w=400&q=80" alt="Oliver" />  
 </div>  
 <div class="player-details">  
 <h4 class="player-name">Oliver</h4>  
 <p class="player-sport">Sport: Golf</p>  
 <p class="player-location">9.4 km away</p>  
 <div class="player-actions">  
 <button class="add-friend-btn" onclick="event.stopPropagation(); addFriend('Oliver')">Add Friend</button>  
 <button class="connect-btn" onclick="event.stopPropagation()">Connect</button>  
 </div>  
 </div>  
 </div>  
  
 <!-- Player Card 12 -->  
 <div class="player-card" onclick="openPlayerModal('Zoe', 'zoe@example.com', 'Rock Climbing', '23', 'Female', '+972-50-123-4567', 'Tiberias', 'https://images.unsplash.com/photo-1522163723043-478ef79a5bb4?ixlib=rb-4.0.3&auto=format&fit=crop&w=400&q=80')">  
 <div class="player-image">  
 <img src="https://images.unsplash.com/photo-1522163723043-478ef79a5bb4?ixlib=rb-4.0.3&auto=format&fit=crop&w=400&q=80" alt="Zoe" />  
 </div>  
 <div class="player-details">  
 <h4 class="player-name">Zoe</h4>  
 <p class="player-sport">Sport: Rock Climbing</p>  
 <p class="player-location">12.1 km away</p>  
 <div class="player-actions">  
 <button class="add-friend-btn" onclick="event.stopPropagation(); addFriend('Zoe')">Add Friend</button>  
 <button class="connect-btn" onclick="event.stopPropagation()">Connect</button>  
 </div>  
 </div>  
 </div>  
  
 <!-- Player Card 13 -->  
 <div class="player-card" onclick="openPlayerModal('Ryan', 'ryan@example.com', 'Surfing', '26', 'Male', '+972-50-234-5678', 'Ashkelon', 'https://images.unsplash.com/photo-1502680390469-be75c86b636f?ixlib=rb-4.0.3&auto=format&fit=crop&w=400&q=80')">  
 <div class="player-image">  
 <img src="https://images.unsplash.com/photo-1502680390469-be75c86b636f?ixlib=rb-4.0.3&auto=format&fit=crop&w=400&q=80" alt="Ryan" />  
 </div>  
 <div class="player-details">  
 <h4 class="player-name">Ryan</h4>  
 <p class="player-sport">Sport: Surfing</p>  
 <p class="player-location">15.7 km away</p>  
 <div class="player-actions">  
 <button class="add-friend-btn" onclick="event.stopPropagation(); addFriend('Ryan')">Add Friend</button>  
 <button class="connect-btn" onclick="event.stopPropagation()">Connect</button>  
 </div>  
 </div>  
 </div>  
  
 <!-- Player Card 14 -->  
 <div class="player-card" onclick="openPlayerModal('Aria', 'aria@example.com', 'Badminton', '24', 'Female', '+972-50-345-6789', 'Kfar Saba', 'https://images.unsplash.com/photo-1594736797933-d0fce0ba4d2b?ixlib=rb-4.0.3&auto=format&fit=crop&w=400&q=80')">  
 <div class="player-image">  
 <img src="https://images.unsplash.com/photo-1594736797933-d0fce0ba4d2b?ixlib=rb-4.0.3&auto=format&fit=crop&w=400&q=80" alt="Aria" />  
 </div>  
 <div class="player-details">  
 <h4 class="player-name">Aria</h4>  
 <p class="player-sport">Sport: Badminton</p>  
 <p class="player-location">4.9 km away</p>  
 <div class="player-actions">  
 <button class="add-friend-btn" onclick="event.stopPropagation(); addFriend('Aria')">Add Friend</button>  
 <button class="connect-btn" onclick="event.stopPropagation()">Connect</button>  
 </div>  
 </div>  
 </div>  
  
 <!-- Player Card 15 -->  
 <div class="player-card" onclick="openPlayerModal('Ethan', 'ethan@example.com', 'Martial Arts', '32', 'Male', '+972-50-456-7890', 'Ramat Gan', 'https://images.unsplash.com/photo-1555597408-51bc5d65bedf?ixlib=rb-4.0.3&auto=format&fit=crop&w=400&q=80')">  
 <div class="player-image">  
 <img src="https://images.unsplash.com/photo-1555597408-51bc5d65bedf?ixlib=rb-4.0.3&auto=format&fit=crop&w=400&q=80" alt="Ethan" />  
 </div>  
 <div class="player-details">  
 <h4 class="player-name">Ethan</h4>  
 <p class="player-sport">Sport: Martial Arts</p>  
 <p class="player-location">6.3 km away</p>  
 <div class="player-actions">  
 <button class="add-friend-btn" onclick="event.stopPropagation(); addFriend('Ethan')">Add Friend</button>  
 <button class="connect-btn" onclick="event.stopPropagation()">Connect</button>  
 </div>  
 </div>  
 </div>  
  
 </div>  
 </div>  
 </div>  
  
 <!-- 📋 Player Modal -->  
 <div class="modal" id="playerModal">  
 <div class="modal-content">  
 <span class="close" onclick="closePlayerModal()">&times;</span>  
   
 <!-- Modal Header with Gradient Background -->  
 <div class="modal-header">  
 <div class="modal-avatar">  
 <img id="modalPlayerImage" src="" alt="Player" />  
 </div>  
 <h2 id="modalPlayerName">Player Name</h2>  
 <p class="modal-subtitle">Profile Information</p>  
 </div>  
   
 <!-- Modal Body with Player Details -->  
 <div class="modal-body">  
 <div class="profile-info">  
 <div class="info-row">  
 <span class="info-label">Username:</span>  
 <span class="info-value" id="modalUsername">-</span>  
 </div>  
 <div class="info-row">  
 <span class="info-label">Email:</span>  
 <span class="info-value" id="modalEmail">-</span>  
 </div>  
 <div class="info-row">  
 <span class="info-label">Favorite Sport:</span>  
 <span class="info-value" id="modalSport">-</span>  
 </div>  
 <div class="info-row">  
 <span class="info-label">Age:</span>  
 <span class="info-value" id="modalAge">-</span>  
 </div>  
 <div class="info-row">  
 <span class="info-label">Gender:</span>  
 <span class="info-value" id="modalGender">-</span>  
 </div>  
 <div class="info-row">  
 <span class="info-label">Phone Number:</span>  
 <span class="info-value" id="modalPhone">-</span>  
 </div>  
 <div class="info-row">  
 <span class="info-label">Location:</span>  
 <span class="info-value" id="modalLocation">-</span>  
 </div>  
 </div>  
 </div>  
 </div>  
 </div>  
  
 <script src="FindPlayer.js" defer></script>  
</body>  
</html>

————————————————————————————————————————

## FindPlayer\FindPlayer.js

// Cleaned version: data removed for DB integration  
let currentPlayers = [];  
let userAvailability = {};  
let isAvailableToPlay = true;  
let searchTimeout;  
let friendsList = []; // Store friends list  
  
// Sample players data (replace with your database data)  
const allPlayers = [  
 { name: 'Messi', email: 'messi@example.com', sport: 'Soccer', age: '36', gender: 'Male', phone: '+972-50-123-4567', location: 'Tel Aviv', distance: '2.5 km away', image: 'https://fcb-abj-pre.s3.amazonaws.com/img/jugadors/MESSI.jpg' },  
 { name: 'Sarah', email: 'sarah@example.com', sport: 'Tennis', age: '28', gender: 'Female', phone: '+972-50-765-4321', location: 'Tel Aviv', distance: '1.8 km away', image: 'https://thumbs.dreamstime.com/b/female-playing-tennis-waiting-ball-38898405.jpg' },  
 { name: 'Michael', email: 'michael@example.com', sport: 'Basketball', age: '24', gender: 'Male', phone: '+972-50-234-5678', location: 'Haifa', distance: '5.2 km away', image: 'https://images.unsplash.com/photo-1546525848-3ce03ca516f6?ixlib=rb-4.0.3&auto=format&fit=crop&w=400&q=80' },  
 { name: 'Emma', email: 'emma@example.com', sport: 'Swimming', age: '22', gender: 'Female', phone: '+972-50-345-6789', location: 'Netanya', distance: '3.7 km away', image: 'https://images.unsplash.com/photo-1530549387789-4c1017266635?ixlib=rb-4.0.3&auto=format&fit=crop&w=400&q=80' },  
 { name: 'David', email: 'david@example.com', sport: 'Running', age: '30', gender: 'Male', phone: '+972-50-456-7890', location: 'Jerusalem', distance: '8.1 km away', image: 'https://images.unsplash.com/photo-1571019613454-1cb2f99b2d8b?ixlib=rb-4.0.3&auto=format&fit=crop&w=400&q=80' },  
 { name: 'Luna', email: 'luna@example.com', sport: 'Volleyball', age: '26', gender: 'Female', phone: '+972-50-567-8901', location: 'Eilat', distance: '4.3 km away', image: 'https://images.unsplash.com/photo-1594736797933-d0fce0ba4d2b?ixlib=rb-4.0.3&auto=format&fit=crop&w=400&q=80' },  
 { name: 'James', email: 'james@example.com', sport: 'Cycling', age: '29', gender: 'Male', phone: '+972-50-678-9012', location: 'Be er Sheva', distance: '6.8 km away', image: 'https://images.unsplash.com/photo-1558618666-fcd25c85cd64?ixlib=rb-4.0.3&auto=format&fit=crop&w=400&q=80' },  
 { name: 'Sophia', email: 'sophia@example.com', sport: 'Yoga', age: '25', gender: 'Female', phone: '+972-50-789-0123', location: 'Herzliya', distance: '2.1 km away', image: 'https://images.unsplash.com/photo-1544367567-0f2fcb009e0b?ixlib=rb-4.0.3&auto=format&fit=crop&w=400&q=80' },  
 { name: 'Alex', email: 'alex@example.com', sport: 'Boxing', age: '27', gender: 'Male', phone: '+972-50-890-1234', location: 'Ashdod', distance: '7.5 km away', image: 'https://images.unsplash.com/photo-1549476464-37392f717541?ixlib=rb-4.0.3&auto=format&fit=crop&w=400&q=80' },  
 { name: 'Maya', email: 'maya@example.com', sport: 'Pilates', age: '31', gender: 'Female', phone: '+972-50-901-2345', location: 'Rishon LeZion', distance: '3.2 km away', image: 'https://images.unsplash.com/photo-1518611012118-696072aa579a?ixlib=rb-4.0.3&auto=format&fit=crop&w=400&q=80' },  
 { name: 'Oliver', email: 'oliver@example.com', sport: 'Golf', age: '35', gender: 'Male', phone: '+972-50-012-3456', location: 'Caesarea', distance: '9.4 km away', image: 'https://images.unsplash.com/photo-1579952363873-27d3bfad9c0d?ixlib=rb-4.0.3&auto=format&fit=crop&w=400&q=80' },  
 { name: 'Zoe', email: 'zoe@example.com', sport: 'Rock Climbing', age: '23', gender: 'Female', phone: '+972-50-123-4567', location: 'Tiberias', distance: '12.1 km away', image: 'https://images.unsplash.com/photo-1522163723043-478ef79a5bb4?ixlib=rb-4.0.3&auto=format&fit=crop&w=400&q=80' },  
 { name: 'Ryan', email: 'ryan@example.com', sport: 'Surfing', age: '26', gender: 'Male', phone: '+972-50-234-5678', location: 'Ashkelon', distance: '15.7 km away', image: 'https://images.unsplash.com/photo-1502680390469-be75c86b636f?ixlib=rb-4.0.3&auto=format&fit=crop&w=400&q=80' },  
 { name: 'Aria', email: 'aria@example.com', sport: 'Badminton', age: '24', gender: 'Female', phone: '+972-50-345-6789', location: 'Kfar Saba', distance: '4.9 km away', image: 'https://images.unsplash.com/photo-1594736797933-d0fce0ba4d2b?ixlib=rb-4.0.3&auto=format&fit=crop&w=400&q=80' },  
 { name: 'Ethan', email: 'ethan@example.com', sport: 'Martial Arts', age: '32', gender: 'Male', phone: '+972-50-456-7890', location: 'Ramat Gan', distance: '6.3 km away', image: 'https://images.unsplash.com/photo-1555597408-51bc5d65bedf?ixlib=rb-4.0.3&auto=format&fit=crop&w=400&q=80' }  
];  
  
const searchInput = document.getElementById('searchInput');  
const sortByMeBtn = document.getElementById('sortByMeBtn');  
const availableToggle = document.getElementById('availableToggle');  
const selectAllBtn = document.getElementById('selectAllBtn');  
const playersGrid = document.getElementById('playersGrid');  
const playerModal = document.getElementById('playerModal');  
const modalBody = document.getElementById('modalBody');  
const closeModal = document.querySelector('.close');  
  
const days = ['sunday','monday','tuesday','wednesday','thursday','friday','saturday'];  
  
document.addEventListener('DOMContentLoaded', () => {  
 initializeAvailability();  
 setupEventListeners();  
 currentPlayers = [...allPlayers]; // Initialize with all players  
 renderPlayers(); // Render players on page load  
});  
  
function initializeAvailability() {  
 days.forEach(day => {  
 const checkbox = document.getElementById(day.substring(0,3));  
 const container = document.getElementById(`slots-${day}`);  
 if (checkbox) {  
 checkbox.addEventListener('change', () => {  
 const dayCard = document.querySelector(`[data-day="${day}"]`);  
 if (checkbox.checked) {  
 dayCard.classList.add('active');  
 showAddButton(day);  
 } else {  
 dayCard.classList.remove('active');  
 container.innerHTML = '';  
 delete userAvailability[day];  
 }  
 });  
 }  
 });  
}  
  
function showAddButton(day) {  
 const container = document.getElementById(`slots-${day}`);  
 if (!container.querySelector('.add-slot-btn')) {  
 const btn = document.createElement('button');  
 btn.className = 'add-slot-btn';  
 btn.textContent = '+ Add Time Slot';  
 btn.onclick = () => addTimeSlot(day);  
 container.appendChild(btn);  
 }  
}  
  
function addTimeSlot(day) {  
 const container = document.getElementById(`slots-${day}`);  
 if (container.querySelectorAll('.time-slot').length >= 3) return;  
 const div = document.createElement('div');  
 div.className = 'time-slot';  
 div.innerHTML = '<input type="time" class="start-time"><span>–</span><input type="time" class="end-time"><button class="delete-slot" onclick="deleteTimeSlot(this)">×</button>';  
 const addBtn = container.querySelector('.add-slot-btn');  
 if (addBtn) addBtn.remove();  
 container.appendChild(div);  
 const inputs = div.querySelectorAll('input');  
 inputs.forEach(input => input.addEventListener('change', () => updateUserAvailability(day)));  
 if (container.querySelectorAll('.time-slot').length < 3) showAddButton(day);  
 updateUserAvailability(day);  
}  
  
function deleteTimeSlot(btn) {  
 const container = btn.parentElement.parentElement;  
 const day = container.id.replace('slots-','');  
 btn.parentElement.remove();  
 if (container.querySelectorAll('.time-slot').length < 3) showAddButton(day);  
 updateUserAvailability(day);  
}  
  
function updateUserAvailability(day) {  
 const container = document.getElementById(`slots-${day}`);  
 const slots = container.querySelectorAll('.time-slot');  
 if (!slots.length) return delete userAvailability[day];  
 userAvailability[day] = [];  
 slots.forEach(slot => {  
 const start = slot.querySelector('.start-time').value;  
 const end = slot.querySelector('.end-time').value;  
 if (start && end) userAvailability[day].push(`${start}-${end}`);  
 });  
}  
  
// Search functionality  
function searchPlayers(searchTerm) {  
 if (!searchTerm.trim()) {  
 // If search is empty, show all players  
 currentPlayers = [...allPlayers];  
 } else {  
 // Filter players by username (case-insensitive)  
 currentPlayers = allPlayers.filter(player =>   
 player.name.toLowerCase().startsWith(searchTerm.toLowerCase())  
 );  
 }  
 renderPlayers();  
}  
  
function renderPlayers() {  
 if (!playersGrid) return;  
   
 if (currentPlayers.length === 0) {  
 playersGrid.innerHTML = '<div class="no-results"><p>No players found matching your search.</p></div>';  
 return;  
 }  
   
 playersGrid.innerHTML = currentPlayers.map(player => {  
 const isFriend = friendsList.includes(player.name);  
 return `  
 <div class="player-card" onclick="openPlayerModal('${player.name}', '${player.email}', '${player.sport}', '${player.age}', '${player.gender}', '${player.phone}', '${player.location}', '${player.image}')">  
 <div class="player-image">  
 <img src="${player.image}" alt="${player.name}" />  
 </div>  
 <div class="player-details">  
 <h4 class="player-name">${player.name}</h4>  
 <p class="player-sport">Sport: ${player.sport}</p>  
 <p class="player-location">${player.distance}</p>  
 <div class="player-actions">  
 <button class="add-friend-btn ${isFriend ? 'added' : ''}"   
 onclick="event.stopPropagation(); addFriend('${player.name}')"   
 ${isFriend ? 'disabled' : ''}>  
 ${isFriend ? 'Friend Added' : 'Add Friend'}  
 </button>  
 <button class="connect-btn" onclick="event.stopPropagation()">Connect</button>  
 </div>  
 </div>  
 </div>  
 `;}).join('');  
}  
  
// Add Friend functionality  
function addFriend(playerName) {  
 // Check if already friends  
 if (friendsList.includes(playerName)) {  
 alert(`${playerName} is already in your friends list!`);  
 return;  
 }  
   
 // Add to friends list  
 friendsList.push(playerName);  
   
 // Show success message  
 alert(`${playerName} has been added to your friends list!`);  
   
 // Re-render players to update button state  
 renderPlayers();  
   
 console.log('Current friends list:', friendsList);  
}  
  
function setupEventListeners() {  
 if (searchInput) {  
 searchInput.addEventListener('input', (e) => {  
 clearTimeout(searchTimeout);  
 // Add small delay for better performance  
 searchTimeout = setTimeout(() => {  
 searchPlayers(e.target.value);  
 }, 300);  
 });  
 }  
   
 if (sortByMeBtn) sortByMeBtn.addEventListener('click', () => {});  
 if (selectAllBtn) selectAllBtn.addEventListener('click', handleSelectAll);  
 if (availableToggle) availableToggle.addEventListener('change', () => {  
 isAvailableToPlay = availableToggle.checked;  
 });  
 if (closeModal) closeModal.addEventListener('click', () => playerModal.style.display = 'none');  
}  
  
function handleSelectAll() {  
 const allChecked = days.every(day => {  
 const checkbox = document.getElementById(day.substring(0,3));  
 return checkbox && checkbox.checked;  
 });  
   
 days.forEach(day => {  
 const checkbox = document.getElementById(day.substring(0,3));  
 const container = document.getElementById(`slots-${day}`);  
 const card = document.querySelector(`[data-day="${day}"]`);  
   
 if (checkbox && container && card) {  
 if (!allChecked) {  
 checkbox.checked = true;  
 card.classList.add('active');  
 showAddButton(day);  
 } else {  
 checkbox.checked = false;  
 card.classList.remove('active');  
 container.innerHTML = '';  
 delete userAvailability[day];  
 }  
 }  
 });  
   
 if (selectAllBtn) {  
 selectAllBtn.textContent = allChecked ? 'Select Everyday' : 'Clear All';  
 }  
}  
  
// Modal Functions for Player Profile  
function openPlayerModal(username, email, sport, age, gender, phone, location, image) {  
 const modalElements = {  
 name: document.getElementById('modalPlayerName'),  
 username: document.getElementById('modalUsername'),  
 email: document.getElementById('modalEmail'),  
 sport: document.getElementById('modalSport'),  
 age: document.getElementById('modalAge'),  
 gender: document.getElementById('modalGender'),  
 phone: document.getElementById('modalPhone'),  
 location: document.getElementById('modalLocation'),  
 image: document.getElementById('modalPlayerImage')  
 };  
   
 // Update modal content if elements exist  
 if (modalElements.name) modalElements.name.textContent = username;  
 if (modalElements.username) modalElements.username.textContent = username;  
 if (modalElements.email) modalElements.email.textContent = email;  
 if (modalElements.sport) modalElements.sport.textContent = sport;  
 if (modalElements.age) modalElements.age.textContent = age;  
 if (modalElements.gender) modalElements.gender.textContent = gender;  
 if (modalElements.phone) modalElements.phone.textContent = phone;  
 if (modalElements.location) modalElements.location.textContent = location;  
 if (modalElements.image) modalElements.image.src = image;  
   
 if (playerModal) {  
 playerModal.style.display = 'block';  
 }  
}  
  
function closePlayerModal() {  
 if (playerModal) {  
 playerModal.style.display = 'none';  
 }  
}  
  
// Close modal when clicking outside of it  
window.onclick = function(event) {  
 if (playerModal && event.target == playerModal) {  
 closePlayerModal();  
 }  
}

————————————————————————————————————————

## FindPlayer\FindPlayer.php

<?php  
require\_once '../../../db.php';  
include '../navbar.php';  
include '../sportsScroll.html';  
include 'FindPlayer.html';  
  
  
  
?>

————————————————————————————————————————

## HomePage\HomePage.css

/\* Light Elegant Theme - Responsive Version \*/  
  
:root {  
 --primary: #1e90ff;  
 --secondary: #22c55e;  
 --light-bg: #f8f6f0;  
 --card-bg: #ffffff;  
 --card-border: #d1d5db;  
 --text-main: #1e293b;  
 --muted-text: #6b7280;  
 --accent-glow: rgba(30, 144, 255, 0.3);  
 --radius: 0.75rem;  
}  
  
\* {  
 box-sizing: border-box;  
 margin: 0;  
 padding: 0;  
}  
  
body {  
 margin: 0;  
 font-family: 'Outfit', sans-serif;  
 background: linear-gradient(135deg, #e0f7fa, #d0f8ce);  
 color: var(--text-main);  
 min-height: 100vh;  
 display: flex;  
 flex-direction: column;  
 align-items: center;  
 overflow-x: hidden;  
 padding: 0;  
}  
  
.header-bar {  
 width: 100%;  
 padding: 20px 5%;  
 display: flex;  
 justify-content: space-between;  
 align-items: center;  
 flex-wrap: wrap;  
 gap: 15px;  
}  
  
.logo-title {  
 font-size: 32px;  
 font-weight: bold;  
 background: linear-gradient(to right, #1e90ff, #22c55e);  
 -webkit-background-clip: text;  
 background-clip: text;  
 -webkit-text-fill-color: transparent;  
 white-space: nowrap;  
}  
  
.logo-title span {  
 color: #f472b6;  
}  
  
.profile-container {  
 position: relative;  
 cursor: pointer;  
 z-index: 100;  
}  
  
.profile-icon {  
 width: 60px;  
 height: 60px;  
 border-radius: 50%;  
 object-fit: cover;  
 box-shadow: 0 0 6px var(--accent-glow);  
}  
  
.profile-menu {  
 position: fixed; /\* instead of absolute \*/  
 top: 70px;  
 right: 30px; /\* you can adjust the number according to spacing \*/  
 background-color: white;  
 border: 1px solid var(--card-border);  
 border-radius: var(--radius);  
 box-shadow: 0 4px 16px rgba(0, 0, 0, 0.06);  
 overflow: hidden;  
 display: none;  
 flex-direction: column;  
 min-width: 180px;  
 z-index: 101;  
 transition: opacity 0.3s ease, transform 0.3s ease;  
}  
  
  
.profile-menu a {  
 padding: 12px 18px;  
 text-decoration: none;  
 color: var(--text-main);  
 font-size: 14px;  
}  
  
.profile-menu a:hover {  
 background-color: #f3f4f6;  
}  
  
.profile-menu.active {  
 display: flex;  
}  
  
.main-buttons {  
 display: flex;  
 gap: 30px;  
 margin-top: 60px;  
 flex-wrap: wrap;  
 justify-content: center;  
 padding: 0 20px;  
}  
  
.secondary-buttons {  
 display: grid;  
 grid-template-columns: repeat(auto-fit, minmax(140px, 1fr));  
 gap: 30px;  
 width: 90%;  
 max-width: 900px;  
 margin: 60px auto;  
 padding: 0 10px;  
}  
  
.glow-card {  
 background: var(--card-bg);  
 border: 1px solid var(--card-border);  
 border-radius: var(--radius);  
 padding: 20px;  
 text-align: center;  
 color: var(--text-main);  
 box-shadow: 0 4px 12px rgba(30, 144, 255, 0.05);  
 transition: all 0.3s ease;  
 cursor: pointer;  
 width: 160px;  
 height: 160px;  
 display: flex;  
 flex-direction: column;  
 align-items: center;  
 justify-content: center;  
}  
  
.main-buttons .glow-card {  
 width: 200px;  
 height: 200px;  
}  
  
.glow-card img.icon {  
 width: 80px;  
 margin-bottom: 12px;  
}  
  
.glow-card span {  
 display: block;  
 font-size: 15px;  
 font-weight: 500;  
 color: var(--text-main);  
}  
  
.glow-card:hover {  
 transform: translateY(-5px);  
 box-shadow: 0 6px 20px rgba(30, 144, 255, 0.15);  
}  
  
.btn-find { box-shadow: 0 0 12px #38bdf8; }  
.btn-group { box-shadow: 0 0 12px gold; }  
.btn-book { box-shadow: 0 0 12px #22c55e; }  
.btn-fav { box-shadow: 0 0 12px #f43f5e; }  
.btn-chat { box-shadow: 0 0 12px #3b82f6; }  
.btn-bookings { box-shadow: 0 0 12px #fb923c; }  
.btn-switch { box-shadow: 0 0 12px #a855f7; }  
  
footer {  
 background-color: transparent;  
 padding: 20px;  
 width: 100%;  
 text-align: center;  
 font-size: 14px;  
 color: var(--muted-text);  
 margin-top: auto;  
}  
  
.footer-content {  
 max-width: 1200px;  
 margin: auto;  
 padding: 0 20px;  
}  
.floating-admin-btn {  
 position: fixed;  
 bottom: 20px;  
 right: 20px;  
 background: linear-gradient(135deg, #facc15, #fef08a);  
 color: #1e293b;  
 border: none;  
 border-radius: 12px;  
 padding: 12px 20px;  
 font-size: 14px;  
 font-weight: 600;  
 box-shadow: 0 4px 12px rgba(250, 204, 21, 0.4);  
 cursor: pointer;  
 transition: all 0.3s ease;  
 z-index: 999;  
}  
  
.floating-admin-btn:hover {  
 background: linear-gradient(135deg, #eab308, #fde047);  
 box-shadow: 0 6px 16px rgba(250, 204, 21, 0.6);  
}  
  
@keyframes fadeInUp {  
 from { opacity: 0; transform: translateY(20px); }  
 to { opacity: 1; transform: translateY(0); }  
}  
  
/\* Responsive Media Queries \*/  
  
/\* Large devices (laptops/desktops, up to 1200px) \*/  
@media screen and (max-width: 1200px) {  
 .secondary-buttons {  
 grid-template-columns: repeat(auto-fit, minmax(130px, 1fr));  
 gap: 25px;  
 }  
}  
  
/\* Medium devices (tablets, up to 992px) \*/  
@media screen and (max-width: 992px) {  
 .main-buttons {  
 gap: 20px;  
 }  
   
 .main-buttons .glow-card {  
 width: 180px;  
 height: 180px;  
 }  
   
 .glow-card img.icon {  
 width: 70px;  
 }  
   
 .secondary-buttons {  
 gap: 20px;  
 }  
}  
  
/\* Small devices (landscape phones, up to 768px) \*/  
@media screen and (max-width: 768px) {  
 .header-bar {  
 padding: 15px 20px;  
 }  
   
 .logo-title {  
 font-size: 28px;  
 }  
   
 .main-buttons {  
 margin-top: 40px;  
 gap: 15px;  
 }  
   
 .main-buttons .glow-card {  
 width: 160px;  
 height: 160px;  
 }  
   
 .secondary-buttons {  
 grid-template-columns: repeat(auto-fit, minmax(120px, 1fr));  
 margin: 40px auto;  
 }  
   
 .glow-card {  
 width: 140px;  
 height: 140px;  
 padding: 15px;  
 }  
   
 .glow-card img.icon {  
 width: 60px;  
 margin-bottom: 8px;  
 }  
   
 .glow-card span {  
 font-size: 14px;  
 }  
}  
  
/\* Extra small devices (phones, up to 576px) \*/  
@media screen and (max-width: 576px) {  
 .header-bar {  
 padding: 15px;  
 justify-content: center;  
 }  
   
 .logo-title {  
 font-size: 24px;  
 text-align: center;  
 }  
   
 .profile-container {  
 position: absolute;  
 top: 15px;  
 right: 15px;  
 }  
   
 .main-buttons {  
 margin-top: 30px;  
 flex-direction: column;  
 align-items: center;  
 gap: 15px;  
 }  
   
 .main-buttons .glow-card {  
 width: 180px;  
 height: 180px;  
 }  
   
 .secondary-buttons {  
 grid-template-columns: repeat(2, 1fr);  
 gap: 15px;  
 margin: 30px auto;  
 }  
   
 .glow-card {  
 width: 100%;  
 height: 130px;  
 padding: 10px;  
 }  
   
 .glow-card img.icon {  
 width: 50px;  
 margin-bottom: 5px;  
 }  
   
 .glow-card span {  
 font-size: 13px;  
 }  
   
 footer {  
 padding: 15px;  
 font-size: 12px;  
 }  
}  
  
/\* Very small devices (small phones, up to 380px) \*/  
@media screen and (max-width: 380px) {  
 .logo-title {  
 font-size: 22px;  
 }  
   
 .main-buttons .glow-card {  
 width: 160px;  
 height: 160px;  
 }  
   
 .secondary-buttons {  
 grid-template-columns: 1fr;  
 width: 85%;  
 }  
   
 .glow-card {  
 height: 110px;  
 }  
   
 .profile-menu {  
 min-width: 160px;  
 right: -10px;  
 }  
}  
.username-display {  
 margin-left: 20px;  
 font-weight: 600;  
 color: var(--text-main);  
 font-size: 15px;  
}

————————————————————————————————————————

## HomePage\HomePage.html

<!DOCTYPE html>  
<html lang="en">  
<head>  
 <meta charset="UTF-8">  
 <meta name="viewport" content="width=device-width, initial-scale=1.0">  
 <title>Book&Play-Player</title>  
  
 <link rel="stylesheet" href="HomePage.css">  
  
 <link href="https://fonts.googleapis.com/css2?family=Outfit:wght@500;600&display=swap" rel="stylesheet">  
</head>  
<body>  
  
 <div class="bg-shapes"></div>  
  
 <div class="header-bar">  
 <div class="logo-title">BOOK&PLAY</div>  
 <div class="profile-container" onclick="toggleProfileMenu()">  
 <span id="usernameDisplay" class="username-display"></span>  
 <img  
 id="userProfileImage"  
 class="profile-icon"  
 src="../../../uploads/users/default.jpg"  
 alt="Profile Image"  
 />  
 <div class="profile-menu" id="profileMenu">  
 <a href="../../auth/EditProfile/EditProfile.php">Edit Profile</a>  
 <a href="../../../logout.php">Logout</a>  
 </div>  
 </div>  
   
 </div>  
  
 <div class="main-buttons">  
 <div class="glow-card btn-find" onclick="window.location.href='../FindPlayer/FindPlayer.php'">  
 <img src="../../../Images/FindPlayer.png" class="icon" alt="Find Player">  
 <span>Find Player</span>  
 </div>  
 <div class="glow-card btn-group" onclick="window.location.href='../JoinGroup/JoinGroup.php'">  
 <img src="../../../Images/JoinGroup.icon.png" class="icon" alt="Join Group">  
 <span>Join Group</span>  
 </div>  
 <div class="glow-card btn-book"onclick="window.location.href='../BookVenue/BookVenue.php'">  
 <img src="../../../Images/SearchVenueIcon.png" class="icon" alt="Book Venue">  
 <span>Book Venue</span>  
 </div>  
 </div>  
  
 <!-- Only the part we modified in secondary-buttons -->  
<div class="secondary-buttons" >  
 <div class="glow-card btn-fav" onclick="window.location.href='../Favorites/Favorites.php'">  
 <img src="../../../Images/Favorite\_icon.png" class="icon" alt="Favorites">  
 <span>Favorites</span>  
 </div>  
 <div class="glow-card btn-chat" onclick="window.location.href='../Chats/Chats.php'">  
 <img src="../../../Images/chat\_icon.png" class="icon" alt="My Chats">  
 <span>My Chats</span>  
 </div>  
 <div class="glow-card btn-bookings" onclick="window.location.href='../MyBookings/MyBookings.php'">  
 <img src="../../../Images/myBooking\_icon.png" class="icon" alt="My Bookings">  
 <span>My Bookings</span>  
 </div>  
 <div class="glow-card btn-switch" onclick="window.location.href='../../auth/User\_Selection\_Page/user-selection.php'">  
 <img src="../../../Images/BackToUserSelecionPurple.png" class="icon" alt="Switch to Facility Owner">  
<span>Back To User Selction Page</span> </div>  
</div>  
  
  
 <footer>  
 <div class="footer-content">  
 <p>&copy; 2025 Book&Play. All rights reserved.</p>  
 </div>  
 </footer>  
  
 <script src="HomePage.js"></script>  
 <button class="floating-admin-btn" id="adminRequestBtn">  
 Request Admin Access  
 </button>  
</body>  
</html>

————————————————————————————————————————

## HomePage\HomePage.js

function toggleProfileMenu() {  
 const menu = document.getElementById('profileMenu');  
 menu.classList.toggle('active');  
 }  
 // Close the menu when clicking outside  
 document.addEventListener('click', function (e) {  
 const profileContainer = document.querySelector('.profile-container');  
 const menu = document.getElementById('profileMenu');  
 if (!profileContainer.contains(e.target)) {  
 menu.classList.remove('active');  
 }  
 });  
document.addEventListener("DOMContentLoaded", () => {  
 const adminBtn = document.getElementById("adminRequestBtn");  
 if (!adminBtn) return; // Exit if the button doesn't exist  
  
 // Check admin status and if a request has already been submitted  
 fetch("../../admin\_actions.php?action=check")  
 .then((res) => res.json())  
 .then((data) => {  
 // Hide the button if the user is an admin  
 if (data.is\_admin) {  
 adminBtn.style.display = "none";  
 return;  
 }  
  
 // Handle button click  
 adminBtn.addEventListener("click", () => {  
 if (data.already\_requested) {  
 // Show waiting message if request already submitted  
 showTempMessage(  
 "🕐 Your admin request is under review. Thank you for your patience.",  
 4000  
 );  
 } else {  
 // Send admin request  
 fetch("../../admin\_actions.php", {  
 method: "POST",  
 headers: {  
 "Content-Type": "application/x-www-form-urlencoded",  
 },  
 body: "action=submit",  
 })  
 .then((res) => res.json())  
 .then((response) => {  
 if (response.success) {  
 showTempMessage(  
 "✅ Your admin request has been submitted. We'll review it as soon as possible.",  
 4000  
 );  
 data.already\_requested = true; // Prevent re-submission  
 } else {  
 showTempMessage(  
 "❌ " + (response.message || "Failed to submit request."),  
 4000  
 );  
 }  
 })  
 .catch((error) => {  
 console.error("Error submitting request:", error);  
 showTempMessage(  
 "❌ Could not connect to server. Try again later.",  
 4000  
 );  
 });  
 }  
 });  
 })  
 .catch((error) => {  
 console.error("Error checking admin status:", error);  
 showTempMessage(  
 "❌ Could not check admin status. Try again later.",  
 4000  
 );  
 });  
});  
  
// Helper function to show temporary messages on screen  
function showTempMessage(message, duration) {  
 const msg = document.createElement("div");  
 msg.textContent = message;  
 msg.style.position = "fixed";  
 msg.style.bottom = "80px";  
 msg.style.right = "20px";  
 msg.style.background = "#1e90ff";  
 msg.style.color = "white";  
 msg.style.padding = "12px 20px";  
 msg.style.borderRadius = "10px";  
 msg.style.boxShadow = "0 4px 12px rgba(0,0,0,0.2)";  
 msg.style.zIndex = 1000;  
 msg.style.fontSize = "14px";  
 document.body.appendChild(msg);  
 setTimeout(() => msg.remove(), duration);  
}  
document.addEventListener("DOMContentLoaded", () => {  
 fetch("HomePage.php?action=get\_user\_image")  
 .then((res) => res.json())  
 .then((data) => {  
 const imgElement = document.getElementById("userProfileImage");  
 const usernameSpan = document.getElementById("usernameDisplay");  
 if (imgElement && data.image) {  
 imgElement.src = `../../uploads/users/${data.image}`;  
 }  
 if (usernameSpan && data.username) {  
 usernameSpan.textContent = data.username;  
 }  
 })  
 .catch((err) => {  
 console.error("Failed to load user image or username", err);  
 });  
});

————————————————————————————————————————

## HomePage\HomePage.php

<?php  
session\_start();  
require\_once '../../../db.php';  
$\_SESSION['previous\_page'] = $\_SERVER['PHP\_SELF'];  
  
// ✅ 1. Check request to fetch user image  
if (isset($\_GET['action']) && $\_GET['action'] === 'get\_user\_image') {  
 $username = $\_SESSION['user\_id'] ?? '';  
  
 if (!$username) {  
 echo json\_encode(['error' => 'Not logged in']);  
 exit;  
 }  
  
 $stmt = $conn->prepare("SELECT user\_image, username FROM users WHERE username = ?");  
 $stmt->bind\_param("s", $username);  
 $stmt->execute();  
 $result = $stmt->get\_result();  
 if ($user = $result->fetch\_assoc()) {  
 echo json\_encode(['image' => $user['user\_image'], 'username' => $user['username']]);  
 } else {  
 echo json\_encode(['error' => 'User not found']);  
 }  
 exit;  
}  
  
// ✅ 2. Check login and load page  
if (!isset($\_SESSION['user\_id'])) {  
 session\_unset();  
 session\_destroy();  
 header('Location: ../../auth/Login\_Page/Login.php');  
 exit();  
}  
  
include 'HomePage.html';  
  
?>

————————————————————————————————————————

## JoinGroup\JoinGroup.css

/\* JoinGroup.css - Additional styles for Join Group page \*/  
  
.search-section {  
 padding: 16px 24px;  
 background: linear-gradient(135deg, rgba(224, 247, 250, 0.3) 0%, rgba(208, 248, 206, 0.3) 100%);  
 border-bottom: 1px solid var(--border-subtle);  
}  
  
.search-container {  
 max-width: 1400px;  
 margin: 0 auto;  
}  
  
.search-box-main {  
 display: flex;  
 align-items: center;  
 background: white;  
 border-radius: var(--radius);  
 padding: 8px;  
 box-shadow: 0 4px 6px var(--shadow-light), 0 1px 3px var(--shadow-medium);  
 border: 1px solid var(--card-border);  
 max-width: 600px;  
 margin: 0 auto;  
}  
  
.player-search {  
 flex: 1;  
 border: none;  
 outline: none;  
 padding: 12px 16px;  
 font-size: 16px;  
 font-family: inherit;  
 background: transparent;  
 color: var(--text-main);  
}  
  
.player-search::placeholder {  
 color: var(--text-muted);  
}  
  
.search-btn {  
 padding: 12px 24px;  
 background: linear-gradient(135deg, var(--primary) 0%, var(--secondary) 100%);  
 color: white;  
 border: none;  
 border-radius: var(--radius-small);  
 font-weight: 600;  
 font-size: 14px;  
 cursor: pointer;  
 transition: var(--transition);  
}  
  
.search-btn:hover {  
 transform: translateY(-1px);  
 box-shadow: 0 4px 12px rgba(30, 144, 255, 0.3);  
}  
  
.layout {  
 display: flex;  
 gap: 24px;  
 max-width: 1400px;  
 margin: 0 auto;  
 padding: 24px;  
}  
  
.sidebar {  
 position: sticky;  
 top: 20px;  
 z-index: 999;  
 width: 200px;  
 background: var(--card-bg);  
 border: 1px solid var(--card-border);  
 border-radius: var(--radius);  
 padding: 20px;  
 box-shadow: 0 4px 6px var(--shadow-light), 0 1px 3px var(--shadow-medium);  
 backdrop-filter: blur(12px);  
 height: fit-content;  
 max-height: calc(100vh - 40px);  
 overflow-y: auto;  
 scrollbar-width: thin;  
 scrollbar-color: var(--primary) transparent;  
 flex-shrink: 0;  
}  
  
.filter-section {  
 margin-bottom: 20px;  
}  
  
.filter-section h3 {  
 font-size: 14px;  
 font-weight: 600;  
 margin-bottom: 12px;  
 color: var(--text-main);  
 border-bottom: 1px solid var(--border-subtle);  
 padding-bottom: 6px;  
 letter-spacing: -0.025em;  
}  
  
.sort-option {  
 display: flex;  
 align-items: center;  
 justify-content: space-between;  
 padding: 8px 12px;  
 margin-bottom: 6px;  
 border-radius: var(--radius-small);  
 background: white;  
 color: var(--text-main);  
 font-size: 12px;  
 font-weight: 500;  
 cursor: pointer;  
 border: 1px solid var(--border-subtle);  
 transition: var(--transition);  
 user-select: none;  
}  
  
.sort-option:hover {  
 background: rgba(30, 144, 255, 0.05);  
 border-color: rgba(30, 144, 255, 0.2);  
 transform: translateY(-1px);  
 box-shadow: 0 2px 8px var(--shadow-light);  
}  
  
.sort-option input {  
 display: none;  
}  
  
.sort-option input:checked + span {  
 background: linear-gradient(135deg, var(--primary), var(--secondary));  
 color: white;  
 font-weight: 600;  
 border-radius: var(--radius-small);  
 padding: 8px 12px;  
 margin: -8px -12px;  
 width: calc(100% + 24px);  
 display: flex;  
 justify-content: space-between;  
 align-items: center;  
 box-shadow: 0 2px 8px rgba(30, 144, 255, 0.25);  
}  
  
.sort-option input:checked + span::after {  
 content: "✓";  
 font-size: 14px;  
 font-weight: bold;  
 opacity: 0.9;  
}  
  
.sort-option:focus-within {  
 outline: 2px solid var(--primary);  
 outline-offset: 2px;  
}  
  
.content {  
 flex: 1;  
 display: grid;  
 grid-template-columns: repeat(auto-fill, minmax(320px, 1fr));  
 gap: 12px;  
}  
  
.venue-card {  
 background: var(--card-bg);  
 border-radius: var(--radius);  
 box-shadow: 0 4px 16px var(--shadow-light);  
 overflow: hidden;  
 position: relative;  
 width: 100%;  
 max-width: 320px;  
 margin: auto;  
 display: flex;  
 flex-direction: column;  
 transition: var(--transition);  
}  
  
.venue-card:hover {  
 transform: translateY(-4px);  
 box-shadow: 0 6px 20px var(--shadow-light);  
}  
  
.venue-card-header {  
 position: absolute;  
 top: 12px;  
 left: 12px;  
 right: 12px;  
 display: flex;  
 justify-content: space-between;  
 align-items: center;  
 z-index: 2;  
}  
  
.group-badge {  
 color: white;  
 font-size: 12px;  
 font-weight: 600;  
 padding: 4px 10px;  
 border-radius: 999px;  
}  
  
.group-badge.public {  
 background: var(--secondary);  
}  
  
.group-badge.private {  
 background: #f59e0b;  
}  
  
.venue-image {  
 width: 100%;  
 height: 180px;  
 background: linear-gradient(45deg, #f0f0f0, #e0e0e0);  
 display: flex;  
 align-items: center;  
 justify-content: center;  
 color: var(--text-muted);  
 font-size: 14px;  
 position: relative;  
}  
  
.venue-image img {  
 width: 100%;  
 height: 200px;  
 object-fit: cover;  
 border-radius: var(--radius-small);  
 transition: var(--transition);  
 cursor: pointer;  
}  
  
.venue-image img:hover {  
 transform: scale(1.02);  
 box-shadow: 0 8px 16px rgba(0, 0, 0, 0.2);  
 filter: brightness(1.05);  
}  
  
/\* Favorite Icon Styles \*/  
.favorite-icon {  
 position: absolute;  
 bottom: 10px;  
 right: 15px;  
 font-size: 28px;  
 color: rgba(255, 255, 255, 0.8);  
 text-shadow: 0 0 5px rgba(0, 0, 0, 0.6);  
 cursor: pointer;  
 transition: 0.3s ease;  
 z-index: 2;  
}  
  
.favorite-icon:hover {  
 transform: scale(1.15);  
}  
  
.favorite-icon.active {  
 color: red;  
 text-shadow: 0 0 8px rgba(0, 0, 0, 0.6);  
}  
  
.venue-info {  
 padding: 16px;  
 display: flex;  
 flex-direction: column;  
 gap: 8px;  
}  
  
.venue-title {  
 font-size: 16px;  
 font-weight: 700;  
 color: var(--text-main);  
 margin: 0;  
}  
  
.venue-location {  
 font-size: 13px;  
 color: var(--text-muted);  
 margin: 0;  
}  
  
.venue-tags {  
 display: flex;  
 align-items: center;  
 justify-content: space-between;  
 margin-top: 6px;  
}  
  
.tag {  
 background: rgba(30, 144, 255, 0.1);  
 color: var(--primary);  
 font-size: 12px;  
 padding: 4px 10px;  
 border-radius: var(--radius-small);  
}  
  
.player-count {  
 font-size: 12px;  
 color: var(--text-muted);  
}  
  
.venue-footer {  
 display: flex;  
 justify-content: space-between;  
 align-items: center;  
 margin-top: 12px;  
}  
  
.venue-price {  
 font-size: 18px;  
 font-weight: 700;  
 color: var(--secondary);  
}  
  
.venue-price .per {  
 font-size: 12px;  
 color: var(--text-muted);  
 margin-left: 2px;  
}  
  
.join-btn {  
 background: linear-gradient(135deg, var(--primary), var(--secondary));  
 color: white;  
 padding: 10px 16px;  
 border: none;  
 border-radius: var(--radius-small);  
 font-size: 14px;  
 font-weight: 600;  
 cursor: pointer;  
 transition: var(--transition);  
}  
  
.join-btn:hover {  
 transform: translateY(-1px);  
 box-shadow: 0 4px 12px rgba(30, 144, 255, 0.2);  
}  
  
/\* Modal Styles \*/  
.modal {  
 display: none;  
 position: fixed;  
 z-index: 1000;  
 left: 0;  
 top: 0;  
 width: 100%;  
 height: 100%;  
 background-color: rgba(0, 0, 0, 0.5);  
 backdrop-filter: blur(5px);  
}  
  
.modal-content {  
 background-color: var(--card-bg);  
 margin: 15% auto;  
 border-radius: var(--radius);  
 width: 90%;  
 max-width: 400px;  
 box-shadow: 0 10px 30px var(--shadow-strong);  
 animation: modalFadeIn 0.3s ease;  
}  
  
@keyframes modalFadeIn {  
 from {  
 opacity: 0;  
 transform: translateY(-50px);  
 }  
 to {  
 opacity: 1;  
 transform: translateY(0);  
 }  
}  
  
.modal-header {  
 padding: 20px 20px 0 20px;  
 display: flex;  
 justify-content: space-between;  
 align-items: center;  
}  
  
.modal-header h3 {  
 margin: 0;  
 color: var(--text-main);  
 font-size: 18px;  
 font-weight: 600;  
}  
  
.close {  
 color: var(--text-muted);  
 font-size: 28px;  
 font-weight: bold;  
 cursor: pointer;  
 line-height: 1;  
}  
  
.close:hover {  
 color: var(--text-main);  
}  
  
.modal-body {  
 padding: 20px;  
}  
  
.modal-body p {  
 margin-bottom: 15px;  
 color: var(--text-main);  
 font-size: 14px;  
}  
  
#accessCodeInput {  
 width: 100%;  
 padding: 12px;  
 border: 1px solid var(--border-subtle);  
 border-radius: var(--radius-small);  
 font-size: 16px;  
 font-family: inherit;  
 margin-bottom: 20px;  
 background: white;  
 color: var(--text-main);  
}  
  
#accessCodeInput:focus {  
 outline: none;  
 border-color: var(--primary);  
 box-shadow: 0 0 0 3px rgba(30, 144, 255, 0.1);  
}  
  
.modal-buttons {  
 display: flex;  
 gap: 10px;  
 justify-content: flex-end;  
}  
  
.modal-btn {  
 padding: 10px 20px;  
 border: none;  
 border-radius: var(--radius-small);  
 font-size: 14px;  
 font-weight: 600;  
 cursor: pointer;  
 transition: var(--transition);  
}  
  
.modal-btn.cancel {  
 background: #f3f4f6;  
 color: var(--text-main);  
}  
  
.modal-btn.cancel:hover {  
 background: #e5e7eb;  
}  
  
.modal-btn.confirm {  
 background: linear-gradient(135deg, var(--primary), var(--secondary));  
 color: white;  
}  
  
.modal-btn.confirm:hover {  
 transform: translateY(-1px);  
 box-shadow: 0 4px 12px rgba(30, 144, 255, 0.2);  
}  
  
/\* Responsive Design \*/  
@media (max-width: 768px) {  
 .layout {  
 flex-direction: column;  
 padding: 16px;  
 }  
  
 .sidebar {  
 position: static;  
 width: 100%;  
 max-height: none;  
 }  
  
 .content {  
 grid-template-columns: 1fr;  
 }  
  
 .search-section {  
 padding: 12px 16px;  
 }  
  
 .search-box-main {  
 padding: 6px;  
 }  
  
 .player-search {  
 padding: 10px 12px;  
 font-size: 14px;  
 }  
  
 .search-btn {  
 padding: 10px 16px;  
 font-size: 12px;  
 }  
  
 .modal-content {  
 margin: 25% auto;  
 width: 95%;  
 }  
  
 .favorite-icon {  
 font-size: 24px;  
 bottom: 8px;  
 right: 12px;  
 }  
}

————————————————————————————————————————

## JoinGroup\JoinGroup.html

<!DOCTYPE html>  
<html lang="en">  
<head>  
 <meta charset="UTF-8" />  
 <meta name="viewport" content="width=device-width, initial-scale=1.0"/>  
 <title>Join Group - Book&Play</title>  
 <link rel="stylesheet" href="JoinGroup.css" />  
</head>  
<body>  
  
 <!-- Search Bar -->  
 <section class="search-section">  
 <div class="search-container">  
 <div class="search-box-main">  
 <input type="text" class="player-search" placeholder="🔍 Search for group by name..." id="playerSearch">  
 </div>  
 </div>  
 </section>  
  
 <!-- Layout: Sidebar + Venue Cards -->  
 <div class="layout">  
   
 <!-- Sidebar Filters -->  
 <aside class="sidebar">  
 <!-- Sort by Group Type -->  
 <div class="filter-section">  
 <h3>Sort by Group Type</h3>  
 <label class="sort-option">  
 <input type="radio" name="public-sort" id="all-groups" checked>  
 <span>All</span>  
 </label>  
 <label class="sort-option">  
 <input type="radio" name="public-sort" id="only-public">  
 <span>Public</span>  
 </label>  
 <label class="sort-option">  
 <input type="radio" name="public-sort" id="only-private">  
 <span>Private</span>  
 </label>  
 </div>  
  
 <!-- Sort by Price -->  
 <div class="filter-section">  
 <h3>Sort by Price</h3>  
 <label class="sort-option">  
 <input type="radio" name="price-sort" id="price-low">  
 <span>Low to High</span>  
 </label>  
 <label class="sort-option">  
 <input type="radio" name="price-sort" id="price-high">  
 <span>High to Low</span>  
 </label>  
 </div>  
  
 <!-- Sort by Rating -->  
 <div class="filter-section">  
 <h3>Sort by Rating</h3>  
 <label class="sort-option">  
 <input type="radio" name="rating-sort" id="rating-low">  
 <span>Low to High</span>  
 </label>  
 <label class="sort-option">  
 <input type="radio" name="rating-sort" id="rating-high">  
 <span>High to Low</span>  
 </label>  
 </div>  
  
 <!-- Sort by Distance -->  
 <div class="filter-section">  
 <h3>Sort by Distance</h3>  
 <label class="sort-option">  
 <input type="radio" name="distance-sort" id="distance-near">  
 <span>Nearest First</span>  
 </label>  
 <label class="sort-option">  
 <input type="radio" name="distance-sort" id="distance-far">  
 <span>Farthest First</span>  
 </label>  
 </div>  
 </aside>  
  
 <!-- Venue Cards Section -->  
 <div class="content" id="groupsContainer"></div>  
  
 <!-- Private Group Access Modal -->  
 <div id="privateModal" class="modal">  
 <div class="modal-content">  
 <div class="modal-header">  
 <h3>Enter Access Code</h3>  
 <span class="close" onclick="closeModal()">&times;</span>  
 </div>  
 <div class="modal-body">  
 <p>This is a private group. Please enter the access code to join:</p>  
 <input type="text" id="accessCodeInput" placeholder="Enter access code" />  
 <div class="modal-buttons">  
 <button class="modal-btn cancel" onclick="closeModal()">Cancel</button>  
 <button class="modal-btn confirm" onclick="validateAccessCode()">Join Group</button>  
 </div>  
 </div>  
 </div>  
 </div>  
  
 <script src="JoinGroup.js"></script>  
</body>  
</html>

————————————————————————————————————————

## JoinGroup\JoinGroup.js

let allGroups = [];  
let userLocation = null;  
let selectedGroup = null;  
  
function toggleFavorite(element) {  
 element.classList.toggle("active");  
}  
  
// ✅ Filter venues by selected sports  
function filterVenuesBySports(selectedSports) {  
 if (!selectedSports || selectedSports.length === 0) {  
 renderGroups(allGroups);  
 return;  
 }  
   
 const filtered = allGroups.filter(group =>   
 selectedSports.includes(group.SportCategory)  
 );  
 renderGroups(filtered);  
}  
  
// ✅ View Booking Details (for image clicks)  
function viewBookingDetails(booking\_id) {  
 const url = `../BookingDetails/BookingDetails.php?booking\_id=${encodeURIComponent(booking\_id)}&view\_only=true`;  
 window.location.href = url;  
}  
  
window.onload = function () {  
 getUserLocation().then(location => {  
 userLocation = location;  
  
 fetch("JoinGroupAPI.php")  
 .then(res => res.json())  
 .then(data => {  
 if (data.success && Array.isArray(data.groups)) {  
 allGroups = data.groups.map(group => ({  
 ...group,  
 distance: calculateDistance(  
 userLocation.lat,  
 userLocation.lng,  
 parseFloat(group.latitude),  
 parseFloat(group.longitude)  
 )  
 }));  
 renderGroups(allGroups);  
 } else {  
 console.error("Invalid response format:", data);  
 }  
 })  
 .catch(err => console.error("Failed to fetch groups:", err));  
 });  
};  
  
function renderGroups(groups) {  
 const container = document.getElementById("groupsContainer");  
 container.innerHTML = "";  
  
 groups.forEach(group => {  
 const isPrivate = group.privacy === 'private';  
 const current = group.current\_members;  
 const max = group.max\_members;  
  
 const card = document.createElement("div");  
 card.className = "venue-card";  
 card.setAttribute("data-group-type", group.privacy);  
 if (isPrivate) card.setAttribute("data-access-code", group.group\_password);  
  
 const groupDataEncoded = JSON.stringify(group).replace(/'/g, "&apos;");  
  
 card.innerHTML = `  
 <div class="venue-card-header">  
 <span class="group-badge ${group.privacy}">${group.privacy.toUpperCase()}</span>  
 </div>  
 <div class="venue-image">  
 <img src="${group.image\_url.replace(/\\/g, "/")}" alt="Venue Image" onclick="viewBookingDetails(${group.booking\_id})" style="cursor: pointer;">  
 </div>  
 <div class="venue-info">  
 <h3 class="venue-title">${group.group\_name}</h3>  
 <p class="venue-location">📍 ${group.location}</p>  
 <div class="venue-tags">  
 <span class="tag">${group.SportCategory}</span>  
 <span class="player-count">  
 ⭐ ${group.rating ? parseFloat(group.rating).toFixed(1) : "N/A"} |  
 📍 ${group.distance ? group.distance.toFixed(2) + " km" : "N/A"}  
 </span>  
 </div>  
 <div class="venue-footer">  
 <span class="venue-price">₪${group.price}<span class="per">/person</span></span>  
 <button class="join-btn" data-group='${groupDataEncoded}' onclick="handleJoinClick(this)">Join Group</button>  
 </div>  
 </div>  
 `;  
  
 container.appendChild(card);  
 });  
}  
  
// ✅ Handle Join Button Click Safely  
function handleJoinClick(button) {  
 try {  
 const groupData = JSON.parse(button.dataset.group.replace(/&apos;/g, "'"));  
 joinGroup(groupData);  
 } catch (e) {  
 console.error("Failed to parse group data:", e);  
 }  
}  
  
// ✅ Join Group  
function joinGroup(group) {  
 selectedGroup = group;  
  
 if (group.privacy === 'private') {  
 document.getElementById('accessCodeInput').value = '';  
 document.getElementById('privateModal').style.display = 'block';  
 } else {  
 redirectToBooking(group.booking\_id);  
 }  
}  
  
// ✅ Validate Private Code  
function validateAccessCode() {  
 const enteredCode = document.getElementById('accessCodeInput').value.trim();  
  
 if (!enteredCode) {  
 alert("Please enter an access code.");  
 return;  
 }  
  
 if (enteredCode === selectedGroup.group\_password) {  
 closeModal();  
 redirectToBooking(selectedGroup.booking\_id);  
 } else {  
 alert("The code you entered is not correct for this group.");  
 document.getElementById('accessCodeInput').value = '';  
 document.getElementById('accessCodeInput').focus();  
 }  
}  
  
function closeModal() {  
 document.getElementById('privateModal').style.display = 'none';  
}  
  
function redirectToBooking(booking\_id) {  
 const url = `../BookingDetails/BookingDetails.php?booking\_id=${encodeURIComponent(booking\_id)}`;  
 window.location.href = url;  
}  
  
// 🌍 Location helpers  
function getUserLocation() {  
 return new Promise((resolve, reject) => {  
 if (!navigator.geolocation) {  
 alert("Geolocation is not supported.");  
 reject("No geolocation");  
 } else {  
 navigator.geolocation.getCurrentPosition(  
 position => {  
 resolve({  
 lat: position.coords.latitude,  
 lng: position.coords.longitude  
 });  
 },  
 error => {  
 alert("Please allow location access to sort by distance.");  
 reject(error);  
 }  
 );  
 }  
 });  
}  
  
function calculateDistance(lat1, lon1, lat2, lon2) {  
 function toRad(x) {  
 return x \* Math.PI / 180;  
 }  
  
 const R = 6371;  
 const dLat = toRad(lat2 - lat1);  
 const dLon = toRad(lon2 - lon1);  
 const a =  
 Math.sin(dLat / 2) \*\* 2 +  
 Math.cos(toRad(lat1)) \*  
 Math.cos(toRad(lat2)) \*  
 Math.sin(dLon / 2) \*\* 2;  
 const c = 2 \* Math.atan2(Math.sqrt(a), Math.sqrt(1 - a));  
 return R \* c;  
}  
  
// ✅ Search  
document.getElementById('playerSearch').addEventListener('input', function () {  
 const query = this.value.trim().toLowerCase();  
 const filtered = allGroups.filter(group =>  
 group.group\_name.toLowerCase().startsWith(query)  
 );  
 renderGroups(filtered);  
});  
  
// ✅ Sort  
document.getElementById('price-low').addEventListener('change', () => {  
 renderGroups([...allGroups].sort((a, b) => a.price - b.price));  
});  
  
document.getElementById('price-high').addEventListener('change', () => {  
 renderGroups([...allGroups].sort((a, b) => b.price - a.price));  
});  
  
document.getElementById('only-public').addEventListener('change', () => {  
 renderGroups(allGroups.filter(g => g.privacy === 'public'));  
});  
  
document.getElementById('only-private').addEventListener('change', () => {  
 renderGroups(allGroups.filter(g => g.privacy === 'private'));  
});  
  
document.getElementById('all-groups').addEventListener('change', () => {  
 renderGroups(allGroups);  
});  
  
document.getElementById('rating-low').addEventListener('change', () => {  
 renderGroups([...allGroups].sort((a, b) => (a.rating || 0) - (b.rating || 0)));  
});  
  
document.getElementById('rating-high').addEventListener('change', () => {  
 renderGroups([...allGroups].sort((a, b) => (b.rating || 0) - (a.rating || 0)));  
});  
  
document.getElementById('distance-near').addEventListener('change', () => {  
 renderGroups([...allGroups].sort((a, b) => a.distance - b.distance));  
});  
  
document.getElementById('distance-far').addEventListener('change', () => {  
 renderGroups([...allGroups].sort((a, b) => b.distance - a.distance));  
});

————————————————————————————————————————

## JoinGroup\JoinGroup.php

<?php  
require\_once '../../../db.php';  
require\_once '../../../components/sports-scroll.php';  
include '../navbar.php';  
include 'JoinGroup.html';  
?>

————————————————————————————————————————

## JoinGroup\JoinGroupAPI.php

<?php  
header('Content-Type: application/json');  
include '../../../db.php';  
  
$query = "  
 SELECT   
 g.\*,  
 f.place\_name,  
 f.location,  
 f.image\_url,  
 f.SportCategory,  
 b.Total\_Price AS price,  
 f.latitude,  
 f.longitude,  
 (SELECT AVG(r.rating\_value) FROM ratings r WHERE r.facilities\_id = f.facilities\_id) AS rating,  
 (SELECT COUNT(\*) FROM group\_members gm WHERE gm.group\_id = g.group\_id) AS current\_members  
 FROM groups g  
 JOIN sportfacilities f ON g.facilities\_id = f.facilities\_id  
 LEFT JOIN bookings b ON g.booking\_id = b.booking\_id  
";  
  
$result = mysqli\_query($conn, $query);  
  
$groups = [];  
  
while ($row = mysqli\_fetch\_assoc($result)) {  
 // Fix image path to include the correct uploads directory  
 if ($row['image\_url'] && !empty($row['image\_url'])) {  
 $row['image\_url'] = '../../../uploads/venues/' . $row['image\_url'];  
 } else {  
 $row['image\_url'] = '../../../Images/staduim\_icon.png'; // Default image  
 }  
   
 $groups[] = $row;  
}  
  
echo json\_encode([  
 "success" => true,  
 "groups" => $groups  
]);  
?>

————————————————————————————————————————

## MyBookings\MyBookings.css

/\* Color Variables \*/  
:root {  
 --status-cancelled: #ef4444;  
  
}  
  
\* {  
 margin: 0;  
 padding: 0;  
 box-sizing: border-box;  
}  
  
  
/\* Main Container \*/  
.container {  
 max-width: 1200px;  
 margin: 0 auto;  
 padding: 32px 24px;  
}  
  
/\* Page Header \*/  
.page-header {  
 margin-bottom: 32px;  
}  
  
.page-title {  
 font-size: 32px;  
 font-weight: 700;  
 margin-bottom: 8px;  
 background: linear-gradient(135deg, var(--primary), var(--secondary));  
 -webkit-background-clip: text;  
 -webkit-text-fill-color: transparent;  
}  
  
.page-subtitle {  
 color: var(--text-muted);  
 font-size: 16px;  
 font-weight: 500;  
}  
  
/\* Stats Cards \*/  
.stat-card {  
 background: var(--card-bg);  
 border: 1px solid var(--card-border);  
 border-radius: var(--radius);  
 padding: 20px;  
 box-shadow: 0 4px 20px var(--shadow-light);  
 backdrop-filter: blur(10px);  
 transition: var(--transition);  
}  
  
.stat-card:hover {  
 transform: translateY(-2px);  
 box-shadow: 0 8px 25px var(--shadow-light);  
}  
  
.stat-value {  
 font-size: 24px;  
 font-weight: 700;  
 margin-bottom: 4px;  
}  
  
.stat-label {  
 color: var(--text-muted);  
 font-size: 14px;  
 font-weight: 500;  
}  
  
.stat-card.total .stat-value {   
 color: var(--primary);   
}  
  
.stat-card.confirmed .stat-value {   
 color: var(--status-confirmed);   
}  
  
.stat-card.pending .stat-value {   
 color: var(--status-pending);   
}  
  
.stat-card.cancelled .stat-value {   
 color: var(--status-cancelled);   
}  
  
/\* Filters \*/  
.filters-section {  
 background: var(--card-bg);  
 border: 1px solid var(--card-border);  
 border-radius: var(--radius);  
 padding: 20px;  
 margin-bottom: 24px;  
 box-shadow: 0 4px 20px var(--shadow-light);  
 backdrop-filter: blur(10px);  
}  
  
.filters-grid {  
 display: grid;  
 grid-template-columns: repeat(auto-fit, minmax(200px, 1fr));  
 gap: 16px;  
 align-items: end;  
}  
  
.filter-group {  
 display: flex;  
 flex-direction: column;  
 gap: 8px;  
}  
  
.filter-label {  
 font-weight: 500;  
 color: var(--text-main);  
 font-size: 14px;  
}  
  
.filter-input {  
 padding: 10px 12px;  
 border: 1px solid var(--card-border);  
 border-radius: var(--radius-small);  
 font-size: 14px;  
 transition: var(--transition);  
 background: rgba(255, 255, 255, 0.8);  
}  
  
.filter-input:focus {  
 outline: none;  
 border-color: var(--primary);  
 box-shadow: 0 0 0 3px rgba(30, 144, 255, 0.1);  
}  
  
/\* Bookings List \*/  
.bookings-section {  
 display: flex;  
 flex-direction: column;  
 gap: 16px;  
}  
  
.booking-card {  
 background: var(--card-bg);  
 border: 1px solid var(--card-border);  
 border-radius: var(--radius);  
 overflow: hidden;  
 box-shadow: 0 4px 20px var(--shadow-light);  
 backdrop-filter: blur(10px);  
 transition: var(--transition);  
}  
  
.booking-card:hover {  
 transform: translateY(-2px);  
 box-shadow: 0 8px 25px var(--shadow-light);  
}  
  
.booking-header {  
 display: grid;  
 grid-template-columns: auto 1fr auto auto;  
 align-items: center;  
 padding: 20px;  
 gap: 16px;  
}  
  
.venue-image {  
 width: 60px;  
 height: 60px;  
 border-radius: var(--radius-small);  
 overflow: hidden;  
 background: linear-gradient(45deg, var(--bg-light), var(--bg-secondary));  
}  
  
.venue-image img {  
 width: 100%;  
 height: 100%;  
 object-fit: cover;  
}  
  
.booking-info {  
 flex: 1;  
}  
  
.venue-name {  
 font-size: 18px;  
 font-weight: 600;  
 margin-bottom: 4px;  
 color: var(--text-main);  
}  
  
.booking-details {  
 color: var(--text-muted);  
 font-size: 14px;  
 display: flex;  
 gap: 16px;  
 flex-wrap: wrap;  
}  
  
.booking-detail {  
 display: flex;  
 align-items: center;  
 gap: 4px;  
}  
  
.booking-price {  
 font-size: 20px;  
 font-weight: 700;  
 background: linear-gradient(135deg, var(--primary), var(--secondary));  
 -webkit-background-clip: text;  
 -webkit-text-fill-color: transparent;  
}  
  
.status-cancelled {  
 background: rgba(239, 68, 68, 0.1);  
 color: var(--status-cancelled);  
 border: 1px solid rgba(239, 68, 68, 0.2);  
}  
  
.status-completed {  
 background: rgba(139, 92, 246, 0.1);  
 color: var(--status-completed);  
 border: 1px solid rgba(139, 92, 246, 0.2);  
}  
  
.booking-actions {  
 display: flex;  
 gap: 8px;  
}  
  
.action-btn {  
 background: linear-gradient(135deg, var(--primary), var(--secondary));  
 color: white;  
 padding: 8px 12px;  
 border: 1px solid var(--card-border);  
 border-radius: var(--radius-small);  
 font-size: 12px;  
 font-weight: 500;  
 cursor: pointer;  
 transition: var(--transition);  
 text-decoration: none;  
}  
  
.action-btn:hover {  
 transform: translateY(-1px);  
 box-shadow: 0 2px 8px var(--shadow-light);  
}  
  
.action-btn.danger {  
 background: rgba(239, 68, 68, 0.1);  
 color: var(--status-cancelled);  
 border-color: rgba(239, 68, 68, 0.2);  
}  
  
.action-btn.danger:hover {  
 background: var(--status-cancelled);  
 color: rgb(252, 8, 8);  
}  
  
/\* Empty State \*/  
.empty-state {  
 text-align: center;  
 padding: 60px 20px;  
 background: var(--card-bg);  
 border: 1px solid var(--card-border);  
 border-radius: var(--radius);  
 box-shadow: 0 4px 20px var(--shadow-light);  
 backdrop-filter: blur(10px);  
}  
  
.empty-state-icon {  
 font-size: 48px;  
 margin-bottom: 16px;  
 opacity: 0.5;  
}  
  
.empty-state-title {  
 font-size: 20px;  
 font-weight: 600;  
 margin-bottom: 8px;  
 color: var(--text-main);  
}  
  
.empty-state-subtitle {  
 color: var(--text-muted);  
 margin-bottom: 20px;  
}  
  
.empty-state-btn {  
 padding: 12px 24px;  
 background: linear-gradient(135deg, var(--primary), var(--secondary));  
 color: white;  
 border: none;  
 border-radius: var(--radius);  
 font-weight: 600;  
 text-decoration: none;  
 transition: var(--transition);  
}  
  
.empty-state-btn:hover {  
 transform: translateY(-2px);  
 box-shadow: 0 8px 25px rgba(30, 144, 255, 0.3);  
}  
  
/\* Responsive Design \*/  
@media (max-width: 480px) {  
 .page-title {  
 font-size: 24px;  
 }  
  
 .venue-image {  
 width: 50px;  
 height: 50px;  
 }  
}

————————————————————————————————————————

## MyBookings\MyBookings.html

<!DOCTYPE html>  
<html lang="en">  
<head>  
 <meta charset="UTF-8">  
 <meta name="viewport" content="width=device-width, initial-scale=1.0">  
 <title>My Bookings - Book&Play</title>  
 <link href="https://fonts.googleapis.com/css2?family=Inter:wght@300;400;500;600;700&display=swap" rel="stylesheet">  
 <link rel="stylesheet" href="MyBookings.css">  
</head>  
<body>  
 <!-- Navbar -->  
  
 <!-- Main Container -->  
 <div class="container">  
 <!-- Page Header -->  
 <div class="page-header">  
 <h1 class="page-title">My Bookings</h1>  
 <p class="page-subtitle">Manage and track all your venue reservations</p>  
 </div>  
  
 <!-- Filters Section -->  
 <div class="filters-section">  
 <div class="filters-grid">  
 <div class="filter-group">  
 <label class="filter-label">Search Booking ID</label>  
 <input type="text" class="filter-input" placeholder="Enter Booking ID..." id="searchInput">  
 </div>  
 </div>  
 </div>  
  
 <!-- Bookings List -->  
 <div class="bookings-section" id="bookingsList">  
 <!-- Database generated content will go here -->  
   
 </div>  
 </div>  
 <script src="MyBookings.js" defer></script>  
</body>  
</html>

————————————————————————————————————————

## MyBookings\MyBookings.js

document.addEventListener('DOMContentLoaded', function () {  
 const searchInput = document.getElementById('searchInput');  
 let allBookings = [];  
  
 fetch('MyBookings.php?ajax=1')  
 .then(res => res.json())  
 .then(data => {  
 const list = document.getElementById('bookingsList');  
  
 if (!data.success || !data.bookings.length) {  
 list.innerHTML = `  
 <div class="empty-state">  
 <div class="empty-state-icon">📭</div>  
 <div class="empty-state-title">No bookings found</div>  
 <div class="empty-state-subtitle">You haven't made any bookings yet.</div>  
 <a href="../BookVenue/BookVenue.php" class="empty-state-btn">Explore Venues</a>  
 </div>`;  
 return;  
 }  
  
 allBookings = data.bookings;  
 renderBookings(allBookings);  
 })  
 .catch(error => {  
 console.error('Error loading bookings:', error);  
 });  
  
 // Filter when typing  
 searchInput.addEventListener('input', function () {  
 const query = searchInput.value.trim();  
 if (query === '') {  
 renderBookings(allBookings);  
 } else {  
 const filtered = allBookings.filter(booking =>  
 booking.booking\_id.toString().startsWith(query)  
 );  
 renderBookings(filtered);  
 }  
 });  
  
 function renderBookings(bookings) {  
 const list = document.getElementById('bookingsList');  
 if (!bookings.length) {  
 list.innerHTML = `  
 <div class="empty-state">  
 <div class="empty-state-icon">📭</div>  
 <div class="empty-state-title">No results</div>  
 <div class="empty-state-subtitle">No bookings found with that ID.</div>  
 </div>`;  
 return;  
 }  
  
 list.innerHTML = bookings.map(booking => `  
 <div class="booking-card">  
 <div class="booking-header">  
 <div class="venue-image">  
 <img src="../../../uploads/venues/${booking.image\_url}" alt="${booking.place\_name}">  
 </div>  
 <div class="booking-info">  
 <div class="venue-name">${booking.place\_name}</div>  
 <div class="booking-details">  
 <div class="booking-detail">📅 ${formatDate(booking.booking\_date)}</div>  
 <div class="booking-detail">⏰ ${booking.start\_time} - ${booking.end\_time}</div>  
 <div class="booking-detail">📍 ${booking.location ?? 'Location not available'}</div>  
 <div class="booking-detail">🆔 Booking ID: ${booking.booking\_id}</div>  
 </div>  
 </div>  
 <div class="booking-price">₪${booking.Total\_Price}</div>  
 <div class="booking-actions">  
 <button class="action-btn" onclick="viewBookingDetails(${booking.booking\_id})">View Details</button>  
 </div>  
 </div>  
 </div>  
 `).join('');  
 }  
  
 function formatDate(dateStr) {  
 const options = { weekday: 'short', year: 'numeric', month: 'short', day: 'numeric' };  
 const date = new Date(dateStr);  
 return date.toLocaleDateString(undefined, options);  
 }  
  
 // Function to navigate to booking details page  
 window.viewBookingDetails = function(bookingId) {  
 if (!bookingId) {  
 alert("Booking ID not available");  
 return;  
 }  
 window.location.href = `../BookingDetails/BookingDetails.php?booking\_id=${bookingId}`;  
 };  
});

————————————————————————————————————————

## MyBookings\MyBookings.php

<?php  
session\_start();  
require\_once '../../../db.php';  
  
// Check if this is an AJAX request using query string  
if (isset($\_GET['ajax']) && $\_GET['ajax'] === '1') {  
 header('Content-Type: application/json');  
  
 if (!isset($\_SESSION['user\_id'])) {  
 echo json\_encode(['success' => false, 'message' => 'User not logged in']);  
 exit;  
 }  
  
 $username = $\_SESSION['user\_id'];  
  
 $sql = "SELECT b.\*, s.place\_name, s.image\_url, s.price, s.location  
 FROM bookings b  
 JOIN sportfacilities s ON b.facilities\_id = s.facilities\_id  
 WHERE b.username = ?  
 ORDER BY b.booking\_date DESC";  
  
 $stmt = $conn->prepare($sql);  
 $stmt->bind\_param("s", $username);  
 $stmt->execute();  
 $result = $stmt->get\_result();  
  
 $bookings = [];  
 while ($row = $result->fetch\_assoc()) {  
 $bookings[] = $row;  
 }  
  
 echo json\_encode(['success' => true, 'bookings' => $bookings]);  
 exit;  
}  
  
// Not AJAX request? Load full HTML page  
include '../navbar.php';  
include 'MyBookings.html';  
exit;  
  
?>

————————————————————————————————————————

## MyFriends\MyFriends.css

/\* MyFriends.css - FreshGlow UI Style \*/  
  
/\* Color Variables \*/  
:root {  
 --primary: #1e90ff;  
 --secondary: #22c55e;  
 --bg-light: #e0f7fa;  
 --bg-secondary: #d0f8ce;  
 --card-bg: rgba(255, 255, 255, 0.95);  
 --card-border: rgba(209, 213, 219, 0.6);  
 --text-main: #0f172a;  
 --text-muted: #64748b;  
 --text-light: #94a3b8;  
 --shadow-light: rgba(30, 144, 255, 0.08);  
 --shadow-medium: rgba(0, 0, 0, 0.08);  
 --shadow-strong: rgba(0, 0, 0, 0.15);  
 --radius: 12px;  
 --radius-medium: 10px;  
 --radius-small: 8px;  
 --transition: all 0.3s cubic-bezier(0.4, 0, 0.2, 1);  
 --border-subtle: rgba(0, 0, 0, 0.06);  
 --navbar-height: 80px;  
 --success: #10b981;  
 --danger: #ef4444;  
 --warning: #f59e0b;  
}  
  
/\* Reset \*/  
\* {  
 margin: 0;  
 padding: 0;  
 box-sizing: border-box;  
}  
  
body {  
 font-family: 'Inter', sans-serif;  
 background: linear-gradient(135deg, var(--bg-light), var(--bg-secondary));  
 background-attachment: fixed;  
 background-size: 100% 100%;  
 background-repeat: no-repeat;  
 color: var(--text-main);  
 min-height: 100vh;  
 line-height: 1.6;  
 -webkit-font-smoothing: antialiased;  
 padding-top: var(--navbar-height);  
}  
  
/\* NAVBAR \*/  
  
.container {  
 max-width: 1200px;  
 margin: 0 auto;  
 padding: 24px;  
}  
  
/\* Page Header \*/  
.page-header {  
 text-align: center;  
 margin-bottom: 32px;  
}  
  
.page-header h1 {  
 font-size: 32px;  
 font-weight: 700;  
 background: linear-gradient(135deg, var(--primary), var(--secondary));  
 -webkit-background-clip: text;  
 -webkit-text-fill-color: transparent;  
 margin-bottom: 24px;  
}  
  
.header-stats {  
 display: flex;  
 justify-content: center;  
 gap: 24px;  
 flex-wrap: wrap;  
}  
  
.stat-card {  
 background: var(--card-bg);  
 border-radius: var(--radius);  
 padding: 20px 24px;  
 box-shadow: 0 2px 8px var(--shadow-light);  
 border: 1px solid var(--border-subtle);  
 backdrop-filter: blur(12px);  
 display: flex;  
 flex-direction: column;  
 align-items: center;  
 gap: 8px;  
 min-width: 100px;  
 transition: var(--transition);  
}  
  
.stat-card:hover {  
 transform: translateY(-2px);  
 box-shadow: 0 4px 12px var(--shadow-light);  
}  
  
.stat-number {  
 font-size: 24px;  
 font-weight: 700;  
 color: var(--primary);  
}  
  
.stat-label {  
 font-size: 13px;  
 color: var(--text-muted);  
 font-weight: 500;  
}  
  
/\* Controls Section \*/  
.controls-section {  
 display: flex;  
 gap: 20px;  
 margin-bottom: 32px;  
 align-items: center;  
 flex-wrap: wrap;  
}  
  
.search-bar {  
 flex: 1;  
 position: relative;  
 background: var(--card-bg);  
 border-radius: 24px;  
 padding: 4px;  
 box-shadow: 0 2px 8px var(--shadow-light);  
 border: 1px solid var(--border-subtle);  
 backdrop-filter: blur(12px);  
 min-width: 300px;  
}  
  
.search-bar input {  
 width: calc(100% - 50px);  
 padding: 14px 20px;  
 border: none;  
 outline: none;  
 background: transparent;  
 font-size: 15px;  
 color: var(--text-main);  
 font-weight: 500;  
}  
  
.search-bar input::placeholder {  
 color: var(--text-light);  
}  
  
.search-btn {  
 position: absolute;  
 right: 4px;  
 top: 50%;  
 transform: translateY(-50%);  
 background: linear-gradient(135deg, var(--primary), var(--secondary));  
 border: none;  
 border-radius: 50%;  
 width: 40px;  
 height: 40px;  
 cursor: pointer;  
 font-size: 16px;  
 transition: var(--transition);  
 box-shadow: 0 2px 8px rgba(30, 144, 255, 0.25);  
 color: white;  
}  
  
.search-btn:hover {  
 transform: translateY(-50%) scale(1.05);  
 box-shadow: 0 4px 12px rgba(30, 144, 255, 0.3);  
}  
  
.filter-buttons {  
 display: flex;  
 gap: 8px;  
}  
  
.filter-btn {  
 padding: 12px 20px;  
 border: 1px solid var(--border-subtle);  
 background: var(--card-bg);  
 border-radius: 24px;  
 cursor: pointer;  
 font-weight: 500;  
 font-size: 14px;  
 color: var(--text-main);  
 transition: var(--transition);  
 box-shadow: 0 1px 3px var(--shadow-light);  
}  
  
.filter-btn:hover {  
 transform: translateY(-1px);  
 box-shadow: 0 4px 12px var(--shadow-light);  
}  
  
.filter-btn.active {  
 background: linear-gradient(135deg, var(--primary), var(--secondary));  
 color: white;  
 border-color: transparent;  
 box-shadow: 0 2px 8px rgba(30, 144, 255, 0.25);  
}  
  
/\* Section Cards \*/  
.section-card {  
 background: var(--card-bg);  
 border-radius: var(--radius);  
 padding: 28px;  
 margin-bottom: 32px;  
 box-shadow: 0 2px 8px var(--shadow-light);  
 border: 1px solid var(--border-subtle);  
 backdrop-filter: blur(12px);  
}  
  
.section-header {  
 display: flex;  
 align-items: center;  
 justify-content: space-between;  
 margin-bottom: 24px;  
}  
  
.section-header h2 {  
 font-size: 20px;  
 font-weight: 600;  
 color: var(--text-main);  
 display: flex;  
 align-items: center;  
 gap: 8px;  
}  
  
.count-badge {  
 background: linear-gradient(135deg, var(--primary), var(--secondary));  
 color: white;  
 font-size: 12px;  
 font-weight: 600;  
 padding: 6px 12px;  
 border-radius: 16px;  
 min-width: 24px;  
 text-align: center;  
}  
  
/\* Friend Requests Grid \*/  
.requests-grid {  
 display: grid;  
 grid-template-columns: repeat(auto-fit, minmax(300px, 1fr));  
 gap: 16px;  
}  
  
.request-card {  
 background: rgba(255, 193, 7, 0.05);  
 border-radius: var(--radius);  
 padding: 20px;  
 box-shadow: 0 2px 8px var(--shadow-light);  
 border: 1px solid rgba(255, 193, 7, 0.2);  
 display: flex;  
 align-items: center;  
 gap: 16px;  
 transition: var(--transition);  
}  
  
.request-card:hover {  
 transform: translateY(-2px);  
 box-shadow: 0 4px 12px var(--shadow-light);  
}  
  
/\* Friends Grid \*/  
.friends-grid {  
 display: grid;  
 grid-template-columns: repeat(auto-fill, minmax(280px, 1fr));  
 gap: 20px;  
}  
  
.friend-card {  
 background: var(--card-bg);  
 border-radius: var(--radius);  
 padding: 24px;  
 box-shadow: 0 2px 8px var(--shadow-light);  
 border: 1px solid var(--border-subtle);  
 transition: var(--transition);  
 backdrop-filter: blur(12px);  
 position: relative;  
}  
  
.friend-card:hover {  
 transform: translateY(-2px);  
 box-shadow: 0 4px 12px var(--shadow-light);  
}  
  
.friend-card.online::before {  
 content: "";  
 position: absolute;  
 top: 20px;  
 right: 20px;  
 width: 12px;  
 height: 12px;  
 background: var(--success);  
 border-radius: 50%;  
 border: 2px solid white;  
 box-shadow: 0 0 8px rgba(16, 185, 129, 0.5);  
}  
  
/\* Friend Avatar \*/  
.friend-avatar {  
 width: 80px;  
 height: 80px;  
 border-radius: 50%;  
 margin: 0 auto 16px;  
 cursor: pointer;  
 border: 3px solid transparent;  
 background: linear-gradient(135deg, var(--primary), var(--secondary));  
 padding: 3px;  
 transition: var(--transition);  
}  
  
.friend-avatar:hover {  
 transform: scale(1.05);  
}  
  
.friend-avatar img {  
 width: 100%;  
 height: 100%;  
 border-radius: 50%;  
 object-fit: cover;  
}  
  
/\* Friend Info \*/  
.friend-info {  
 text-align: center;  
 margin-bottom: 20px;  
}  
  
.friend-name {  
 font-size: 18px;  
 font-weight: 600;  
 color: var(--text-main);  
 margin-bottom: 4px;  
}  
  
.friend-details {  
 color: var(--text-muted);  
 font-size: 14px;  
 margin-bottom: 8px;  
}  
  
.friend-sport {  
 background: rgba(30, 144, 255, 0.1);  
 color: var(--primary);  
 padding: 4px 12px;  
 border-radius: 16px;  
 font-size: 12px;  
 font-weight: 600;  
 display: inline-block;  
 margin-bottom: 8px;  
}  
  
.last-active {  
 color: var(--text-light);  
 font-size: 12px;  
}  
  
/\* Friend Actions \*/  
.friend-actions {  
 display: flex;  
 gap: 12px;  
}  
  
.action-btn {  
 flex: 1;  
 padding: 12px 16px;  
 border: none;  
 border-radius: var(--radius-small);  
 cursor: pointer;  
 font-weight: 600;  
 font-size: 13px;  
 transition: var(--transition);  
}  
  
.accept-btn {  
 background: linear-gradient(135deg, var(--secondary), #16a34a);  
 color: white;  
 box-shadow: 0 2px 8px rgba(34, 197, 94, 0.25);  
}  
  
.reject-btn {  
 background: rgba(239, 68, 68, 0.1);  
 color: #dc2626;  
 border: 1px solid rgba(239, 68, 68, 0.2);  
}  
  
.message-btn {  
 background: linear-gradient(135deg, var(--primary), var(--secondary));  
 color: white;  
 box-shadow: 0 2px 8px rgba(30, 144, 255, 0.25);  
}  
  
.remove-btn {  
 background: rgba(239, 68, 68, 0.1);  
 color: #dc2626;  
 border: 1px solid rgba(239, 68, 68, 0.2);  
}  
  
.play-btn {  
 background: linear-gradient(135deg, var(--warning), #f59e0b);  
 color: white;  
 box-shadow: 0 2px 8px rgba(245, 158, 11, 0.25);  
}  
  
.action-btn:hover {  
 transform: translateY(-1px);  
}  
  
.accept-btn:hover, .play-btn:hover {  
 box-shadow: 0 4px 12px rgba(34, 197, 94, 0.3);  
}  
  
.reject-btn:hover, .remove-btn:hover {  
 background: #dc2626;  
 color: white;  
 border-color: #dc2626;  
}  
  
.message-btn:hover {  
 box-shadow: 0 4px 12px rgba(30, 144, 255, 0.3);  
}  
  
/\* Modal \*/  
.modal {  
 display: none;  
 position: fixed;  
 z-index: 1000;  
 left: 0;  
 top: 0;  
 width: 100%;  
 height: 100%;  
 background-color: rgba(0, 0, 0, 0.4);  
 backdrop-filter: blur(8px);  
}  
  
.modal-content {  
 background: var(--card-bg);  
 margin: 5% auto;  
 padding: 32px;  
 border-radius: var(--radius);  
 width: 90%;  
 max-width: 500px;  
 position: relative;  
 box-shadow: 0 20px 60px var(--shadow-strong);  
 border: 1px solid var(--border-subtle);  
 backdrop-filter: blur(12px);  
}  
  
.close {  
 position: absolute;  
 top: 16px;  
 right: 20px;  
 font-size: 24px;  
 font-weight: 600;  
 cursor: pointer;  
 color: var(--text-light);  
 transition: var(--transition);  
}  
  
.close:hover {  
 color: var(--text-main);  
}  
  
/\* Empty States \*/  
.empty-state {  
 text-align: center;  
 padding: 40px 20px;  
 color: var(--text-light);  
}  
  
.empty-state h3 {  
 font-size: 18px;  
 margin-bottom: 8px;  
}  
  
.empty-state p {  
 font-size: 14px;  
}  
  
/\* Responsive Design \*/  
@media (max-width: 768px) {  
 .container {  
 padding: 16px;  
 }  
   
 .controls-section {  
 flex-direction: column;  
 align-items: stretch;  
 }  
   
 .search-bar {  
 min-width: unset;  
 }  
   
 .filter-buttons {  
 justify-content: center;  
 }  
   
 .header-stats {  
 gap: 16px;  
 }  
   
 .stat-card {  
 min-width: 80px;  
 padding: 16px;  
 }  
   
 .friends-grid, .requests-grid {  
 grid-template-columns: 1fr;  
 }  
   
 .section-card {  
 padding: 20px;  
 }  
   
 .friend-actions {  
 flex-direction: column;  
 }  
}

————————————————————————————————————————

## MyFriends\MyFriends.html

<!DOCTYPE html>  
<html lang="en">  
<head>  
 <meta charset="UTF-8" />  
 <meta name="viewport" content="width=device-width, initial-scale=1.0" />  
 <title>My Friends - Book&Play</title>  
 <link rel="stylesheet" href="MyFriends.css" />  
 <link href="https://fonts.googleapis.com/css2?family=Inter:wght@300;400;500;600;700&display=swap" rel="stylesheet" />  
</head>  
<body>  
  
 <div class="container">  
 <!-- Header Section -->  
 <div class="page-header">  
 <h1>My Friends</h1>  
 <div class="header-stats">  
 <div class="stat-card">  
 <span class="stat-number" id="totalFriends">0</span>  
 <span class="stat-label">Friends</span>  
 </div>  
 <div class="stat-card">  
 <span class="stat-number" id="totalRequests">0</span>  
 <span class="stat-label">Requests</span>  
 </div>  
 </div>  
 </div>  
  
 <!-- Search Section -->  
 <div class="controls-section">  
 <div class="search-bar">  
 <input type="text" id="friendSearch" placeholder="Search friends by name or sport..." />  
 <button class="search-btn">🔍</button>  
 </div>  
 </div>  
  
 <!-- Friend Requests Section -->  
 <div class="section-card requests-section">  
 <div class="section-header">  
 <h2>📩 Friend Requests</h2>  
 <span class="count-badge" id="requestsBadge">0</span>  
 </div>  
 <div id="friendRequests" class="requests-grid">  
 <!-- Friend requests populated by JS -->  
 </div>  
 </div>  
  
 <!-- Current Friends Section -->  
 <div class="section-card friends-section">  
 <div class="section-header">  
 <h2>👥 My Friends</h2>  
 <span class="count-badge" id="friendsBadge">0</span>  
 </div>  
 <div id="currentFriends" class="friends-grid">  
 <!-- Current friends populated by JS -->  
 </div>  
 </div>  
 </div>  
  
 <!-- Friend Profile Modal -->  
 <div id="friendModal" class="modal">  
 <div class="modal-content">  
 <span class="close">&times;</span>  
 <div id="friendModalBody">  
 <!-- Friend details populated by JS -->  
 </div>  
 </div>  
 </div>  
  
 <script src="MyFriends.js"></script>  
</body>  
</html>

————————————————————————————————————————

## MyFriends\MyFriends.js

// Cleaned MyFriends.js - Ready for DB integration  
  
let friendRequests = [];  
let currentFriends = [];  
  
const friendSearch = document.getElementById('friendSearch');  
const friendRequestsContainer = document.getElementById('friendRequests');  
const currentFriendsContainer = document.getElementById('currentFriends');  
const friendModal = document.getElementById('friendModal');  
const friendModalBody = document.getElementById('friendModalBody');  
const closeModal = document.querySelector('.close');  
  
document.addEventListener('DOMContentLoaded', function() {  
 console.log('My Friends page loaded');  
 updateStats();  
 renderFriendRequests();  
 renderCurrentFriends();  
 setupEventListeners();  
});  
  
function setupEventListeners() {  
 if (friendSearch) {  
 friendSearch.addEventListener('input', handleSearch);  
 }  
  
 if (closeModal) {  
 closeModal.addEventListener('click', () => {  
 friendModal.style.display = 'none';  
 });  
 }  
  
 window.addEventListener('click', (e) => {  
 if (e.target === friendModal) {  
 friendModal.style.display = 'none';  
 }  
 });  
}  
  
function updateStats() {  
 document.getElementById('totalFriends').textContent = currentFriends.length;  
 document.getElementById('totalRequests').textContent = friendRequests.length;  
 document.getElementById('requestsBadge').textContent = friendRequests.length;  
 document.getElementById('friendsBadge').textContent = currentFriends.length;  
}  
  
function renderFriendRequests() {  
 if (friendRequests.length === 0) {  
 friendRequestsContainer.innerHTML = `  
 <div class="empty-state">  
 <h3>No Friend Requests</h3>  
 <p>You're all caught up! No pending friend requests.</p>  
 </div>  
 `;  
 return;  
 }  
  
 friendRequestsContainer.innerHTML = ''; // Content will be dynamically injected later  
}  
  
function renderCurrentFriends() {  
 let filteredFriends = [...currentFriends];  
 const searchTerm = friendSearch ? friendSearch.value.toLowerCase().trim() : '';  
 if (searchTerm) {  
 filteredFriends = filteredFriends.filter(friend =>  
 friend.first\_name.toLowerCase().includes(searchTerm) ||  
 friend.last\_name.toLowerCase().includes(searchTerm) ||  
 friend.sport.toLowerCase().includes(searchTerm)  
 );  
 }  
  
 if (filteredFriends.length === 0) {  
 currentFriendsContainer.innerHTML = `  
 <div class="empty-state">  
 <h3>No Friends Found</h3>  
 <p>No friends match your search criteria.</p>  
 </div>  
 `;  
 return;  
 }  
  
 currentFriendsContainer.innerHTML = ''; // Content will be dynamically injected later  
}  
  
function handleSearch() {  
 renderCurrentFriends();  
}  
  
function acceptFriendRequest(requestId) {  
 // To be implemented with database logic  
}  
  
function rejectFriendRequest(requestId) {  
 // To be implemented with database logic  
}  
  
function messageFriend(friendId) {  
 // To be implemented with chat system  
}  
  
function showFriendModal(friendId) {  
 // To be implemented with friend profile data  
}  
  
function goToFindPlayers() {}  
function goToBookings() {}  
function goToMessages() {}  
function goToProfile() {}  
function logout() {}

————————————————————————————————————————

## MyFriends\MyFriends.php

<?php  
require\_once '../../../db.php';  
include '../navbar.php';  
include '../sportsScroll.html';  
include 'MyFriends.html';  
?>

————————————————————————————————————————

## VenueDetails\SubmitRating.php

<?php  
session\_start();  
require\_once '../../../db.php';  
  
header('Content-Type: application/json');  
  
// Add debugging  
error\_log("SubmitRating.php called");  
error\_log("Session user\_id: " . ($\_SESSION['user\_id'] ?? 'null'));  
  
$input = json\_decode(file\_get\_contents("php://input"), true);  
error\_log("Input data: " . json\_encode($input));  
  
$username = $\_SESSION['user\_id'] ?? null;  
$facilityId = $input['facility\_id'] ?? null;  
$rating = $input['rating'] ?? null;  
$comment = $input['comment'] ?? null;  
  
if (!$username || !$facilityId || !$rating || $comment === null) {  
 error\_log("Missing data - username: $username, facilityId: $facilityId, rating: $rating, comment: $comment");  
 echo json\_encode(['success' => false, 'message' => 'Missing data']);  
 exit;  
}  
  
$stmt = $conn->prepare("INSERT INTO ratings (facilities\_id, username, rating\_value, comment) VALUES (?, ?, ?, ?)");  
$stmt->bind\_param("isis", $facilityId, $username, $rating, $comment);  
  
if ($stmt->execute()) {  
 error\_log("Rating submitted successfully");  
 echo json\_encode(['success' => true, 'message' => 'Rating submitted']);  
} else {  
 error\_log("Failed to submit rating: " . $stmt->error);  
 echo json\_encode(['success' => false, 'message' => 'Failed to submit rating']);  
}  
?>

————————————————————————————————————————

## VenueDetails\SubmitReport.php

<?php  
session\_start();  
require\_once '../../../db.php';  
header('Content-Type: application/json');  
  
// Add debugging  
error\_log("=== SUBMITREPORT.PHP DEBUG START ===");  
error\_log("Request method: " . $\_SERVER['REQUEST\_METHOD']);  
error\_log("Session user\_id: " . ($\_SESSION['user\_id'] ?? 'null'));  
  
// Check if user is logged in  
if (!isset($\_SESSION['user\_id'])) {  
 error\_log("❌ User not logged in");  
 echo json\_encode(['success' => false, 'message' => 'User not logged in']);  
 exit;  
}  
  
// Check if it's a POST request  
if ($\_SERVER['REQUEST\_METHOD'] !== 'POST') {  
 error\_log("❌ Not a POST request");  
 echo json\_encode(['success' => false, 'message' => 'Only POST method allowed']);  
 exit;  
}  
  
// Get the JSON input  
$input = json\_decode(file\_get\_contents('php://input'), true);  
  
if (!$input) {  
 error\_log("❌ Invalid JSON input");  
 echo json\_encode(['success' => false, 'message' => 'Invalid JSON input']);  
 exit;  
}  
  
// Extract data  
$facility\_id = $input['facility\_id'] ?? null;  
$reason = $input['reason'] ?? '';  
$details = $input['details'] ?? '';  
$username = $\_SESSION['user\_id'];  
  
error\_log("Received data - facility\_id: $facility\_id, reason: $reason, details: $details, username: $username");  
  
// Validate required fields  
if (!$facility\_id || !$reason) {  
 error\_log("❌ Missing required fields");  
 echo json\_encode(['success' => false, 'message' => 'Missing required fields']);  
 exit;  
}  
  
// Validate facility exists  
$check\_stmt = $conn->prepare("SELECT facilities\_id FROM sportfacilities WHERE facilities\_id = ?");  
$check\_stmt->bind\_param("i", $facility\_id);  
$check\_stmt->execute();  
$result = $check\_stmt->get\_result();  
  
if ($result->num\_rows === 0) {  
 error\_log("❌ Venue not found for ID: $facility\_id");  
 echo json\_encode(['success' => false, 'message' => 'Venue not found']);  
 exit;  
}  
  
error\_log("✅ Venue found, proceeding with report insertion");  
  
// Insert the report into database  
$stmt = $conn->prepare("INSERT INTO reports (username, type, facilities\_id, Reason, message, created\_at) VALUES (?, 'report\_place', ?, ?, ?, NOW())");  
$stmt->bind\_param("siss", $username, $facility\_id, $reason, $details);  
  
if ($stmt->execute()) {  
 error\_log("✅ Report submitted successfully");  
 echo json\_encode(['success' => true, 'message' => 'Report submitted successfully']);  
} else {  
 error\_log("❌ Failed to submit report: " . $conn->error);  
 echo json\_encode(['success' => false, 'message' => 'Failed to submit report: ' . $conn->error]);  
}  
  
$stmt->close();  
error\_log("=== SUBMITREPORT.PHP DEBUG END ===");  
?>

————————————————————————————————————————

## VenueDetails\VenueAPI.php

<?php  
// VenueAPI.php - returns facility details + comments  
  
// Add debugging at the very beginning  
error\_log("=== VENUEAPI.PHP DEBUG START ===");  
error\_log("Script started at: " . date('Y-m-d H:i:s'));  
  
// Check if db.php exists and get its path  
$dbPath = '../../../db.php';  
$realDbPath = realpath($dbPath);  
error\_log("DB file path requested: " . $dbPath);  
error\_log("DB file real path: " . ($realDbPath ? $realDbPath : 'FILE NOT FOUND'));  
  
if (!$realDbPath) {  
 error\_log("❌ CRITICAL ERROR: db.php file not found!");  
 echo json\_encode(['success' => false, 'message' => 'Database file not found']);  
 exit;  
}  
  
require\_once $dbPath;  
header('Content-Type: application/json');  
  
// Add debugging  
error\_log("VenueAPI.php called with facility\_id: " . ($\_GET['facility\_id'] ?? 'null'));  
error\_log("Current working directory: " . getcwd());  
error\_log("DB file path: " . realpath('../../../db.php'));  
  
// Simple test endpoint  
if (isset($\_GET['test'])) {  
 echo json\_encode(['success' => true, 'message' => 'API is working', 'timestamp' => date('Y-m-d H:i:s')]);  
 exit;  
}  
  
// Check database connection  
error\_log("Checking database connection...");  
if ($conn->connect\_error) {  
 error\_log("❌ Database connection failed: " . $conn->connect\_error);  
 echo json\_encode(['success' => false, 'message' => 'Database connection failed: ' . $conn->connect\_error]);  
 exit;  
}  
  
error\_log("✅ Database connection successful");  
  
$facilityId = $\_GET['facility\_id'] ?? null;  
error\_log("Facility ID received: " . $facilityId);  
  
if (!$facilityId) {  
 error\_log("❌ No facility\_id provided");  
 echo json\_encode(['success' => false, 'message' => 'Facility ID is missing']);  
 exit;  
}  
  
error\_log("Processing facility\_id: " . $facilityId);  
  
// Test database query first  
error\_log("Testing basic database query...");  
$testQuery = "SELECT COUNT(\*) as count FROM sportfacilities";  
$testResult = $conn->query($testQuery);  
if (!$testResult) {  
 error\_log("❌ Basic database query failed: " . $conn->error);  
 echo json\_encode(['success' => false, 'message' => 'Database query failed: ' . $conn->error]);  
 exit;  
}  
$testRow = $testResult->fetch\_assoc();  
error\_log("✅ Database query test successful. Total facilities: " . $testRow['count']);  
  
// Fetch facility details with owner information  
error\_log("Preparing facility query...");  
$stmt = $conn->prepare("  
 SELECT sf.\*   
 FROM sportfacilities sf   
 WHERE sf.facilities\_id = ?  
");  
  
if (!$stmt) {  
 error\_log("❌ Prepare failed: " . $conn->error);  
 echo json\_encode(['success' => false, 'message' => 'Database prepare failed: ' . $conn->error]);  
 exit;  
}  
  
error\_log("✅ Prepare successful, binding parameters...");  
$stmt->bind\_param("i", $facilityId);  
error\_log("✅ Parameters bound, executing query...");  
$stmt->execute();  
$result = $stmt->get\_result();  
  
if (!$row = $result->fetch\_assoc()) {  
 error\_log("❌ No facility found for ID: " . $facilityId);  
 echo json\_encode(['success' => false, 'message' => 'Facility not found with ID: ' . $facilityId]);  
 exit;  
}  
  
error\_log("✅ Facility found: " . json\_encode($row));  
  
// Fetch comments for this facility  
error\_log("Fetching comments for facility\_id: " . $facilityId);  
$commentsStmt = $conn->prepare("SELECT username, rating\_value, comment FROM ratings WHERE facilities\_id = ?");  
if (!$commentsStmt) {  
 error\_log("❌ Comments prepare failed: " . $conn->error);  
 echo json\_encode(['success' => false, 'message' => 'Comments query failed: ' . $conn->error]);  
 exit;  
}  
  
$commentsStmt->bind\_param("i", $facilityId);  
$commentsStmt->execute();  
$commentsResult = $commentsStmt->get\_result();  
  
$comments = [];  
while ($commentRow = $commentsResult->fetch\_assoc()) {  
 $comments[] = $commentRow;  
}  
  
error\_log("✅ Comments fetched: " . count($comments) . " comments");  
  
// Calculate average rating  
error\_log("Calculating average rating...");  
$avgRatingStmt = $conn->prepare("SELECT AVG(rating\_value) as avg\_rating FROM ratings WHERE facilities\_id = ?");  
if (!$avgRatingStmt) {  
 error\_log("❌ Average rating prepare failed: " . $conn->error);  
 echo json\_encode(['success' => false, 'message' => 'Average rating query failed: ' . $conn->error]);  
 exit;  
}  
  
$avgRatingStmt->bind\_param("i", $facilityId);  
$avgRatingStmt->execute();  
$avgRatingResult = $avgRatingStmt->get\_result();  
$avgRow = $avgRatingResult->fetch\_assoc();  
$averageRating = round($avgRow['avg\_rating'], 1); // e.g., 3.7  
  
error\_log("✅ Average rating calculated: " . $averageRating);  
  
// Return combined JSON  
$response = [  
 'success' => true,  
 'facility' => $row,  
 'comments' => $comments,  
 'average\_rating' => $averageRating  
];  
  
error\_log("✅ Preparing final response...");  
error\_log("Response data: " . json\_encode($response));  
echo json\_encode($response);  
error\_log("=== VENUEAPI.PHP DEBUG END ===");  
  
?>

————————————————————————————————————————

## VenueDetails\VenueDetails.css

.container {  
 max-width: 1100px;  
 margin: 40px auto;  
 background: var(--card-bg);  
 border-radius: 16px;  
 padding: 24px;  
 box-shadow: 0 4px 10px var(--shadow-light);  
}  
  
/\* Align venue image & info \*/  
.venue-header {  
 display: flex;  
 gap: 24px;  
 flex-wrap: wrap;  
}  
  
.venue-image img {  
 width: 100%;  
 max-width: 500px;  
 height: 320px;  
 object-fit: cover;  
 border-radius: var(--radius);  
 box-shadow: 0 4px 12px rgba(0,0,0,0.1);  
}  
  
/\* Title row to align Report button beside the venue name \*/  
.venue-title-row {  
 display: flex;  
 align-items: center;  
 gap: 16px;  
 flex-wrap: wrap;  
}  
  
.venue-info h1 {  
 margin: 0;  
 font-size: 32px;  
}  
  
.info {  
 margin: 8px 0;  
 font-size: 16px;  
}  
  
.price {  
 font-weight: bold;  
 color: var(--secondary);  
 font-size: 20px;  
}  
  
.stars {  
 font-size: 26px;  
 color: #facc15;  
}  
  
.section-title {  
 margin-top: 40px;  
 font-size: 24px;  
 border-bottom: 2px solid var(--border-subtle);  
 padding-bottom: 10px;  
}  
  
/\* Comments \*/  
.comments {  
 margin-top: 20px;  
}  
  
.comment-card {  
 background: var(--bg-light);  
 border-radius: var(--radius);  
 padding: 16px;  
 margin-bottom: 16px;  
 box-shadow: 0 2px 5px rgba(0,0,0,0.1);  
}  
  
.user {  
 font-weight: bold;  
 margin-bottom: 8px;  
}  
  
.comment-stars {  
 margin-bottom: 8px;  
 color: #facc15;  
 font-size: 20px;  
}  
  
/\* Add Comment \*/  
.add-comment {  
 margin-top: 40px;  
}  
  
textarea {  
 width: 100%;  
 border-radius: var(--radius-small);  
 padding: 12px;  
 font-size: 16px;  
 border: 1px solid var(--card-border);  
 resize: none;  
}  
  
.submit-btn {  
 margin-top: 16px;  
 padding: 12px 24px;  
 border: none;  
 background: linear-gradient(135deg, var(--primary), var(--secondary));  
 color: #fff;  
 font-weight: 600;  
 border-radius: var(--radius-small);  
 cursor: pointer;  
 transition: var(--transition);  
}  
  
.submit-btn:hover {  
 transform: translateY(-1px);  
 box-shadow: 0 4px 12px rgba(30, 144, 255, 0.3);  
}  
  
.rating-stars {  
 display: flex;  
 gap: 8px;  
 margin-top: 10px;  
 font-size: 30px;  
 cursor: pointer;  
}  
  
.rating-stars i {  
 color: #e5e7eb;  
 transition: color 0.3s;  
}  
  
.rating-stars i.filled {  
 color: #facc15;  
}  
  
/\* Back button \*/  
.back-button {  
 margin-bottom: 20px;  
}  
  
.back-button a {  
 display: inline-block;  
 padding: 10px 20px;  
 background: linear-gradient(135deg, var(--primary), var(--secondary));  
 color: white;  
 border-radius: var(--radius-small);  
 text-decoration: none;  
 font-weight: 600;  
 box-shadow: 0 2px 8px var(--shadow-light);  
 transition: var(--transition);  
}  
  
.back-button a:hover {  
 transform: translateY(-2px);  
 box-shadow: 0 4px 16px var(--shadow-medium);  
}  
  
/\* View more comments button \*/  
.view-more-btn {  
 padding: 10px 20px;  
 margin: 20px auto;  
 display: block;  
 background: linear-gradient(to right, #1e90ff, #22c55e);  
 color: white;  
 border: none;  
 border-radius: 8px;  
 font-weight: 600;  
 cursor: pointer;  
 transition: 0.3s;  
}  
  
.view-more-btn:hover {  
 background: linear-gradient(to right, #22c55e, #1e90ff);  
}  
  
/\* Report Button \*/  
.report-btn {  
 background: #f87171;  
 color: white;  
 border: none;  
 padding: 10px 20px;  
 border-radius: 8px;  
 font-weight: 600;  
 font-size: 14px;  
 cursor: pointer;  
 transition: all 0.3s;  
}  
  
.report-btn:hover {  
 background: #dc2626;  
}  
  
/\* Modal Styling \*/  
.modal {  
 display: none;  
 position: fixed;  
 z-index: 9999;  
 left: 0; top: 0; right: 0; bottom: 0;  
 background-color: rgba(0,0,0,0.4);  
 justify-content: center;  
 align-items: center;  
}  
  
.modal-content {  
 background: #fff;  
 border-radius: 12px;  
 padding: 20px;  
 width: 400px;  
 box-shadow: 0 4px 12px rgba(0,0,0,0.3);  
}  
  
.modal-content h3 {  
 margin-bottom: 15px;  
 font-size: 20px;  
 font-weight: 600;  
}  
  
.modal-content select, .modal-content textarea {  
 width: 100%;  
 padding: 10px;  
 border: 1px solid #ddd;  
 border-radius: 8px;  
 margin-bottom: 15px;  
 font-size: 14px;  
}  
  
.modal-buttons {  
 display: flex;  
 justify-content: flex-end;  
 gap: 10px;  
}  
  
.modal-buttons button {  
 padding: 8px 16px;  
 border-radius: 8px;  
 font-weight: 600;  
 cursor: pointer;  
 border: none;  
 transition: all 0.3s;  
}  
  
.modal-buttons button:first-child {  
 background: #1e90ff;  
 color: white;  
}  
  
.modal-buttons button:last-child {  
 background: #e5e7eb;  
}  
.char-counter {  
 font-size: 12px;  
 color: var(--text-muted);  
 text-align: right;  
 margin-top: -10px;  
 margin-bottom: 10px;  
}

————————————————————————————————————————

## VenueDetails\VenueDetails.html

<!DOCTYPE html>  
<html lang="en">  
<head>  
 <meta charset="UTF-8">  
 <title>Venue Details - Book&Play</title>  
 <link href="https://fonts.googleapis.com/css2?family=Inter:wght@300;400;500;600;700&display=swap" rel="stylesheet">  
 <link rel="stylesheet" href="../global.css">  
 <link rel="stylesheet" href="VenueDetails.css">  
</head>  
  
<body>  
 <div class="container">  
  
 <!-- Back Button -->  
 <div class="back-button">  
 <a href="../BookVenue/BookVenue.php">← Back to Venues</a>  
 </div>  
  
 <!-- Venue Header -->  
 <div class="venue-header-wrapper">  
 <div class="venue-header">  
 <div class="venue-image">  
 <img id="venue-image" src="" alt="Venue Image">  
 </div>  
  
 <div class="venue-info">  
 <div class="venue-title-row">  
 <h1 id="venue-name">Loading...</h1>  
 <button class="report-btn" onclick="openReportModal()">Report Venue</button>  
 </div>  
  
 <div class="info" id="venue-location"></div>  
 <div class="info" id="venue-sport"></div>  
 <div class="info" id="venue-owner"></div>  
 <div class="info price" id="venue-price"></div>  
 <div class="stars" id="venue-stars">★★★★☆</div>  
 </div>  
 </div>  
 </div>  
  
 <!-- Description -->  
 <h2 class="section-title">Description</h2>  
 <p id="venue-description">Loading description...</p>  
  
 <!-- Reviews -->  
 <h2 class="section-title">Reviews</h2>  
 <div class="comments" id="commentsContainer">  
 <!-- Comments will be dynamically inserted here -->  
 </div>  
  
 <!-- View More / View Less Buttons -->  
 <button id="view-more-btn" class="view-more-btn" style="display: none;">View More Comments</button>  
 <button id="view-less-btn" class="view-more-btn" style="display: none;">View Less Comments</button>  
  
 <!-- Add Review -->  
 <div class="add-comment">  
 <h3>Add your review:</h3>  
 <textarea rows="3" placeholder="Write your comment..."></textarea>  
 <div class="rating-stars" id="rating">  
 <i data-value="1">&#9733;</i>  
 <i data-value="2">&#9733;</i>  
 <i data-value="3">&#9733;</i>  
 <i data-value="4">&#9733;</i>  
 <i data-value="5">&#9733;</i>  
 </div>  
 <button class="submit-btn">Submit Review</button>  
 </div>  
 </div>  
  
 <!-- Report Modal -->  
 <div id="reportModal" class="modal">  
 <div class="modal-content">  
 <h3>Report Venue</h3>  
 <label for="reason">Reason:</label>  
 <select id="reason">  
 <option value="" disabled selected>Select a reason</option>  
 <option>Inaccurate Information</option>  
 <option>Offensive Content</option>  
 <option>Duplicate Listing</option>  
 <option>Wrong Owner</option>  
 <option>Fake Images</option>  
 <option>Other</option>  
 </select>  
 <label for="details">Additional Details:</label>  
 <textarea id="details" rows="3" placeholder="Write details..." maxlength="100"></textarea>  
 <div class="char-counter" id="charCounter">0 / 100</div>  
 <div class="modal-buttons">  
 <button onclick="submitReport()">Submit</button>  
 <button onclick="closeReportModal()">Cancel</button>  
 </div>  
 </div>  
 </div>  
  
 <script src="VenueDetails.js"></script>  
</body>  
</html>

————————————————————————————————————————

## VenueDetails\VenueDetails.js

// Make currentRating global  
let currentRating = 0;  
  
document.addEventListener('DOMContentLoaded', () => {  
 console.log('🚀 VenueDetails.js DOMContentLoaded triggered');  
   
 const params = new URLSearchParams(window.location.search);  
 const facilityId = params.get('facility\_id');  
  
 console.log('🔍 VenueDetails loading for facility\_id:', facilityId);  
 console.log('📍 Current URL:', window.location.href);  
 console.log('🔗 All URL params:', Object.fromEntries(params));  
  
 if (!facilityId) {  
 console.error('❌ No facility\_id found in URL');  
 alert("Facility ID not found in URL.");  
 return;  
 }  
  
 console.log('📡 Making API request to VenueAPI.php...');  
 fetch(`VenueAPI.php?facility\_id=${facilityId}`)  
 .then(res => {  
 console.log('📡 API Response status:', res.status);  
 console.log('📡 API Response headers:', res.headers);  
 if (!res.ok) {  
 throw new Error(`HTTP error! status: ${res.status}`);  
 }  
 return res.json();  
 })  
 .then(data => {  
 console.log('📊 API Response data:', data);  
   
 if (!data.success) {  
 console.error('❌ API returned error:', data.message);  
 alert("Error loading facility: " + data.message);  
 return;  
 }  
  
 console.log('✅ API call successful, populating data...');  
  
 // Show average rating in stars  
 const starsContainer = document.getElementById('venue-stars');  
 const average = parseFloat(data.average\_rating || 0);  
  
 const fullStars = Math.floor(average);  
 const halfStar = average - fullStars >= 0.5;  
 let starsHTML = '';  
  
 for (let i = 1; i <= 5; i++) {  
 if (i <= fullStars) {  
 starsHTML += '★'; // full star  
 } else {  
 starsHTML += '☆'; // empty star  
 }  
 }  
 starsContainer.textContent = starsHTML;  
  
 const facility = data.facility;  
 console.log('🏟️ Facility data:', facility);  
  
 // Populate venue info  
 const imageElement = document.getElementById('venue-image');  
 const imagePath = `../../../uploads/venues/${facility.image\_url}`;  
 console.log('🖼️ Setting image path:', imagePath);  
 imageElement.src = imagePath;  
   
 document.getElementById('venue-name').textContent = facility.place\_name;  
 document.getElementById('venue-location').textContent = `📍 Location: ${facility.location}`;  
 document.getElementById('venue-sport').textContent = `⚽ Sport: ${facility.SportCategory}`;  
 document.getElementById('venue-owner').textContent = `👤 Owner: ${facility.owner\_username || 'Not specified'}`;  
 document.getElementById('venue-price').textContent = `₪${facility.price} / hour`;  
 document.getElementById('venue-description').textContent = facility.description || 'No description available.';  
  
 // Populate comments  
 const commentsContainer = document.getElementById('commentsContainer');  
 const viewMoreBtn = document.getElementById('view-more-btn');  
 const viewLessBtn = document.getElementById('view-less-btn');  
 commentsContainer.innerHTML = '';  
  
 const allComments = [];  
  
 data.comments.forEach((comment, index) => {  
 const card = document.createElement('div');  
 card.className = 'comment-card';  
 card.style.backgroundColor = '#e0f7ff';  
  
 const user = document.createElement('div');  
 user.className = 'user';  
 user.textContent = comment.username;  
  
 const stars = document.createElement('div');  
 stars.className = 'comment-stars';  
 stars.textContent = '★'.repeat(comment.rating\_value) + '☆'.repeat(5 - comment.rating\_value);  
  
 const text = document.createElement('div');  
 text.className = 'text';  
 text.textContent = comment.comment;  
  
 card.appendChild(user);  
 card.appendChild(stars);  
 card.appendChild(text);  
 card.style.display = index < 3 ? 'block' : 'none';  
  
 commentsContainer.appendChild(card);  
 allComments.push(card);  
 });  
  
 if (allComments.length > 3) {  
 viewMoreBtn.style.display = 'block';  
 viewLessBtn.style.display = 'none';  
  
 viewMoreBtn.onclick = () => {  
 allComments.forEach(card => card.style.display = 'block');  
 viewMoreBtn.style.display = 'none';  
 viewLessBtn.style.display = 'block';  
 };  
  
 viewLessBtn.onclick = () => {  
 allComments.forEach((card, index) => {  
 card.style.display = index < 3 ? 'block' : 'none';  
 });  
 viewMoreBtn.style.display = 'block';  
 viewLessBtn.style.display = 'none';  
 };  
 } else {  
 viewMoreBtn.style.display = 'none';  
 viewLessBtn.style.display = 'none';  
 }  
 })  
 .catch(err => {  
 console.error("Fetch error:", err);  
 alert("Failed to fetch facility data.");  
 });  
  
 // Star Rating UI  
 const stars = document.querySelectorAll('#rating i');  
 stars.forEach(star => {  
 star.addEventListener('mouseover', () => {  
 resetStars();  
 highlightStars(star.dataset.value);  
 });  
 star.addEventListener('mouseout', () => {  
 resetStars();  
 highlightStars(currentRating);  
 });  
 star.addEventListener('click', () => {  
 currentRating = parseInt(star.dataset.value); // Store globally  
 highlightStars(currentRating);  
 });  
 });  
  
 function highlightStars(rating) {  
 stars.forEach(star => {  
 if (parseInt(star.dataset.value) <= rating) {  
 star.classList.add('filled');  
 }  
 });  
 }  
  
 function resetStars() {  
 stars.forEach(star => {  
 star.classList.remove('filled');  
 });  
 }  
  
 // Character counter  
 const detailsInput = document.getElementById("details");  
 const charCounter = document.getElementById("charCounter");  
 detailsInput?.addEventListener('input', () => {  
 charCounter.textContent = `${detailsInput.value.length} / 100`;  
 });  
});  
  
// Report modal handling  
function openReportModal() {  
 document.getElementById("reportModal").style.display = "flex";  
}  
function closeReportModal() {  
 document.getElementById("reportModal").style.display = "none";  
}  
function submitReport() {  
 const reason = document.getElementById("reason").value;  
 const details = document.getElementById("details").value;  
   
 if (!reason) {  
 alert("Please select a reason.");  
 return;  
 }  
  
 // Get facility ID from URL  
 const params = new URLSearchParams(window.location.search);  
 const facilityId = params.get('facility\_id');  
   
 if (!facilityId) {  
 alert("Venue ID not found. Please try again.");  
 return;  
 }  
  
 // Send report to backend  
 fetch('SubmitReport.php', {  
 method: 'POST',  
 headers: { 'Content-Type': 'application/json' },  
 body: JSON.stringify({   
 facility\_id: facilityId,   
 reason: reason,   
 details: details   
 })  
 })  
 .then(res => res.json())  
 .then(data => {  
 if (data.success) {  
 alert("Thank you for your report! It has been submitted successfully.");  
 closeReportModal();  
 // Clear the form  
 document.getElementById("reason").value = "";  
 document.getElementById("details").value = "";  
 document.getElementById("charCounter").textContent = "0 / 100";  
 } else {  
 alert("Failed to submit report: " + data.message);  
 }  
 })  
 .catch(err => {  
 console.error("Report submission error:", err);  
 alert("An error occurred while submitting your report. Please try again.");  
 });  
}  
  
// Submit review  
document.querySelector('.submit-btn').addEventListener('click', () => {  
 const params = new URLSearchParams(window.location.search);  
 const facilityId = params.get('facility\_id');  
 const rating = parseInt(currentRating);  
 const comment = document.querySelector('.add-comment textarea').value.trim();  
  
 if (!rating || comment === "") {  
 alert("Please provide both rating and comment.");  
 return;  
 }  
  
 fetch('SubmitRating.php', {  
 method: 'POST',  
 headers: { 'Content-Type': 'application/json' },  
 body: JSON.stringify({ facility\_id: facilityId, rating, comment })  
 })  
 .then(res => res.json())  
 .then(data => {  
 if (data.success) {  
 alert("Thank you for your feedback!");  
 location.reload(); // Optional: reload to see new comment  
 } else {  
 alert("Failed to submit: " + data.message);  
 }  
 })  
 .catch(err => {  
 console.error("Submit error:", err);  
 alert("An error occurred while submitting your rating.");  
 });  
});

————————————————————————————————————————

## VenueDetails\VenueDetails.php

<?php  
session\_start();  
require\_once '../../../db.php';  
$\_SESSION['previous\_page'] = $\_SERVER['PHP\_SELF'];  
  
// ✅ 1. Check request to fetch user image  
if (isset($\_GET['action']) && $\_GET['action'] === 'get\_user\_image') {  
 $username = $\_SESSION['user\_id'] ?? '';  
  
 if (!$username) {  
 echo json\_encode(['error' => 'Not logged in']);  
 exit;  
 }  
}  
  
include 'VenueDetails.html';  
exit;  
?>

————————————————————————————————————————

## global.css

/\* ===================== CSS VARIABLES ===================== \*/  
:root {  
 /\* Primary Colors \*/  
 --primary: #1e90ff;  
 --primary-alt: #00bcd4;  
 --secondary: #22c55e;  
 --secondary-alt: #4caf50;  
 --accent: #f59e0b;  
 --danger: #ef4444;  
 --danger-alt: #f44336;  
   
 /\* Gradients \*/  
 --gradient-primary: linear-gradient(to right, #4fc3f7, #81c784);  
 --gradient-secondary: linear-gradient(to right, #00bcd4, #4caf50);  
 --gradient-danger: linear-gradient(to right, #f44336, #ff7961);  
 --gradient-success: linear-gradient(to right, #4caf50, #81c784);  
 --gradient-bg: linear-gradient(to bottom right, #e0f7fa, #d0f8ce);  
 --gradient-navbar: linear-gradient(to right, #e0f7fa, #d0f8ce);  
   
 /\* Text Colors \*/  
 --text-main: #0f172a;  
 --text-secondary: #1e293b;  
 --text-muted: #6b7280;  
 --text-light: #9ca3af;  
   
 /\* Background Colors \*/  
 --background: #f8fafc;  
 --light-bg: #f8f6f0;  
 --card-bg: rgba(255, 255, 255, 0.85);  
 --card-bg-alt: rgba(255, 255, 255, 0.8);  
 --card-bg-blur: rgba(255, 255, 255, 0.6);  
   
 /\* Borders & Shadows \*/  
 --card-border: #d1d5db;  
 --border-subtle: rgba(0, 0, 0, 0.08);  
 --radius: 12px;  
 --radius-lg: 16px;  
 --radius-xl: 20px;  
 --shadow: 0 4px 20px rgba(0, 0, 0, 0.08);  
 --shadow-md: 0 6px 20px rgba(30, 144, 255, 0.1);  
 --shadow-lg: 0 0 30px rgba(0, 128, 128, 0.2);  
 --shadow-xl: 0 0 40px rgba(0, 150, 136, 0.25);  
   
 /\* Layout \*/  
 --container-max-width: 1200px;  
 --navbar-height: 80px;  
   
 /\* Transitions \*/  
 --transition: all 0.3s ease;  
 --transition-fast: all 0.2s ease;  
 }  
   
 /\* ===================== GLOBAL RESET & BASE STYLES ===================== \*/  
 \* {  
 box-sizing: border-box;  
 margin: 0;  
 padding: 0;  
 }  
   
 body {  
 font-family: 'Outfit', -apple-system, BlinkMacSystemFont, 'Segoe UI', Roboto, sans-serif;  
 background: var(--gradient-bg);  
 color: var(--text-main);  
 line-height: 1.6;  
 margin: 0;  
 padding: 0;  
 min-height: 100vh;  
 overflow-x: hidden;  
 }  
   
 /\* ===================== TYPOGRAPHY ===================== \*/  
 h1, h2, h3, h4, h5, h6 {  
 font-weight: 600;  
 line-height: 1.3;  
 margin-bottom: 0.5em;  
 }  
   
 h1 { font-size: 2.5rem; }  
 h2 {   
 font-size: 2rem;  
 text-align: center;  
 background: var(--gradient-secondary);  
 -webkit-background-clip: text;  
 -webkit-text-fill-color: transparent;  
 background-clip: text;  
 }  
 h3 { font-size: 1.75rem; }  
 h4 { font-size: 1.5rem; }  
 h5 { font-size: 1.25rem; }  
 h6 { font-size: 1rem; }  
   
 p {  
 margin-bottom: 1rem;  
 }  
   
 /\* ===================== NAVBAR STYLES ===================== \*/  
 .admin-navbar,  
 .facility-owner-navbar,  
 .fresh-navbar {  
 background: var(--gradient-navbar);  
 box-shadow: var(--shadow);  
 padding: 16px 0;  
 position: sticky;  
 top: 0;  
 z-index: 1000;  
 width: 100%;  
 }  
   
 .nav-container {  
 max-width: var(--container-max-width);  
 margin: auto;  
 padding: 0 20px;  
 display: flex;  
 justify-content: space-between;  
 align-items: center;  
 flex-wrap: wrap;  
 gap: 15px;  
 }  
   
 .logo {  
 font-size: 24px;  
 font-weight: bold;  
 background: var(--gradient-secondary);  
 -webkit-background-clip: text;  
 -webkit-text-fill-color: transparent;  
 background-clip: text;  
 margin: 0;  
 }  
   
 .nav-links {  
 list-style: none;  
 display: flex;  
 gap: 15px;  
 margin: 0;  
 padding: 0;  
 flex-wrap: wrap;  
 }  
   
 .nav-links li a {  
 text-decoration: none;  
 font-weight: 500;  
 font-size: 15px;  
 padding: 10px 18px;  
 border-radius: var(--radius);  
 transition: var(--transition);  
 color: var(--text-main);  
 background: var(--card-bg-blur);  
 backdrop-filter: blur(6px);  
 display: inline-block;  
 text-align: center;  
 }  
   
 .nav-links li a:hover,  
 .nav-links li a.active {  
 background: var(--gradient-primary);  
 color: white;  
 transform: translateY(-1px);  
 }  
   
 .nav-links li a.logout {  
 background: rgba(244, 67, 54, 0.15);  
 color: var(--danger-alt);  
 font-weight: 600;  
 }  
   
 .nav-links li a.logout:hover {  
 background: var(--danger-alt);  
 color: white;  
 }  
   
 /\* Mobile Menu Toggle \*/  
 .menu-toggle {  
 display: none;  
 flex-direction: column;  
 justify-content: space-between;  
 width: 30px;  
 height: 21px;  
 cursor: pointer;  
 }  
   
 .menu-toggle span {  
 display: block;  
 height: 3px;  
 width: 100%;  
 background-color: var(--text-main);  
 border-radius: 3px;  
 transition: var(--transition);  
 }  
   
 /\* ===================== CARD STYLES ===================== \*/  
 .card,  
 .fresh-card,  
 .venue-card {  
 background: var(--card-bg);  
 border-radius: var(--radius-xl);  
 box-shadow: var(--shadow-xl);  
 padding: 25px;  
 margin: 20px auto;  
 backdrop-filter: blur(10px);  
 border: 1px solid var(--card-border);  
 transition: var(--transition);  
 }  
   
 .fresh-card {  
 width: 95%;  
 max-width: 1000px;  
 animation: fadeInUp 0.4s ease;  
 }  
   
 .venue-card {  
 max-width: 700px;  
 width: 95%;  
 margin-top: 60px;  
 }  
   
 .card:hover,  
 .fresh-card:hover,  
 .venue-card:hover {  
 transform: translateY(-2px);  
 box-shadow: 0 8px 32px rgba(0, 150, 136, 0.3);  
 }  
   
 /\* ===================== BUTTON STYLES ===================== \*/  
 .btn {  
 padding: 10px 20px;  
 border: none;  
 border-radius: var(--radius);  
 font-weight: bold;  
 font-size: 14px;  
 cursor: pointer;  
 transition: var(--transition);  
 display: inline-block;  
 text-align: center;  
 text-decoration: none;  
 min-width: 100px;  
 }  
   
 .btn:hover {  
 transform: translateY(-2px) scale(1.05);  
 }  
   
 .btn.primary,  
 .btn-view {  
 background: var(--gradient-primary);  
 color: white;  
 box-shadow: 0 0 10px rgba(79, 195, 247, 0.5);  
 }  
   
 .btn.secondary,  
 .btn.green,  
 .btn.accept,  
 .btn-resolve {  
 background: var(--gradient-success);  
 color: white;  
 box-shadow: 0 0 10px rgba(76, 175, 80, 0.5);  
 }  
   
 .btn.blue {  
 background: var(--primary);  
 color: white;  
 }  
   
 .btn.danger,  
 .btn.delete,  
 .btn-remove {  
 background: var(--gradient-danger);  
 color: white;  
 box-shadow: 0 0 10px rgba(244, 67, 54, 0.5);  
 }  
   
 .btn-back {  
 background: var(--gradient-primary);  
 color: white;  
 font-size: 20px;  
 padding: 16px 36px;  
 border-radius: var(--radius-lg);  
 font-weight: 600;  
 box-shadow: 0 0 20px rgba(79, 195, 247, 0.4);  
 }  
   
 .btn-back:hover {  
 transform: translateY(-3px) scale(1.04);  
 box-shadow: 0 0 25px rgba(129, 199, 132, 0.5);  
 }  
   
 /\* ===================== TABLE STYLES ===================== \*/  
 table,  
 .fresh-table {  
 width: 100%;  
 border-collapse: collapse;  
 margin-top: 25px;  
 border: 1px solid #e2e8f0;  
 }  
   
 .table-container {  
 width: 100%;  
 overflow-x: auto;  
 -webkit-overflow-scrolling: touch;  
 }  
   
 th, td,  
 .fresh-table th,  
 .fresh-table td {  
 padding: 14px;  
 text-align: center;  
 border: 1px solid #e2e8f0;  
 background-color: var(--card-bg);  
 }  
   
 th {  
 background-color: #f1f5f9;  
 font-weight: 600;  
 }  
   
 /\* ===================== FORM STYLES ===================== \*/  
 input, textarea, select {  
 padding: 12px;  
 font-size: 15px;  
 border: 1px solid #cbd5e1;  
 border-radius: 8px;  
 background-color: #f8fafc;  
 color: var(--text-main);  
 transition: var(--transition);  
 width: 100%;  
 max-width: 100%;  
 }  
   
 input:focus, textarea:focus, select:focus {  
 outline: none;  
 border-color: var(--primary);  
 box-shadow: 0 0 0 3px rgba(30, 144, 255, 0.1);  
 }  
   
 .add-sport-form {  
 display: flex;  
 gap: 15px;  
 justify-content: center;  
 margin-top: 20px;  
 flex-wrap: wrap;  
 }  
   
 .add-sport-form input {  
 width: 250px;  
 max-width: 100%;  
 }  
   
 /\* ===================== MODAL STYLES ===================== \*/  
 .modal {  
 display: none;  
 position: fixed;  
 z-index: 2000;  
 left: 0;  
 top: 0;  
 width: 100%;  
 height: 100%;  
 overflow: auto;  
 background-color: rgba(0, 0, 0, 0.5);  
 backdrop-filter: blur(4px);  
 align-items: center;  
 justify-content: center;  
 }  
   
 .modal-content {  
 background: white;  
 margin: auto;  
 padding: 30px;  
 border-radius: var(--radius-xl);  
 width: 80%;  
 max-width: 700px;  
 max-height: 90vh;  
 overflow-y: auto;  
 box-shadow: 0 0 30px rgba(0, 0, 0, 0.3);  
 position: relative;  
 animation: modalFadeIn 0.3s ease;  
 }  
   
 @keyframes modalFadeIn {  
 from { opacity: 0; transform: translateY(-20px); }  
 to { opacity: 1; transform: translateY(0); }  
 }  
   
 .close,  
 .close-btn {  
 position: absolute;  
 right: 20px;  
 top: 20px;  
 font-size: 24px;  
 font-weight: bold;  
 cursor: pointer;  
 color: #999;  
 height: 30px;  
 width: 30px;  
 display: flex;  
 align-items: center;  
 justify-content: center;  
 background: #f1f1f1;  
 border-radius: 50%;  
 transition: var(--transition);  
 }  
   
 .close:hover,  
 .close-btn:hover {  
 color: #333;  
 background: #e0e0e0;  
 transform: scale(1.1);  
 }  
   
 /\* Modal Images \*/  
 .modal-content .images {  
 display: flex;  
 flex-wrap: wrap;  
 gap: 10px;  
 margin-bottom: 20px;  
 justify-content: center;  
 }  
   
 .modal-content .images img {  
 width: 120px;  
 height: 80px;  
 object-fit: cover;  
 border-radius: var(--radius);  
 }  
   
 /\* ===================== UTILITY CLASSES ===================== \*/  
 .container {  
 max-width: var(--container-max-width);  
 margin: 0 auto;  
 padding: 0 20px;  
 }  
   
 .text-center { text-align: center; }  
 .text-left { text-align: left; }  
 .text-right { text-align: right; }  
   
 .hidden { display: none; }  
 .visible { display: block; }  
   
 .mt-1 { margin-top: 0.25rem; }  
 .mt-2 { margin-top: 0.5rem; }  
 .mt-3 { margin-top: 1rem; }  
 .mt-4 { margin-top: 1.5rem; }  
 .mt-5 { margin-top: 3rem; }  
   
 .mb-1 { margin-bottom: 0.25rem; }  
 .mb-2 { margin-bottom: 0.5rem; }  
 .mb-3 { margin-bottom: 1rem; }  
 .mb-4 { margin-bottom: 1.5rem; }  
 .mb-5 { margin-bottom: 3rem; }  
   
 /\* ===================== MESSAGE STYLES ===================== \*/  
 .message-box {  
 margin: 10px 0;  
 padding: 10px;  
 font-weight: bold;  
 border-radius: 8px;  
 display: none;  
 width: 100%;  
 text-align: center;  
 }  
   
 .message-success {  
 color: green;  
 background-color: #e3f9e5;  
 border: 1px solid #8bd48b;  
 }  
   
 .message-error,  
 .error-message {  
 color: var(--danger-alt);  
 background-color: #ffebee;  
 border: 1px solid #ffcdd2;  
 font-weight: 500;  
 text-align: center;  
 padding: 15px;  
 border-radius: var(--radius);  
 }  
   
 /\* ===================== ANIMATIONS ===================== \*/  
 @keyframes fadeInUp {  
 from { opacity: 0; transform: translateY(20px); }  
 to { opacity: 1; transform: translateY(0); }  
 }  
   
 @keyframes spin {  
 0% { transform: rotate(0deg); }  
 100% { transform: rotate(360deg); }  
 }  
   
 /\* ===================== RESPONSIVE DESIGN ===================== \*/  
   
 /\* Large devices (desktops, less than 1200px) \*/  
 @media screen and (max-width: 1200px) {  
 .fresh-card, .venue-card {  
 padding: 25px;  
 width: 95%;  
 }  
 }  
   
 /\* Medium devices (tablets, less than 992px) \*/  
 @media screen and (max-width: 992px) {  
 .nav-container {  
 padding: 0 15px;  
 }  
   
 .nav-links li a {  
 padding: 8px 14px;  
 font-size: 14px;  
 }  
   
 .logo {  
 font-size: 22px;  
 }  
   
 .fresh-card, .venue-card {  
 padding: 20px;  
 }  
   
 .fresh-table th, .fresh-table td,  
 th, td {  
 padding: 12px 10px;  
 font-size: 14px;  
 }  
   
 .btn {  
 padding: 10px 15px;  
 font-size: 13px;  
 }  
 }  
   
 /\* Small devices (landscape phones, less than 768px) \*/  
 @media screen and (max-width: 768px) {  
 body {  
 padding: 15px;  
 }  
   
 .nav-container {  
 justify-content: center;  
 flex-direction: column;  
 }  
   
 .nav-links {  
 width: 100%;  
 justify-content: center;  
 margin-top: 15px;  
 }  
   
 .fresh-card, .venue-card {  
 padding: 20px 15px;  
 width: 100%;  
 margin-top: 40px;  
 }  
   
 h2 {  
 font-size: 24px;  
 }  
   
 .btn-back {  
 font-size: 18px;  
 padding: 14px 30px;  
 }  
   
 .add-sport-form {  
 flex-direction: column;  
 align-items: center;  
 }  
   
 .add-sport-form input {  
 width: 100%;  
 max-width: 300px;  
 }  
 }  
   
 /\* Extra small devices (portrait phones, less than 576px) \*/  
 @media screen and (max-width: 576px) {  
 body {  
 padding: 10px;  
 }  
   
 /\* Mobile navbar \*/  
 .menu-toggle {  
 display: flex;  
 position: absolute;  
 right: 20px;  
 top: 20px;  
 }  
   
 .nav-container {  
 flex-direction: column;  
 align-items: center;  
 padding-top: 15px;  
 }  
   
 .nav-links {  
 flex-direction: column;  
 width: 100%;  
 gap: 8px;  
 max-height: 0;  
 overflow: hidden;  
 transition: max-height 0.5s ease;  
 }  
   
 .nav-links.active {  
 max-height: 500px;  
 }  
   
 .nav-links li {  
 width: 100%;  
 }  
   
 .nav-links li a {  
 text-align: center;  
 width: 100%;  
 }  
   
 .fresh-card, .venue-card {  
 padding: 15px 10px;  
 margin-top: 30px;  
 }  
   
 h2 {  
 font-size: 20px;  
 }  
   
 .btn {  
 width: 100%;  
 max-width: 300px;  
 }  
   
 .btn-back {  
 font-size: 16px;  
 padding: 12px 25px;  
 width: 80%;  
 max-width: 250px;  
 }  
   
 .fresh-table th, .fresh-table td,  
 th, td {  
 padding: 8px 5px;  
 font-size: 12px;  
 }  
   
 .modal-content {  
 padding: 15px;  
 width: 95%;  
 }  
 }  
   
 /\* ===================== PRINT STYLES ===================== \*/  
 @media print {  
 .nav-links,  
 .btn,  
 .modal,  
 .menu-toggle {  
 display: none !important;  
 }  
   
 body {  
 padding: 0;  
 background: white;  
 }  
   
 .card, .fresh-card, .venue-card {  
 box-shadow: none;  
 border: 1px solid #ddd;  
 }  
 }

————————————————————————————————————————

## navbar.php

<?php  
/\*\*  
 \* Admin Navbar Component  
 \* For Admin only with conditions to determine active page  
 \*/  
  
// Get current page name  
$current\_page = basename($\_SERVER['PHP\_SELF'], '.php');  
$current\_folder = basename(dirname($\_SERVER['PHP\_SELF']));  
  
// Determine active page based on conditions  
$active\_page = '';  
  
if ($current\_page == 'HomePage' || $current\_folder == 'HomePage') {  
 $active\_page = 'HomePage';  
} elseif ($current\_page == 'ManageVenueRequests' || $current\_folder == 'ManageVenueRequests') {  
 $active\_page = 'ManageVenueRequests';  
} elseif ($current\_page == 'MangeSports' || $current\_folder == 'MangeSports') {  
 $active\_page = 'MangeSports';  
} elseif ($current\_page == 'ReviewComplaints' || $current\_folder == 'ReviewComplaints') {  
 $active\_page = 'ReviewComplaints';  
}  
  
// Function to check if page is active  
function isActive($pageName) {  
 global $active\_page;  
 return ($active\_page === $pageName) ? 'active' : '';  
}  
?>  
  
<!-- Admin Navbar -->  
<link rel="stylesheet" href="../global.css">  
<nav class="admin-navbar">  
 <div class="nav-container">  
 <h1 class="logo">Book&Play Admin Panel</h1>  
 <ul class="nav-links">  
 <li><a href="../HomePage/HomePage.php" class="<?php echo isActive('HomePage'); ?>">HomePage</a></li>  
 <li><a href="../ManageVenueRequests/ManageVenueRequests.php" class="<?php echo isActive('ManageVenueRequests'); ?>">Manage Venue Requests</a></li>  
 <li><a href="../MangeSports/MangeSports.php" class="<?php echo isActive('MangeSports'); ?>">Manage Sports</a></li>  
 <li><a href="../ReviewComplaints/ReviewComplaints.php" class="<?php echo isActive('ReviewComplaints'); ?>">Review Complaints</a></li>  
 <li><a href="../../../logout.php" class="logout">Logout</a></li>  
 </ul>  
 </div>  
</nav>  
  
<!-- Debug Information -->  
<script>  
console.log('=== ADMIN NAVBAR DEBUG ===');  
console.log('Current Page:', '<?php echo $current\_page; ?>');  
console.log('Current Folder:', '<?php echo $current\_folder; ?>');  
console.log('Active Page:', '<?php echo $active\_page; ?>');  
</script>

————————————————————————————————————————

## HomePage\HomePage.css

/\* Light Elegant Theme - FreshGlow UI for Admin - Responsive Version \*/  
  
:root {  
 --primary: #1e90ff;  
 --secondary: #22c55e;  
 --light-bg: #f8f6f0;  
 --card-bg: #ffffff;  
 --card-border: #d1d5db;  
 --text-main: #1e293b;  
 --muted-text: #6b7280;  
 --accent-glow: rgba(30, 144, 255, 0.3);  
 --radius: 0.75rem;  
}  
  
\* {  
 box-sizing: border-box;  
 margin: 0;  
 padding: 0;  
}  
  
body {  
 margin: 0;  
 font-family: 'Outfit', sans-serif;  
 background: linear-gradient(135deg, #e0f7fa, #d0f8ce);  
 color: var(--text-main);  
 min-height: 100vh;  
 display: flex;  
 flex-direction: column;  
 align-items: center;  
 overflow-x: hidden;  
 padding: 0 15px;  
}  
  
  
  
.header-bar {  
 width: 100%;  
 padding: 20px 40px;  
 display: flex;  
 justify-content: space-between;  
 align-items: center;  
 box-sizing: border-box;  
 }  
 .logo-title {  
 font-size: 32px;  
 font-weight: bold;  
 background: linear-gradient(to right, #1e90ff, #22c55e);  
 -webkit-background-clip: text;  
 background-clip: text;  
 -webkit-text-fill-color: transparent;  
 }  
 .logo-title span {  
 color: #f472b6;  
 }  
  
.logo-title small {  
 font-size: 16px;  
 font-weight: normal;  
 color: var(--muted-text);  
 margin-left: 10px;  
 display: inline-block;  
}  
  
/\* Card grid \*/  
.admin-buttons {  
 display: flex;  
 flex-wrap: wrap;  
 justify-content: center;  
 gap: 40px;  
 padding: 0 20px 60px;  
 margin-bottom: 40px;  
 width: 100%;  
 max-width: 1200px;  
}  
  
/\* Design for each card \*/  
.glow-card {  
 width: 200px;  
 height: 200px;  
 background: var(--card-bg);  
 border: 1px solid var(--card-border);  
 border-radius: var(--radius);  
 padding: 20px;  
 text-align: center;  
 color: var(--text-main);  
 box-shadow: 0 4px 12px rgba(30, 144, 255, 0.05);  
 transition: all 0.3s ease;  
 cursor: pointer;  
 display: flex;  
 flex-direction: column;  
 justify-content: center;  
 align-items: center;  
}  
  
.glow-card img.icon {  
 width: 80px;  
 margin-bottom: 12px;  
 max-width: 100%;  
 height: auto;  
}  
  
.glow-card span {  
 font-size: 15px;  
 font-weight: 500;  
 color: var(--text-main);  
 text-align: center;  
}  
  
.glow-card:hover {  
 transform: translateY(-5px);  
 box-shadow: 0 6px 20px rgba(30, 144, 255, 0.15);  
}  
  
/\* Glow colors according to card type \*/  
.btn-VenueRequests { box-shadow: 0 0 12px #22c55e; }  
.btn-complaints { box-shadow: 0 0 12px gold; }  
.btn-sports { box-shadow: 0 0 12px #38bdf8; }  
.btn-logout { box-shadow: 0 0 12px #f43f5e; }  
  
/\* Footer \*/  
footer {  
 background-color: transparent;  
 padding: 20px;  
 width: 100%;  
 text-align: center;  
 font-size: 14px;  
 color: var(--muted-text);  
 margin-top: auto;  
}  
  
.footer-content {  
 max-width: 1200px;  
 margin: auto;  
 padding: 0 15px;  
}  
  
@keyframes fadeInUp {  
 from { opacity: 0; transform: translateY(20px); }  
 to { opacity: 1; transform: translateY(0); }  
}  
  
/\* Responsive Media Queries \*/  
  
/\* Large devices (desktops, less than 1200px) \*/  
@media screen and (max-width: 1200px) {  
 .admin-buttons {  
 gap: 30px;  
 }  
}  
  
/\* Medium devices (tablets, less than 992px) \*/  
@media screen and (max-width: 992px) {  
 .logo-title {  
 font-size: 36px;  
 margin-bottom: 50px;  
 }  
   
 .admin-title {  
 margin-top: 50px;  
 }  
   
 .glow-card {  
 width: 180px;  
 height: 180px;  
 }  
   
 .glow-card img.icon {  
 width: 70px;  
 }  
   
 .admin-buttons {  
 gap: 25px;  
 padding-bottom: 50px;  
 }  
}  
  
/\* Small devices (landscape phones, less than 768px) \*/  
@media screen and (max-width: 768px) {  
 .logo-title {  
 font-size: 30px;  
 margin-bottom: 40px;  
 }  
   
 .logo-title small {  
 font-size: 14px;  
 margin-left: 5px;  
 }  
   
 .admin-title {  
 margin-top: 40px;  
 }  
   
 .glow-card {  
 width: 160px;  
 height: 160px;  
 padding: 15px;  
 }  
   
 .glow-card img.icon {  
 width: 60px;  
 margin-bottom: 10px;  
 }  
   
 .glow-card span {  
 font-size: 14px;  
 }  
   
 .admin-buttons {  
 gap: 20px;  
 padding-bottom: 40px;  
 margin-bottom: 30px;  
 }  
}  
 .logo-title {  
 font-size: 32px;  
 font-weight: bold;  
 background: linear-gradient(to right, #1e90ff, #22c55e);  
 -webkit-background-clip: text;  
 background-clip: text;  
 -webkit-text-fill-color: transparent;  
 }  
 .logo-title span {  
 color: #f472b6;  
 }  
 /\* Profile dropdown \*/  
 .profile-container {  
 position: relative;  
 cursor: pointer;  
 z-index: 100;  
}  
.profile-container {  
 position: absolute;  
 top: 15px;  
 right: 15px;  
 }  
.profile-icon {  
 width: 60px;  
 height: 60px;  
 border-radius: 50%;  
 object-fit: cover;  
 box-shadow: 0 0 6px var(--accent-glow);  
}  
  
 .profile-menu {  
 position: absolute;  
 top: 50px;  
 right: 0;  
 background-color: white;  
 border: 1px solid var(--card-border);  
 border-radius: var(--radius);  
 box-shadow: 0 4px 16px rgba(0, 0, 0, 0.06);  
 overflow: hidden;  
 display: none;  
 flex-direction: column;  
 min-width: 180px;  
 z-index: 101;  
}  
  
.profile-menu a {  
 padding: 12px 18px;  
 text-decoration: none;  
 color: var(--text-main);  
 font-size: 14px;  
}  
  
.profile-menu a:hover {  
 background-color: #f3f4f6;  
}  
  
.profile-menu.active {  
 display: flex;  
}  
 .profile-menu {  
 min-width: 160px;  
 right: -10px;  
 }  
   
/\* Extra small devices (portrait phones, less than 576px) \*/  
@media screen and (max-width: 576px) {  
 .logo-title {  
 font-size: 26px;  
 margin-bottom: 30px;  
 }  
   
 .logo-title small {  
 font-size: 12px;  
 display: block;  
 margin-left: 0;  
 margin-top: 5px;  
 }  
   
  
   
 .admin-buttons {  
 gap: 15px;  
 padding-bottom: 30px;  
 }  
   
 .glow-card {  
 width: calc(50% - 15px);  
 min-width: 140px;  
 height: 150px;  
 padding: 12px;  
 }  
   
 .glow-card img.icon {  
 width: 50px;  
 margin-bottom: 8px;  
 }  
   
 .glow-card span {  
 font-size: 13px;  
 }  
   
 footer {  
 padding: 15px;  
 font-size: 12px;  
 }  
}  
  
/\* Very small devices (small phones, less than 380px) \*/  
@media screen and (max-width: 380px) {  
 .logo-title {  
 font-size: 24px;  
 }  
   
 .admin-buttons {  
 flex-direction: column;  
 align-items: center;  
 gap: 15px;  
 }  
   
 .glow-card {  
 width: 90%;  
 max-width: 200px;  
 height: 130px;  
 }  
}  
.username-display {  
 margin-left: 20px;  
 font-weight: 600;  
 color: var(--text-main);  
 font-size: 15px;  
}

————————————————————————————————————————

## HomePage\HomePage.html

<!DOCTYPE html>  
<html lang="en">  
<head>  
 <meta charset="UTF-8" />  
 <meta name="viewport" content="width=device-width, initial-scale=1.0"/>  
 <title>Admin Dashboard | Book&Play</title>  
 <link rel="stylesheet" href="HomePage.css" />  
 <link href="https://fonts.googleapis.com/css2?family=Outfit:wght@500;600&display=swap" rel="stylesheet">  
</head>  
<body>  
 <div class="header-bar">  
 <div class="logo-title">BOOK&PLAY</div>  
 <div class="profile-container" onclick="toggleProfileMenu()">  
 <span id="usernameDisplay" class="username-display"></span>  
 <img  
 id="userProfileImage"  
 class="profile-icon"  
 src="../../../uploads/users/default.jpg"  
 alt="Profile Image"  
 />  
 <div class="profile-menu" id="profileMenu">  
 <a href="../../auth/EditProfile/EditProfile.php">Edit Profile</a>  
 <a href="../../../logout.php">Logout</a>  
 </div>  
 </div>  
   
 </div>  
  
  
 <!-- Admin Cards -->  
 <div class="admin-buttons">  
 <a href="../ManageVenueRequests/ManageVenueRequests.php" class="glow-card btn-VenueRequests">  
 <img src="../../../Images/MangeVenue\_icon.png" class="icon" alt="Venue">  
 <span>Manage Venue Requests</span>  
 </a>  
  
  
  
 <a href="../ReviewComplaints/ReviewComplaints.php" class="glow-card btn-complaints">  
 <img src="../../../Images/Complaints\_icon.png" class="icon" alt="Complaints">  
 <span>Review Complaints</span>  
 </a>  
  
 <a href="../MangeSports/MangeSports.php" class="glow-card btn-sports">  
 <img src="../../../Images/MangeSport\_icon.png" class="icon" alt="Edit Sports">  
 <span>Manage Sports</span>  
 </a>  
 </div>  
  
 <div class="admin-buttons">  
 <a href="../../auth/User\_Selection\_Page/user-selection.php" class="glow-card btn-logout">  
 <img src="../../../Images/BackToUserSelecionRed.png" class="icon" alt="back to user selection">  
 <span>Back To User Selction Page</span>  
 </a>  
 </div>  
  
 <footer>  
 <div class="footer-content">  
 © 2025 Book&Play. All rights reserved.  
 </div>  
 </footer>  
 <script src="HomePage.js"></script>  
  
</body>  
</html>

————————————————————————————————————————

## HomePage\HomePage.js

function toggleProfileMenu() {  
 const menu = document.getElementById('profileMenu');  
 menu.classList.toggle('active');  
 }  
  
 // Close the menu when clicking outside  
 document.addEventListener('click', function (e) {  
 const profileContainer = document.querySelector('.profile-container');  
 const menu = document.getElementById('profileMenu');  
 if (!profileContainer.contains(e.target)) {  
 menu.classList.remove('active');  
 }  
 });  
document.addEventListener("DOMContentLoaded", () => {  
 fetch("HomePage.php?action=get\_user\_image")  
 .then((res) => res.json())  
 .then((data) => {  
 const imgElement = document.getElementById("userProfileImage");  
 const usernameSpan = document.getElementById("usernameDisplay");  
 if (imgElement && data.image) {  
 imgElement.src = `../../../uploads/users/${data.image}`;  
 }  
 if (usernameSpan && data.username) {  
 usernameSpan.textContent = data.username;  
 }  
 })  
 .catch((err) => {  
 console.error("Failed to load user image or username", err);  
 });  
});

————————————————————————————————————————

## HomePage\HomePage.php

<?php  
session\_start();  
require\_once '../../../db.php';   
// Save current page as the last visited (for back button or redirects)  
$\_SESSION['previous\_page'] = $\_SERVER['PHP\_SELF'];  
  
// ✅ 1. Handle AJAX request to get user image and username  
if (isset($\_GET['action']) && $\_GET['action'] === 'get\_user\_image') {  
 $username = $\_SESSION['user\_id'] ?? '';  
  
 if (!$username) {  
 echo json\_encode(['error' => 'Not logged in']);  
 exit;  
 }  
  
 $stmt = $conn->prepare("SELECT user\_image, username FROM users WHERE username = ?");  
 $stmt->bind\_param("s", $username);  
 $stmt->execute();  
 $result = $stmt->get\_result();  
  
 if ($user = $result->fetch\_assoc()) {  
 echo json\_encode([  
 'image' => $user['user\_image'],  
 'username' => $user['username']  
 ]);  
 } else {  
 echo json\_encode(['error' => 'User not found']);  
 }  
 exit;  
}  
  
// ✅ 2. Protect this page: only accessible if user is an admin  
  
$user\_name = $\_SESSION['user\_id'] ?? null;  
  
if (!$user\_name) {  
 // No user session – redirect to login  
 header('Location: ../../auth/Login\_Page/Login.php');  
 exit;  
}  
  
$stmt = $conn->prepare("SELECT \* FROM users WHERE username = ?");  
$stmt->bind\_param("s", $user\_name);  
$stmt->execute();  
$result = $stmt->get\_result();  
$user = $result->fetch\_assoc();  
  
// If not an admin, log them out and redirect  
if (!$user || $user['is\_admin'] != 1) {  
 session\_unset();  
 session\_destroy();  
 header('Location: ../../auth/Login\_Page/Login.php');  
 exit();  
}  
  
// ✅ 3. Load admin homepage HTML if access is allowed  
include 'HomePage.html';  
exit;  
?>

————————————————————————————————————————

## ManageVenueRequests\ManageVenueRequests.css

/\* Base styles for the page \*/  
body {  
 font-family: 'Inter', sans-serif;  
 margin: 0;  
 padding: 0;  
 background-color: #f5f5f5;  
}  
  
.container {  
 max-width: 1200px;  
 margin: 0 auto;  
 padding: 20px;  
}  
  
/\* Venue card styling \*/  
.venue-card {  
 background: white;  
 border-radius: 12px;  
 padding: 20px;  
 margin-bottom: 20px;  
 box-shadow: 0 2px 10px rgba(0, 0, 0, 0.1);  
 border: 1px solid #e0e0e0;  
}  
  
.venue-card h3 {  
 color: #2c3e50;  
 margin: 0 0 15px 0;  
 font-size: 20px;  
 font-weight: 600;  
}  
  
.venue-card p {  
 color: #555;  
 margin: 8px 0;  
 font-size: 14px;  
}  
  
/\* Button styling \*/  
.actions {  
 display: flex;  
 gap: 10px;  
 margin-top: 15px;  
 flex-wrap: wrap;  
}  
  
.actions button {  
 padding: 10px 16px;  
 border: none;  
 border-radius: 6px;  
 font-size: 14px;  
 font-weight: 500;  
 cursor: pointer;  
 transition: all 0.3s ease;  
 min-width: 80px;  
}  
  
.btn-view {  
 background-color: #3498db;  
 color: white;  
}  
  
.btn-view:hover {  
 background-color: #2980b9;  
 transform: translateY(-1px);  
}  
  
.btn-approve {  
 background-color: #27ae60;  
 color: white;  
}  
  
.btn-approve:hover {  
 background-color: #229954;  
 transform: translateY(-1px);  
}  
  
.btn-reject {  
 background-color: #e74c3c;  
 color: white;  
}  
  
.btn-reject:hover {  
 background-color: #c0392b;  
 transform: translateY(-1px);  
}  
  
/\* Modal styling \*/  
.modal {  
 display: none;  
 position: fixed;  
 z-index: 1000;  
 left: 0;  
 top: 0;  
 width: 100%;  
 height: 100%;  
 background-color: rgba(0, 0, 0, 0.5);  
 justify-content: center;  
 align-items: center;  
}  
  
.modal-content {  
 background-color: white;  
 padding: 30px;  
 border-radius: 12px;  
 max-width: 500px;  
 width: 90%;  
 position: relative;  
 box-shadow: 0 4px 20px rgba(0, 0, 0, 0.15);  
}  
  
.close-btn {  
 position: absolute;  
 right: 15px;  
 top: 15px;  
 font-size: 24px;  
 cursor: pointer;  
 color: #999;  
}  
  
.close-btn:hover {  
 color: #333;  
}  
  
.modal-buttons {  
 margin-top: 20px;  
 display: flex;  
 gap: 10px;  
 justify-content: flex-end;  
}  
  
.modal-buttons button {  
 padding: 10px 20px;  
 border: none;  
 border-radius: 6px;  
 font-size: 14px;  
 font-weight: 500;  
 cursor: pointer;  
 transition: all 0.3s ease;  
}  
  
.btn-resolve {  
 background-color: #27ae60;  
 color: white;  
}  
  
.btn-resolve:hover {  
 background-color: #229954;  
}  
  
/\* Responsive design \*/  
@media screen and (max-width: 768px) {  
 .actions {  
 flex-direction: column;  
 }  
   
 .actions button {  
 width: 100%;  
 }  
   
 .modal-content {  
 margin: 20px;  
 padding: 20px;  
 }  
}

————————————————————————————————————————

## ManageVenueRequests\ManageVenueRequests.html

<!DOCTYPE html>  
<html lang="en">  
<head>  
 <meta charset="UTF-8">  
 <title>Manage Venue Requests</title>  
 <link rel="stylesheet" href="../global.css">  
 <link rel="stylesheet" href="ManageVenueRequests.css">  
</head>  
<body>  
  
 <!-- Cards Container - Dynamic -->  
 <div class="cards-container" id="venueContainer"></div>  
  
 <!-- Modal Container -->  
 <div id="reportModal" class="modal" style="display: none;">  
 <div class="modal-content">  
 <span class="close" onclick="closeModal()">&times;</span>  
 <h2 id="modalTitle">Venue Details</h2>  
 <p id="modalMessage" class="long-description"></p>  
 <div class="actions">  
 <button class="approve">Approve</button>  
 <button class="reject">Reject</button>  
 </div>  
 </div>  
 </div>  
  
 <script src="ManageVenueRequests.js"></script>  
</body>  
</html>

————————————————————————————————————————

## ManageVenueRequests\ManageVenueRequests.js

// ManageVenueRequests.js  
  
document.addEventListener("DOMContentLoaded", function () {  
 const venueContainer = document.getElementById("venueContainer");  
 const modal = document.getElementById("reportModal");  
   
 console.log("🔄 Loading venue requests and unaccepted venues...");  
   
 // Use the consolidated API endpoint with the 'get\_reports' action  
 fetch("VenueAPI.php?action=get\_reports")  
 .then(res => {  
 console.log("📡 Response status:", res.status);  
 if (!res.ok) {  
 throw new Error(`HTTP error! status: ${res.status}`);  
 }  
 return res.json();  
 })  
 .then(data => {  
 console.log("📊 Received data:", data);  
 console.log("📊 Data length:", data.length);  
   
 // Check if data is an error response  
 if (data.success === false) {  
 console.error("❌ API returned error:", data.message);  
 venueContainer.innerHTML = `<p>❌ Error: ${data.message}</p>`;  
 return;  
 }  
   
 renderReports(data);  
 })  
 .catch(err => {  
 console.error("❌ Error loading reports:", err);  
 venueContainer.innerHTML = `<p>❌ Failed to load reports: ${err.message}</p>`;  
 });  
  
 function renderReports(reports) {  
 venueContainer.innerHTML = "";  
  
 if (reports.length === 0) {  
 venueContainer.innerHTML = "<p>No venue requests or unaccepted venues found.</p>";  
 return;  
 }  
  
 console.log("📋 Rendering", reports.length, "items");  
   
 reports.forEach((item, index) => {  
 console.log(`📋 Item ${index + 1}:`, item);  
   
 const card = document.createElement("div");  
 card.className = "venue-card";  
   
 // Determine if this is a report or venue  
 const isReport = item.source\_type === 'report';  
 const placeName = item.place\_name || "Unnamed Place";  
 const submittedBy = item.username || "Unknown";  
 const createdAt = item.created\_at || "Unknown date";  
 const reportId = item.report\_id || null;  
 const facilitiesId = item.facilities\_id;  
   
 card.innerHTML = `  
 <h3>${placeName}</h3>  
 <p><strong>Type:</strong> ${isReport ? 'Venue Suggestion' : 'Unaccepted Venue'}</p>  
 <p><strong>Submitted by:</strong> ${submittedBy}</p>  
 <p><strong>Created:</strong> ${createdAt}</p>  
 ${item.location ? `<p><strong>Location:</strong> ${item.location}</p>` : ''}  
 ${item.price ? `<p><strong>Price:</strong> ₪${item.price}</p>` : ''}  
 <div class="actions">  
 <button class="btn-view" onclick="openVenue(${facilitiesId})">View Details</button>  
 ${isReport && reportId ? `  
 <button class="btn-approve" onclick="handleAction('approve', ${reportId}, ${facilitiesId})">Approve</button>  
 <button class="btn-reject" onclick="handleAction('reject', ${reportId}, ${facilitiesId})">Reject</button>  
 ` : `  
 <button class="btn-approve" onclick="handleAction('approve', null, ${facilitiesId})">Approve</button>  
 <button class="btn-reject" onclick="handleAction('reject', null, ${facilitiesId})">Reject</button>  
 `}  
 </div>  
 `;  
 venueContainer.appendChild(card);  
 });  
 }  
  
 window.openVenue = function(id) {  
 if (!id) {  
 alert("Venue ID not available");  
 return;  
 }  
 window.location.href = `../VenueDetails/VenueDetails.php?id=${id}`;  
 };  
  
 window.closeModal = function () {  
 modal.style.display = "none";  
 };  
  
 window.handleAction = function (subaction, reportId, facilitiesId) {  
 // Updated to use the consolidated API with action=handle\_action  
 fetch("VenueAPI.php", {  
 method: "POST",  
 headers: {  
 "Content-Type": "application/x-www-form-urlencoded"  
 },  
 body: new URLSearchParams({  
 action: "handle\_action",  
 subaction: subaction,  
 report\_id: reportId,  
 facilities\_id: facilitiesId  
 })  
 })  
 .then(res => res.json())  
 .then(data => {  
 alert(data.message);  
 if (data.success) {  
 location.reload();  
 }  
 })  
 .catch(err => {  
 alert("❌ Something went wrong");  
 console.error(err);  
 });  
 };  
  
 // Add a function to update report status (previously in UpdateReportStatus.php)  
 window.updateReportStatus = function(reportId, status) {  
 fetch("VenueAPI.php", {  
 method: "POST",  
 headers: {  
 "Content-Type": "application/x-www-form-urlencoded"  
 },  
 body: new URLSearchParams({  
 action: "update\_status",  
 report\_id: reportId,  
 status: status  
 })  
 })  
 .then(res => res.json())  
 .then(data => {  
 alert(data.message);  
 if (data.success) {  
 location.reload();  
 }  
 })  
 .catch(err => {  
 alert("❌ Something went wrong");  
 console.error(err);  
 });  
 };  
});

————————————————————————————————————————

## ManageVenueRequests\ManageVenueRequests.php

<?php  
session\_start();  
require\_once '../../../db.php';   
// Check if user is admin by querying the database  
$user\_name = $\_SESSION['user\_id'];  
$stmt = $conn->prepare("SELECT \* FROM users WHERE username = ?");  
$stmt->bind\_param("s", $user\_name);  
$stmt->execute();  
$result = $stmt->get\_result();  
  
if (!isset($\_SESSION['user\_id']) || $result->fetch\_assoc()['is\_admin'] != 1) {  
 // Unset all session variables  
 session\_unset();  
 // Destroy the session completely  
 session\_destroy();  
 header('Location: ../../auth/Login\_Page/Login.php');  
 exit();  
}  
include '../navbar.php';  
include 'ManageVenueRequests.html';  
exit();  
?>

————————————————————————————————————————

## ManageVenueRequests\VenueAPI.php

<?php  
session\_start();  
require\_once '../../../db.php';  
header('Content-Type: application/json');  
  
// Add debugging  
error\_log("=== VENUEAPI.PHP DEBUG START ===");  
error\_log("Request method: " . $\_SERVER['REQUEST\_METHOD']);  
error\_log("Session user\_id: " . ($\_SESSION['user\_id'] ?? 'null'));  
  
// Process the request based on the action parameter  
$action = $\_REQUEST['action'] ?? '';  
error\_log("Action requested: " . $action);  
  
switch ($action) {  
 case 'get\_reports':  
 getReports();  
 break;  
 case 'handle\_action':  
 handleAction();  
 break;  
 case 'update\_status':  
 updateReportStatus();  
 break;  
 default:  
 echo json\_encode(["success" => false, "message" => "⛔ Invalid action specified."]);  
 break;  
}  
  
// Function to get all reports  
function getReports() {  
 global $conn;  
   
 error\_log("=== GETREPORTS DEBUG START ===");  
 error\_log("Session user\_id: " . ($\_SESSION['user\_id'] ?? 'null'));  
   
 // Get both venue suggestions (reports) and unaccepted venues  
 // For venue suggestions: exclude suggestions submitted by the current admin  
 // For unaccepted venues: exclude venues owned by the current admin  
 $sql = "SELECT   
 'report' as source\_type,  
 r.report\_id,  
 r.username,  
 r.suggested\_place\_name as place\_name,  
 r.facilities\_id,  
 r.created\_at,  
 r.message,  
 sf.location,  
 sf.price,  
 sf.image\_url  
 FROM reports r  
 LEFT JOIN sportfacilities sf ON r.facilities\_id = sf.facilities\_id  
 WHERE r.type = 'suggest\_place'  
 AND r.username != ?  
   
 UNION ALL  
   
 SELECT   
 'venue' as source\_type,  
 NULL as report\_id,  
 sf.owner\_username as username,  
 sf.place\_name,  
 sf.facilities\_id,  
 NOW() as created\_at,  
 CONCAT('Sport: ', sf.SportCategory, ' | Price: ', sf.price, ' | Location: ', sf.location) as message,  
 sf.location,  
 sf.price,  
 sf.image\_url  
 FROM sportfacilities sf  
 WHERE sf.is\_Accepted = 0  
 AND sf.owner\_username != ?  
   
 ORDER BY created\_at DESC";  
   
 error\_log("SQL Query: " . $sql);  
 error\_log("Filtering out suggestions by: " . $\_SESSION['user\_id']);  
 error\_log("Filtering out venues owned by: " . $\_SESSION['user\_id']);  
   
 $stmt = $conn->prepare($sql);  
 $stmt->bind\_param("ss", $\_SESSION['user\_id'], $\_SESSION['user\_id']);  
 $stmt->execute();  
 $result = $stmt->get\_result();  
   
 if (!$result) {  
 error\_log("❌ SQL Error: " . $conn->error);  
 echo json\_encode(["success" => false, "message" => "Database error: " . $conn->error]);  
 return;  
 }  
   
 error\_log("Query executed. Found " . ($result ? $result->num\_rows : 0) . " items");  
   
 $reports = [];  
   
 if ($result && $result->num\_rows > 0) {  
 while ($row = $result->fetch\_assoc()) {  
 $reports[] = $row;  
 error\_log("Item found: " . $row['source\_type'] . " - " . $row['place\_name'] . " (Owner: " . $row['username'] . ")");  
 }  
 } else {  
 error\_log("No venue requests or unaccepted venues found");  
 }  
   
 error\_log("Returning " . count($reports) . " items");  
 error\_log("=== GETREPORTS DEBUG END ===");  
 echo json\_encode($reports);  
}  
  
// Function to handle approve/reject actions  
function handleAction() {  
 global $conn;  
   
 if ($\_SERVER['REQUEST\_METHOD'] !== 'POST') {  
 echo json\_encode(["success" => false, "message" => "⛔ Only POST method allowed."]);  
 exit;  
 }  
   
 $action = $\_POST['subaction'] ?? '';  
 $reportId = $\_POST['report\_id'] ?? null;  
 $facilitiesId = $\_POST['facilities\_id'] ?? null;  
   
 error\_log("Handle action: $action, reportId: $reportId, facilitiesId: $facilitiesId");  
   
 if (!$facilitiesId || !in\_array($action, ['approve', 'reject'])) {  
 echo json\_encode(["success" => false, "message" => "❌ Invalid input."]);  
 exit;  
 }  
   
 if ($action === 'approve') {  
 // ✅ Update is\_Accepted to 1  
 $update = $conn->prepare("UPDATE sportfacilities SET is\_Accepted = 1 WHERE facilities\_id = ?");  
 $update->bind\_param("i", $facilitiesId);  
 $update->execute();  
 $update->close();  
   
 // If there's a report\_id, delete the report  
 if ($reportId) {  
 $deleteReport = $conn->prepare("DELETE FROM reports WHERE report\_id = ?");  
 $deleteReport->bind\_param("i", $reportId);  
 $deleteReport->execute();  
 $deleteReport->close();  
 }  
   
 echo json\_encode(["success" => true, "message" => "✅ Approved successfully."]);  
 exit;  
 } elseif ($action === 'reject') {  
 // ❌ Delete from sportfacilities table  
 $deleteFacility = $conn->prepare("DELETE FROM sportfacilities WHERE facilities\_id = ?");  
 $deleteFacility->bind\_param("i", $facilitiesId);  
 $deleteFacility->execute();  
 $deleteFacility->close();  
   
 // If there's a report\_id, delete the report too  
 if ($reportId) {  
 $deleteReport = $conn->prepare("DELETE FROM reports WHERE report\_id = ?");  
 $deleteReport->bind\_param("i", $reportId);  
 $deleteReport->execute();  
 $deleteReport->close();  
 }  
   
 echo json\_encode(["success" => true, "message" => "❌ Rejected and deleted."]);  
 exit;  
 }  
}  
  
// Function to update report status  
function updateReportStatus() {  
 global $conn;  
   
 if ($\_SERVER['REQUEST\_METHOD'] !== 'POST') {  
 echo json\_encode(["success" => false, "message" => "⛔ Only POST method allowed."]);  
 exit;  
 }  
   
 $report\_id = $\_POST['report\_id'] ?? null;  
 $status = $\_POST['status'] ?? null;  
   
 if (!$report\_id || !in\_array($status, ['accepted', 'rejected'])) {  
 echo json\_encode(["success" => false, "message" => "❌ Invalid input."]);  
 exit;  
 }  
   
 $stmt = $conn->prepare("UPDATE reports SET status = ? WHERE report\_id = ?");  
 $stmt->bind\_param("si", $status, $report\_id);  
   
 if ($stmt->execute()) {  
 echo json\_encode(["success" => true, "message" => "✅ Report updated successfully."]);  
 } else {  
 echo json\_encode(["success" => false, "message" => "❌ Failed to update report."]);  
 }  
   
 $stmt->close();  
}  
  
error\_log("=== VENUEAPI.PHP DEBUG END ===");  
?>

————————————————————————————————————————

## MangeSports\MangeSports.css

.menu-toggle span {  
 display: block;  
 height: 3px;  
 width: 100%;  
 background-color: #0f172a;  
 border-radius: 3px;  
 transition: all 0.3s ease;  
}  
  
/\* Card & Table \*/  
.fresh-card {  
 background: var(--card-bg);  
 backdrop-filter: blur(10px);  
 border: 1px solid var(--card-border);  
 border-radius: var(--radius);  
 box-shadow: var(--shadow);  
 padding: 30px;  
 margin-bottom: 40px;  
 width: 95%;  
 max-width: 1000px;  
 animation: fadeInUp 0.4s ease;  
}  
  
.fresh-card h2 {  
 margin-bottom: 20px;  
 font-size: 22px;  
}  
  
.fresh-table {  
 width: 100%;  
 border-collapse: collapse;  
 border: 1px solid #e2e8f0;  
 overflow-x: auto;  
}  
  
.table-container {  
 width: 100%;  
 overflow-x: auto;  
 -webkit-overflow-scrolling: touch;  
}  
  
.fresh-table th, .fresh-table td {  
 padding: 14px;  
 text-align: center;  
 border: 1px solid #e2e8f0;  
 background-color: #ffffffc7;  
}  
  
/\* Buttons \*/  
.toggle-buttons {  
 display: flex;  
 gap: 20px;  
 margin-bottom: 40px;  
 flex-wrap: wrap;  
 justify-content: center;  
}  
  
.add-sport-form {  
 display: flex;  
 gap: 15px;  
 justify-content: center;  
 margin-top: 20px;  
 flex-wrap: wrap;  
}  
  
.add-sport-form input {  
 padding: 12px;  
 font-size: 15px;  
 border: 1px solid #cbd5e1;  
 border-radius: 8px;  
 width: 250px;  
 max-width: 100%;  
 background-color: #f8fafc;  
 color: var(--text-main);  
}  
  
.btn {  
 padding: 10px 20px;  
 border-radius: 10px;  
 border: none;  
 font-weight: bold;  
 cursor: pointer;  
 font-size: 14px;  
 transition: all 0.3s ease;  
 min-width: 100px;  
}  
  
.btn:hover {  
 transform: scale(1.05);  
}  
  
.btn.primary {  
 background: linear-gradient(to right, var(--primary), var(--secondary));  
 color: white;  
}  
  
.btn.green {  
 background: var(--secondary);  
 color: white;  
}  
  
.btn.blue {  
 background: var(--primary);  
 color: white;  
}  
  
.btn.delete {  
 background-color: var(--danger);  
 color: white;  
}  
  
.btn.accept {  
 background-color: var(--secondary);  
 color: white;  
}  
  
.hidden {  
 display: none;  
}  
  
.message-box {  
 margin: 10px 0;  
 padding: 10px;  
 font-weight: bold;  
 border-radius: 5px;  
 display: none;  
 width: 100%;  
 text-align: center;  
}  
  
.message-success {  
 color: green;  
 background-color: #e3f9e5;  
 border: 1px solid #8bd48b;  
}  
  
.message-error {  
 color: red;  
 background-color: #fce4e4;  
 border: 1px solid #f5a3a3;  
}  
  
@keyframes fadeInUp {  
 from { opacity: 0; transform: translateY(20px); }  
 to { opacity: 1; transform: translateY(0); }  
}  
  
/\* Responsive Media Queries \*/  
  
/\* Large screens (desktops, up to 1200px) \*/  
@media screen and (max-width: 1200px) {  
 .fresh-card {  
 padding: 25px;  
 }  
   
 .toggle-buttons {  
 gap: 15px;  
 }  
}  
  
/\* Medium screens (tablets, up to 992px) \*/  
@media screen and (max-width: 992px) {  
 .nav-container {  
 padding: 0 15px;  
 }  
   
 .logo-title {  
 font-size: 36px;  
 }  
   
 .fresh-card {  
 padding: 20px;  
 }  
   
 .fresh-table th, .fresh-table td {  
 padding: 12px 10px;  
 font-size: 14px;  
 }  
   
 .btn {  
 padding: 10px 15px;  
 font-size: 13px;  
 }  
   
 .add-sport-form input {  
 width: 220px;  
 }  
}  
  
/\* Small screens (landscape phones, up to 768px) \*/  
@media screen and (max-width: 768px) {  
 body {  
 padding: 30px 15px;  
 }  
   
 .admin-navbar {  
 padding: 12px 0;  
 }  
   
 .nav-container {  
 justify-content: center;  
 }  
   
 .logo {  
 margin-bottom: 10px;  
 text-align: center;  
 }  
   
 .nav-links {  
 width: 100%;  
 justify-content: center;  
 }  
   
 .nav-links li a {  
 padding: 8px 14px;  
 font-size: 14px;  
 }  
   
 .logo-title {  
 font-size: 32px;  
 }  
   
 .admin-title {  
 margin: 30px 0 15px;  
 }  
   
 .fresh-card {  
 padding: 20px 15px;  
 width: 100%;  
 }  
   
 .fresh-card h2 {  
 font-size: 20px;  
 margin-bottom: 15px;  
 }  
   
 .toggle-buttons {  
 gap: 10px;  
 margin-bottom: 30px;  
 }  
   
 .fresh-table th, .fresh-table td {  
 padding: 10px 8px;  
 font-size: 13px;  
 }  
   
 .add-sport-form {  
 gap: 10px;  
 flex-direction: column;  
 align-items: center;  
 }  
   
 .add-sport-form input {  
 width: 100%;  
 max-width: 300px;  
 }  
   
 .btn {  
 padding: 8px 12px;  
 width: 100%;  
 max-width: 300px;  
 }  
}  
  
/\* Extra small screens (phones, up to 576px) \*/  
@media screen and (max-width: 576px) {  
 body {  
 padding: 20px 10px;  
 }  
   
 /\* Implement hamburger menu for mobile \*/  
 .menu-toggle {  
 display: flex;  
 position: absolute;  
 right: 20px;  
 top: 20px;  
 }  
   
 .nav-container {  
 flex-direction: column;  
 align-items: center;  
 padding-top: 15px;  
 padding-bottom: 15px;  
 }  
   
 .nav-links {  
 flex-direction: column;  
 width: 100%;  
 gap: 8px;  
 max-height: 0;  
 overflow: hidden;  
 transition: max-height 0.5s ease;  
 }  
   
 .nav-links.active {  
 max-height: 500px;  
 }  
   
 .nav-links li {  
 width: 100%;  
 }  
   
 .nav-links li a {  
 text-align: center;  
 width: 100%;  
 }  
   
 .logo-title {  
 font-size: 28px;  
 }  
   
 .admin-title {  
 margin: 25px 0 15px;  
 }  
   
 .fresh-card {  
 padding: 15px 10px;  
 }  
   
 .fresh-card h2 {  
 font-size: 18px;  
 }  
   
 .toggle-buttons {  
 flex-direction: column;  
 width: 100%;  
 max-width: 250px;  
 margin-left: auto;  
 margin-right: auto;  
 gap: 8px;  
 }  
   
 .btn {  
 width: 100%;  
 }  
   
 .fresh-table th, .fresh-table td {  
 padding: 8px 5px;  
 font-size: 12px;  
 }  
   
 .message-box {  
 padding: 8px;  
 font-size: 13px;  
 }  
}  
  
/\* Very small screens (small phones, up to 375px) \*/  
@media screen and (max-width: 375px) {  
 .logo-title {  
 font-size: 24px;  
 }  
   
 .fresh-card {  
 padding: 12px 8px;  
 }  
   
 .fresh-table th, .fresh-table td {  
 padding: 6px 4px;  
 font-size: 11px;  
 }  
}  
  
/\* CSS for JavaScript-controlled mobile menu \*/  
/\* Add this JavaScript to make the mobile menu work:  
document.addEventListener('DOMContentLoaded', function() {  
 const menuToggle = document.querySelector('.menu-toggle');  
 const navLinks = document.querySelector('.nav-links');  
   
 if (menuToggle && navLinks) {  
 menuToggle.addEventListener('click', function() {  
 navLinks.classList.toggle('active');  
 });  
 }  
});  
\*/

————————————————————————————————————————

## MangeSports\MangeSports.html

<!DOCTYPE html>  
<html lang="en">  
<head>  
 <meta charset="UTF-8" />  
 <meta name="viewport" content="width=device-width, initial-scale=1.0"/>  
 <title>Manage Sports | Admin</title>  
 <link rel="stylesheet" href="MangeSports.css" />  
 <link href="https://fonts.googleapis.com/css2?family=Outfit:wght@500;600&display=swap" rel="stylesheet">  
 <script defer src="MangeSports.js"></script>  
</head>  
<body>  
  
  
 <!-- ✅ Current Sports Section -->  
 <section class="fresh-card">  
 <h2>Current Sports</h2>  
 <table class="fresh-table">  
 <thead>  
 <tr>  
 <th>#</th>  
 <th>Sport Name</th>  
 <th>Action</th>  
 </tr>  
 </thead>  
 <tbody id="currentSportsTableBody">  
 <!-- Will be filled using JavaScript -->  
 </tbody>  
 </table>  
 </section>  
 <div id="sportMessage" class="message-box"></div>  
 <!-- ✅ Toggle Buttons -->  
 <div class="toggle-buttons">  
 <button onclick="toggleSection('suggestedSection'); loadSuggestedSports()" class="btn blue">  
 View Suggested Sports  
 </button>   
 <button onclick="toggleSection('addSection')" class="btn green">+ Add New Sport</button>  
 </div>  
  
 <!-- ✅ Suggested Sports Section -->  
 <section class="fresh-card hidden" id="suggestedSection">  
 <h2>Suggested Sports</h2>  
 <table class="fresh-table">  
 <thead>  
 <tr>  
 <th>#</th>  
 <th>Suggested Sport</th>  
 <th>Actions</th>  
 </tr>  
 </thead>  
 <tbody id="suggestedSportsTableBody">  
 <!-- Will be filled using JavaScript -->  
 </tbody>  
 </table>  
 </section>  
   
  
 <!-- ✅ Add New Sport Section -->  
 <section class="fresh-card hidden" id="addSection">  
 <h2>Add New Sport</h2>  
 <form class="add-sport-form">  
 <input type="text" placeholder="Sport Name" id="newSportName" required />  
 <button id="addSportBtn" class="btn primary">Add Sport</button>  
 </form>  
 </section>  
  
</body>  
</html>

————————————————————————————————————————

## MangeSports\MangeSports.js

document.addEventListener("DOMContentLoaded", function () {  
 loadAcceptedSports(); // 🔹 Load accepted sports  
  
 // 🔹 Load accepted sports  
 function loadAcceptedSports() {  
 fetch("MangeSportsController.php?action=get\_accepted\_sports")  
 .then(res => res.json())  
 .then(data => {  
 const tableBody = document.getElementById("currentSportsTableBody");  
 tableBody.innerHTML = "";  
  
 data.forEach((sport, index) => {  
 const row = document.createElement("tr");  
 row.innerHTML = `  
 <td>${index + 1}</td>  
 <td>${sport.sport\_name}</td>  
 <td><button class="btn delete" data-sport-id="${sport.sport\_id}" onclick="deleteSport(${sport.sport\_id})">Delete</button></td>  
 `;  
 tableBody.appendChild(row);  
 });  
 })  
 .catch(err => {  
 console.error("❌ Failed to load accepted sports:", err);  
 });  
 }  
  
 // 🔹 Toggle section visibility  
 window.toggleSection = function (id) {  
 document.getElementById(id).classList.toggle("hidden");  
 };  
  
 // 🔹 Load suggested sports  
 window.loadSuggestedSports = function () {  
 fetch("MangeSportsController.php?action=get\_suggested\_sports")  
 .then(res => res.json())  
 .then(data => {  
 const table = document.getElementById("suggestedSportsTableBody");  
 table.innerHTML = "";  
 data.forEach((report, index) => {  
 const row = document.createElement("tr");  
 row.innerHTML = `  
 <td>${index + 1}</td>  
 <td>${report.suggested\_sport\_name}</td>  
 <td>  
 <button class="btn accept" onclick="handleAction('accept', ${report.report\_id}, '${report.suggested\_sport\_name}')">Accept</button>  
 <button class="btn delete" onclick="handleAction('reject', ${report.report\_id})">Reject</button>  
 </td>  
 `;  
 table.appendChild(row);  
 });  
 })  
 .catch(err => {  
 console.error("❌ Failed to load suggested sports:", err);  
 });  
 };  
  
 // 🔹 Handle accept/reject action  
 window.handleAction = function (action, reportId, sportName = null) {  
 const bodyData = new URLSearchParams({ action, report\_id: reportId });  
 if (sportName) bodyData.append("sport\_name", sportName);  
  
 fetch("MangeSportsController.php", {  
 method: "POST",  
 headers: { "Content-Type": "application/x-www-form-urlencoded" },  
 body: bodyData  
 })  
 .then(res => res.json())  
 .then(data => {  
 alert(data.message);  
 if (data.success) {  
 loadSuggestedSports();  
 loadAcceptedSports();  
 }  
 })  
 .catch(err => {  
 console.error("❌ Failed to handle action:", err);  
 });  
 };  
  
 // 🔹 Add a new sport  
 document.getElementById("addSportBtn").addEventListener("click", () => {  
 const name = document.getElementById("newSportName").value.trim();  
 if (!name) return alert("Please enter sport name");  
  
 fetch("MangeSportsController.php", {  
 method: "POST",  
 headers: { "Content-Type": "application/x-www-form-urlencoded" },  
 body: new URLSearchParams({ action: "add\_sport", name })  
 })  
 .then(res => res.json())  
 .then(data => {  
 alert(data.message);  
 if (data.success) {  
 loadSuggestedSports();  
 loadAcceptedSports();  
 }  
 })  
 .catch(err => {  
 console.error("❌ Failed to add sport:", err);  
 });  
 });  
  
 // 🔹 Delete a sport  
 window.deleteSport = function (sportId) {  
 const button = document.querySelector(`button[data-sport-id="${sportId}"]`);  
 const row = button.closest("tr");  
  
 if (!confirm("Are you sure you want to delete this sport?")) return;  
  
 button.disabled = true;  
 button.textContent = "Deleting...";  
  
 fetch("MangeSportsController.php", {  
 method: "POST",  
 headers: { "Content-Type": "application/x-www-form-urlencoded" },  
 body: new URLSearchParams({ action: "delete\_sport", sport\_id: sportId })  
 })  
 .then(res => res.json())  
 .then(data => {  
 showMessage(data.message, data.success);  
  
 if (data.success) {  
 row.remove();  
 setTimeout(() => {  
 loadAcceptedSports();  
 }, 1000);  
 } else {  
 button.disabled = false;  
 button.textContent = "Delete";  
 }  
 })  
 .catch(err => {  
 console.error("❌ Failed to delete sport:", err);  
 showMessage("Something went wrong while deleting!", false);  
 button.disabled = false;  
 button.textContent = "Delete";  
 });  
 };  
  
 // 🔹 Show success or error message  
 function showMessage(message, isSuccess = true) {  
 const msgBox = document.getElementById("sportMessage");  
 msgBox.textContent = message;  
 msgBox.className = "message-box " + (isSuccess ? "message-success" : "message-error");  
 msgBox.style.display = "block";  
  
 // Hide message automatically after 4 seconds  
 setTimeout(() => {  
 msgBox.style.display = "none";  
 }, 4000);  
 }  
});

————————————————————————————————————————

## MangeSports\MangeSports.php

<?php  
session\_start();  
require\_once '../../../db.php';   
// Check if user is admin by querying the database  
$user\_name = $\_SESSION['user\_id'];  
$stmt = $conn->prepare("SELECT \* FROM users WHERE username = ?");  
$stmt->bind\_param("s", $user\_name);  
$stmt->execute();  
$result = $stmt->get\_result();  
  
if (!isset($\_SESSION['user\_id']) || $result->fetch\_assoc()['is\_admin'] != 1) {  
 // Unset all session variables  
 session\_unset();  
 // Destroy the session completely  
 session\_destroy();  
 header('Location: ../../auth/Login\_Page/Login.php');  
 exit();  
}  
include '../navbar.php';  
include 'MangeSports.html';  
exit();  
?>

————————————————————————————————————————

## MangeSports\MangeSportsController.php

<?php  
require\_once '../../../db.php';  
header('Content-Type: application/json');  
  
// Determine requested action  
$action = $\_POST['action'] ?? $\_GET['action'] ?? '';  
  
switch ($action) {  
  
 // 🟢 Add new sport directly (already accepted)  
 case 'add\_sport':  
 $name = $\_POST['name'] ?? '';  
 if (!$name) {  
 echo json\_encode(["success" => false, "message" => "❌ Sport name is required."]);  
 exit;  
 }  
  
 $stmt = $conn->prepare("INSERT INTO sports (sport\_name, is\_Accepted) VALUES (?, 1)");  
 $stmt->bind\_param("s", $name);  
  
 if ($stmt->execute()) {  
 echo json\_encode(["success" => true, "message" => "✅ Sport added and pending approval"]);  
 } else {  
 echo json\_encode(["success" => false, "message" => "❌ Failed to add sport"]);  
 }  
 $stmt->close();  
 break;  
  
 // ✅ Accept suggested sport and add to sports table  
 case 'accept':  
 $report\_id = intval($\_POST['report\_id'] ?? 0);  
 $sport\_name = trim($\_POST['sport\_name'] ?? '');  
  
 if (!$sport\_name) {  
 echo json\_encode(["success" => false, "message" => "❌ Sport name is missing."]);  
 exit;  
 }  
  
 $stmt = $conn->prepare("INSERT INTO sports (sport\_name, is\_Accepted) VALUES (?, 1)");  
 $stmt->bind\_param("s", $sport\_name);  
 $stmt->execute();  
 $stmt->close();  
  
 $deleteReport = $conn->prepare("DELETE FROM reports WHERE report\_id = ?");  
 $deleteReport->bind\_param("i", $report\_id);  
 $deleteReport->execute();  
 $deleteReport->close();  
  
 echo json\_encode(["success" => true, "message" => "✅ Sport accepted and added successfully."]);  
 break;  
  
 // ❌ Reject suggested sport (only delete report)  
 case 'reject':  
 $report\_id = intval($\_POST['report\_id'] ?? 0);  
 $deleteReport = $conn->prepare("DELETE FROM reports WHERE report\_id = ?");  
 $deleteReport->bind\_param("i", $report\_id);  
 $deleteReport->execute();  
 $deleteReport->close();  
  
 echo json\_encode(["success" => true, "message" => "🗑️ Sport suggestion rejected and removed."]);  
 break;  
  
 // 📋 Fetch accepted sports  
 case 'get\_accepted\_sports':  
 $sports = [];  
 $sql = "SELECT \* FROM sports WHERE is\_Accepted = 1";  
 $result = $conn->query($sql);  
  
 if ($result && $result->num\_rows > 0) {  
 while ($row = $result->fetch\_assoc()) {  
 $sports[] = $row;  
 }  
 }  
  
 echo json\_encode($sports);  
 break;  
  
 // 📥 Fetch sport suggestion reports  
 case 'get\_suggested\_sports':  
 $reports = [];  
 $sql = "SELECT \* FROM reports WHERE type = 'suggest\_sport' ORDER BY created\_at DESC";  
 $result = $conn->query($sql);  
  
 if ($result && $result->num\_rows > 0) {  
 while ($row = $result->fetch\_assoc()) {  
 $reports[] = $row;  
 }  
 }  
  
 echo json\_encode($reports);  
 break;  
  
 // 🗑️ Delete a sport  
 case 'delete\_sport':  
 $sport\_id = intval($\_POST['sport\_id'] ?? 0);  
 if ($sport\_id <= 0) {  
 echo json\_encode(["success" => false, "message" => "❌ Invalid sport ID."]);  
 exit;  
 }  
  
 $stmt = $conn->prepare("DELETE FROM sports WHERE sport\_id = ?");  
 $stmt->bind\_param("i", $sport\_id);  
 if ($stmt->execute()) {  
 echo json\_encode(["success" => true, "message" => "🗑️ Sport deleted successfully."]);  
 } else {  
 echo json\_encode(["success" => false, "message" => "❌ Failed to delete sport."]);  
 }  
 $stmt->close();  
 break;  
  
 // ⛔ Unrecognized or missing action  
 default:  
 echo json\_encode(["success" => false, "message" => "⛔ Invalid or missing action."]);  
 break;  
}  
  
$conn->close();  
?>

————————————————————————————————————————

## ReviewComplaints\ReviewComplaints.css

/\* Card \*/  
.card {  
 background: rgba(255, 255, 255, 0.8);  
 border-radius: 20px;  
 box-shadow: 0 0 40px rgba(0, 150, 136, 0.25);  
 padding: 25px;  
 width: 90%;  
 max-width: 1000px;  
 margin: 40px auto;  
 backdrop-filter: blur(10px);  
}  
  
h2 {  
 text-align: center;  
 font-size: 28px;  
 background: linear-gradient(to right, #00bcd4, #4caf50);  
 -webkit-background-clip: text;  
 -webkit-text-fill-color: transparent;  
}  
  
table {  
 width: 100%;  
 border-collapse: collapse;  
 margin-top: 25px;  
}  
  
th, td {  
 padding: 14px;  
 text-align: center;  
 border-bottom: 1px solid #ccc;  
}  
  
th {  
 background-color: #f1f5f9;  
}  
  
.btn {  
 padding: 8px 16px;  
 border: none;  
 border-radius: 12px;  
 font-weight: bold;  
 cursor: pointer;  
 transition: all 0.3s ease;  
}  
  
.btn-view {  
 background: linear-gradient(to right, #4fc3f7, #81c784);  
 color: white;  
 box-shadow: 0 0 10px rgba(79, 195, 247, 0.5);  
}  
  
.btn-view:hover {  
 transform: scale(1.05);  
}  
  
/\* Modal Styles \*/  
.modal {  
 display: none;  
 position: fixed;  
 z-index: 1050;  
 left: 0;  
 top: 0;  
 width: 100%;  
 height: 100%;  
 overflow: auto;  
 background-color: rgba(0,0,0,0.5);  
 align-items: center;  
 justify-content: center;  
}  
  
.modal-content {  
 background: white;  
 margin: auto;  
 padding: 25px;  
 border-radius: 20px;  
 width: 80%;  
 max-width: 500px;  
 box-shadow: 0 5px 30px rgba(0,0,0,0.3);  
 position: relative;  
 animation: modalFadeIn 0.3s;  
}  
  
@keyframes modalFadeIn {  
 from {opacity: 0; transform: translateY(-20px);}  
 to {opacity: 1; transform: translateY(0);}  
}  
  
.close-btn {  
 position: absolute;  
 right: 20px;  
 top: 15px;  
 font-size: 28px;  
 font-weight: bold;  
 cursor: pointer;  
 color: #aaa;  
}  
  
.close-btn:hover {  
 color: #333;  
}  
  
.modal-buttons {  
 display: flex;  
 justify-content: space-between;  
 margin-top: 20px;  
}  
  
.btn-resolve {  
 background: linear-gradient(to right, #4caf50, #81c784);  
 color: white;  
 box-shadow: 0 0 10px rgba(76, 175, 80, 0.5);  
}  
  
.btn-remove {  
 background: linear-gradient(to right, #f44336, #ff7961);  
 color: white;  
 box-shadow: 0 0 10px rgba(244, 67, 54, 0.5);  
}  
  
.btn-resolve:hover, .btn-remove:hover {  
 transform: scale(1.05);  
}

————————————————————————————————————————

## ReviewComplaints\ReviewComplaints.html

<!DOCTYPE html>  
<html lang="en">  
<head>  
 <meta charset="UTF-8">  
 <title>Review Venue Complaints | Book&Play</title>  
 <link rel="stylesheet" href="ReviewComplaints.css">  
</head>  
<body>  
  
  
 <div class="card">  
 <h2>Review Venue Complaints</h2>  
 <table>  
 <thead>  
 <tr>  
 <th>Sender</th>  
 <th>Venue ID</th>  
 <th>Venue Owner</th>  
 <th>Complaint Reason</th>  
 <th>Action</th>  
 </tr>  
 </thead>  
 <tbody id="complaintsTable">  
 <!-- Complaints will be inserted here -->  
 </tbody>  
 </table>  
 </div>  
  
 <!-- Modal for complaint details -->  
 <div id="complaintModal" class="modal">  
 <div class="modal-content">  
 <span class="close-btn" onclick="document.getElementById('complaintModal').style.display='none'">&times;</span>  
 <h3>Complaint Details</h3>  
 <p id="modalMessage"></p>  
 <div class="modal-buttons">  
 <button class="btn btn-resolve" onclick="resolve()">Mark as Resolved</button>  
 </div>  
 </div>  
 </div>  
  
 <script src="ReviewComplaints.js"></script>  
</body>  
</html>

————————————————————————————————————————

## ReviewComplaints\ReviewComplaints.js

document.addEventListener("DOMContentLoaded", function () {  
 loadPlaceReports();  
});  
  
// Store the current complaint data  
let currentComplaint = null;  
  
function loadPlaceReports() {  
 console.log("🔄 Loading place reports...");  
 fetch("ReviewComplaintsController.php?action=get\_place\_reports")  
 .then(res => {  
 console.log("📡 Response status:", res.status);  
 return res.json();  
 })  
 .then(data => {  
 console.log("📊 Received data:", data);  
 console.log("📊 Data length:", data.length);  
   
 const table = document.getElementById("complaintsTable");  
 table.innerHTML = "";  
  
 if (data.length === 0) {  
 console.log("📭 No complaints found");  
 const emptyRow = document.createElement("tr");  
 emptyRow.innerHTML = `<td colspan="5" style="text-align: center;">No complaints found</td>`;  
 table.appendChild(emptyRow);  
 return;  
 }  
  
 console.log("📋 Rendering", data.length, "complaints");  
 data.forEach((report, index) => {  
 console.log(`📋 Report ${index + 1}:`, report);  
 const row = document.createElement("tr");  
 row.innerHTML = `  
 <td>${report.username}</td>  
 <td>${report.facilities\_id || 'N/A'}</td>  
 <td>${report.owner\_username || 'Unknown'}</td>  
 <td>${report.Reason || 'No reason provided'}</td>  
 <td>  
 <button class="btn btn-view" onclick="openVenue(${report.facilities\_id})">View Venue</button>  
 <button class="btn btn-view" style="margin-left: 5px" onclick="showComplaintDetails(${JSON.stringify(report).replace(/"/g, '&quot;')})">Details</button>  
 </td>  
 `;  
 table.appendChild(row);  
 });  
 })  
 .catch(err => {  
 console.error("❌ Failed to load place reports:", err);  
 const table = document.getElementById("complaintsTable");  
 table.innerHTML = `<tr><td colspan="5" style="text-align: center; color: red;">Failed to load reports. Please try again.</td></tr>`;  
 });  
}  
  
function openVenue(id) {  
 if (!id) {  
 alert("Venue ID not available");  
 return;  
 }  
 window.location.href = `../VenueDetails/VenueDetails.php?id=${id}`;  
}  
  
function showComplaintDetails(report) {  
 currentComplaint = report;  
   
 let message = `  
 <strong>Reported by:</strong> ${report.username}<br>  
 <strong>Venue Owner:</strong> ${report.owner\_username || 'Unknown'}<br>  
 <strong>Reason:</strong> ${report.Reason || 'No reason provided'}<br>  
 <strong>Venue ID:</strong> ${report.facilities\_id || 'N/A'}<br>  
 <strong>Date:</strong> ${new Date(report.created\_at).toLocaleString()}<br>  
 <strong>Message:</strong><br>${report.message || 'No additional message'}  
 `;  
   
 openModal(message);  
}  
  
function openModal(message) {  
 document.getElementById("modalMessage").innerHTML = message;  
 document.getElementById("complaintModal").style.display = "flex";  
}  
  
function resolve() {  
 if (!currentComplaint) {  
 alert("No complaint selected");  
 return;  
 }  
   
 const formData = new FormData();  
 formData.append('action', 'mark\_resolved');  
 formData.append('id', currentComplaint.report\_id);  
   
 fetch("ReviewComplaintsController.php", {  
 method: 'POST',  
 body: formData  
 })  
 .then(res => res.json())  
 .then(data => {  
 if (data.success) {  
 alert("✅ Complaint marked as resolved");  
 document.getElementById("complaintModal").style.display = "none";  
 loadPlaceReports(); // Reload the table  
 } else {  
 alert("❌ Error: " + (data.message || "Failed to mark as resolved"));  
 }  
 })  
 .catch(err => {  
 console.error("❌ Failed to resolve complaint:", err);  
 alert("Failed to mark complaint as resolved");  
 });  
}  
  
  
window.onclick = function(event) {  
 const modal = document.getElementById("complaintModal");  
 if (event.target === modal) modal.style.display = "none";  
};

————————————————————————————————————————

## ReviewComplaints\ReviewComplaints.php

<?php  
session\_start();  
require\_once '../../../db.php';   
// Check if user is admin by querying the database  
$user\_name = $\_SESSION['user\_id'];  
$stmt = $conn->prepare("SELECT \* FROM users WHERE username = ?");  
$stmt->bind\_param("s", $user\_name);  
$stmt->execute();  
$result = $stmt->get\_result();  
  
if (!isset($\_SESSION['user\_id']) || $result->fetch\_assoc()['is\_admin'] != 1) {  
 // Unset all session variables  
 session\_unset();  
 // Destroy the session completely  
 session\_destroy();  
 header('Location: ../../auth/Login\_Page/Login.php');  
 exit();  
}  
include '../navbar.php';  
include 'ReviewComplaints.html';  
exit();  
?>

————————————————————————————————————————

## ReviewComplaints\ReviewComplaintsController.php

<?php  
session\_start();  
require\_once '../../../db.php';  
header('Content-Type: application/json');  
  
// Add debugging  
error\_log("=== REVIEWCOMPLAINTS CONTROLLER DEBUG START ===");  
error\_log("Request method: " . $\_SERVER['REQUEST\_METHOD']);  
error\_log("Session user\_id: " . ($\_SESSION['user\_id'] ?? 'null'));  
error\_log("Action: " . ($\_POST['action'] ?? $\_GET['action'] ?? 'none'));  
  
// Determine required operation type  
$action = $\_POST['action'] ?? $\_GET['action'] ?? '';  
  
switch ($action) {  
 case 'get\_place\_reports':  
 $reports = [];  
 error\_log("Fetching place reports...");  
   
 // Get venue reports specifically (type = 'report\_place')  
 // Exclude reports where the admin is the venue owner to prevent conflicts of interest  
 $sql = "SELECT r.\*, sf.owner\_username   
 FROM reports r  
 LEFT JOIN sportfacilities sf ON r.facilities\_id = sf.facilities\_id  
 WHERE r.type = 'report\_place'   
 AND (sf.owner\_username IS NULL OR sf.owner\_username != ?)  
 ORDER BY r.created\_at DESC";  
 $stmt = $conn->prepare($sql);  
 $stmt->bind\_param("s", $\_SESSION['user\_id']);  
 $stmt->execute();  
 $result = $stmt->get\_result();  
  
 error\_log("Query executed. Found " . $result->num\_rows . " reports");  
  
 if ($result && $result->num\_rows > 0) {  
 while ($row = $result->fetch\_assoc()) {  
 $reports[] = $row;  
 error\_log("Report found: ID=" . $row['report\_id'] . ", Type=" . $row['type'] . ", Username=" . $row['username'] . ", Venue Owner=" . $row['owner\_username']);  
 }  
 } else {  
 error\_log("No reports found in database");  
 }  
  
 error\_log("Returning " . count($reports) . " reports");  
 echo json\_encode($reports);  
 break;  
   
 case 'mark\_resolved':  
 $report\_id = isset($\_POST['id']) ? intval($\_POST['id']) : 0;  
   
 if ($report\_id <= 0) {  
 echo json\_encode(['success' => false, 'message' => 'Invalid report ID']);  
 exit;  
 }  
   
 // Here you would typically update a status field, but since your table   
 // structure doesn't have this field, we'll just delete the report  
 $sql = "DELETE FROM reports WHERE report\_id = ?";  
 $stmt = $conn->prepare($sql);  
 $stmt->bind\_param("i", $report\_id);  
   
 if ($stmt->execute()) {  
 echo json\_encode(['success' => true]);  
 } else {  
 echo json\_encode(['success' => false, 'message' => 'Failed to update report']);  
 }  
 break;  
   
 case 'remove\_venue':  
 $facility\_id = isset($\_POST['facility\_id']) ? intval($\_POST['facility\_id']) : 0;  
   
 if ($facility\_id <= 0) {  
 echo json\_encode(['success' => false, 'message' => 'Invalid facility ID']);  
 exit;  
 }  
   
 // In a real implementation, you might want to:  
 // 1. Remove related records first (bookings, reports, etc.)  
 // 2. Then remove the facility itself  
 // 3. Maybe mark as inactive instead of deleting  
   
 // Check if a status column exists in the facilities table  
 $statusCheck = $conn->query("SHOW COLUMNS FROM facilities LIKE 'status'");  
   
 if ($statusCheck && $statusCheck->num\_rows > 0) {  
 // If status column exists, update it  
 $sql = "UPDATE facilities SET status = 'removed' WHERE facilities\_id = ?";  
 $stmt = $conn->prepare($sql);  
 $stmt->bind\_param("i", $facility\_id);  
 } else {  
 // If no status column, just delete the facility  
 $sql = "DELETE FROM facilities WHERE facilities\_id = ?";  
 $stmt = $conn->prepare($sql);  
 $stmt->bind\_param("i", $facility\_id);  
 }  
   
 if ($stmt->execute()) {  
 // Now delete related reports  
 $sql = "DELETE FROM reports WHERE report\_id = ?";  
 $stmt = $conn->prepare($sql);  
 $stmt->bind\_param("i", $report\_id);  
 $stmt->execute();  
   
 echo json\_encode(['success' => true]);  
 } else {  
 echo json\_encode(['success' => false, 'message' => 'Failed to remove venue']);  
 }  
 break;  
   
 default:  
 echo json\_encode(['error' => 'Invalid action']);  
 break;  
}  
  
error\_log("=== REVIEWCOMPLAINTS CONTROLLER DEBUG END ===");  
?>

————————————————————————————————————————

## ReviewComplaints\test\_reports.php

<?php  
session\_start();  
require\_once '../../../db.php';  
  
echo "<h2>Database Reports Test</h2>";  
  
// Check all reports  
$sql = "SELECT \* FROM reports ORDER BY created\_at DESC";  
$result = $conn->query($sql);  
  
echo "<h3>All Reports in Database:</h3>";  
if ($result && $result->num\_rows > 0) {  
 echo "<table border='1'>";  
 echo "<tr><th>ID</th><th>Username</th><th>Type</th><th>Facility ID</th><th>Reason</th><th>Created</th></tr>";  
 while ($row = $result->fetch\_assoc()) {  
 echo "<tr>";  
 echo "<td>" . $row['report\_id'] . "</td>";  
 echo "<td>" . $row['username'] . "</td>";  
 echo "<td>" . $row['type'] . "</td>";  
 echo "<td>" . $row['facilities\_id'] . "</td>";  
 echo "<td>" . $row['Reason'] . "</td>";  
 echo "<td>" . $row['created\_at'] . "</td>";  
 echo "</tr>";  
 }  
 echo "</table>";  
} else {  
 echo "<p>No reports found in database.</p>";  
}  
  
// Check specifically for report\_place type  
echo "<h3>Venue Reports (type = 'report\_place'):</h3>";  
$sql2 = "SELECT \* FROM reports WHERE type = 'report\_place' ORDER BY created\_at DESC";  
$result2 = $conn->query($sql2);  
  
if ($result2 && $result2->num\_rows > 0) {  
 echo "<table border='1'>";  
 echo "<tr><th>ID</th><th>Username</th><th>Type</th><th>Facility ID</th><th>Reason</th><th>Created</th></tr>";  
 while ($row = $result2->fetch\_assoc()) {  
 echo "<tr>";  
 echo "<td>" . $row['report\_id'] . "</td>";  
 echo "<td>" . $row['username'] . "</td>";  
 echo "<td>" . $row['type'] . "</td>";  
 echo "<td>" . $row['facilities\_id'] . "</td>";  
 echo "<td>" . $row['Reason'] . "</td>";  
 echo "<td>" . $row['created\_at'] . "</td>";  
 echo "</tr>";  
 }  
 echo "</table>";  
} else {  
 echo "<p>No venue reports found.</p>";  
}  
  
// Check table structure  
echo "<h3>Reports Table Structure:</h3>";  
$sql3 = "DESCRIBE reports";  
$result3 = $conn->query($sql3);  
  
if ($result3) {  
 echo "<table border='1'>";  
 echo "<tr><th>Field</th><th>Type</th><th>Null</th><th>Key</th><th>Default</th><th>Extra</th></tr>";  
 while ($row = $result3->fetch\_assoc()) {  
 echo "<tr>";  
 echo "<td>" . $row['Field'] . "</td>";  
 echo "<td>" . $row['Type'] . "</td>";  
 echo "<td>" . $row['Null'] . "</td>";  
 echo "<td>" . $row['Key'] . "</td>";  
 echo "<td>" . $row['Default'] . "</td>";  
 echo "<td>" . $row['Extra'] . "</td>";  
 echo "</tr>";  
 }  
 echo "</table>";  
}  
?>

————————————————————————————————————————

## VenueDetails\VenueDetails.css

\* {  
 box-sizing: border-box;  
 margin: 0;  
 padding: 0;  
}  
  
body {  
 font-family: "Outfit", sans-serif;  
 background: linear-gradient(to bottom right, #e0f7fa, #d0f8ce);  
 color: #0f172a;  
 margin: 0;  
 padding: 15px;  
 display: flex;  
 flex-direction: column;  
 align-items: center;  
 min-height: 100vh;  
 overflow-x: hidden;  
}  
  
.venue-card {  
 background: rgba(255, 255, 255, 0.85);  
 border-radius: 20px;  
 box-shadow: 0 0 30px rgba(0, 128, 128, 0.2);  
 padding: 30px;  
 max-width: 700px;  
 width: 95%;  
 margin-top: 60px;  
 backdrop-filter: blur(10px);  
}  
  
h2 {  
 font-size: 28px;  
 text-align: center;  
 margin-bottom: 20px;  
 background: linear-gradient(to right, #00bcd4, #4caf50);  
 -webkit-background-clip: text;  
 background-clip: text;  
 -webkit-text-fill-color: transparent;  
 word-wrap: break-word;  
}  
  
.info {  
 font-size: 18px;  
 margin: 10px 0;  
 padding: 10px;  
 border-bottom: 1px solid #ccc;  
 word-wrap: break-word;  
}  
  
#imagePreview {  
 width: 100%;  
 max-height: 300px;  
 object-fit: cover;  
 border-radius: 16px;  
 margin-bottom: 20px;  
}  
  
/\* Back Button Styling \*/  
.back-wrapper {  
 width: 100%;  
 display: flex;  
 justify-content: center;  
 margin: 40px 0 30px 0;  
}  
  
.btn-back {  
 background: linear-gradient(to right, #4fc3f7, #81c784);  
 color: #ffffff;  
 font-size: 20px;  
 padding: 16px 36px;  
 border: none;  
 border-radius: 16px;  
 font-weight: 600;  
 cursor: pointer;  
 transition: 0.3s ease-in-out;  
 box-shadow: 0 0 20px rgba(79, 195, 247, 0.4);  
 text-align: center;  
}  
  
.btn-back:hover {  
 transform: translateY(-3px) scale(1.04);  
 box-shadow: 0 0 25px rgba(129, 199, 132, 0.5);  
}  
  
/\* Error message styling \*/  
.error-message {  
 color: #f44336;  
 font-weight: 500;  
 text-align: center;  
 padding: 15px;  
 border: 1px solid #ffcdd2;  
 border-radius: 10px;  
 background-color: #ffebee;  
}  
  
/\* Responsive styles \*/  
@media screen and (max-width: 768px) {  
 .venue-card {  
 padding: 25px;  
 margin-top: 40px;  
 }  
  
 h2 {  
 font-size: 24px;  
 margin-bottom: 15px;  
 }  
  
 .info {  
 font-size: 16px;  
 padding: 8px;  
 }  
  
 #imagePreview {  
 max-height: 250px;  
 }  
  
 .back-wrapper {  
 margin: 30px 0 25px 0;  
 }  
  
 .btn-back {  
 font-size: 18px;  
 padding: 14px 30px;  
 }  
}  
  
@media screen and (max-width: 576px) {  
 body {  
 padding: 10px;  
 }  
  
 .venue-card {  
 padding: 20px;  
 margin-top: 30px;  
 width: 100%;  
 }  
  
 h2 {  
 font-size: 22px;  
 margin-bottom: 15px;  
 }  
  
 .info {  
 font-size: 15px;  
 padding: 8px 5px;  
 }  
  
 #imagePreview {  
 max-height: 200px;  
 border-radius: 12px;  
 }  
  
 .back-wrapper {  
 margin: 25px 0 20px 0;  
 }  
  
 .btn-back {  
 font-size: 16px;  
 padding: 12px 25px;  
 border-radius: 12px;  
 width: 80%;  
 max-width: 250px;  
 }  
}  
  
@media screen and (max-width: 375px) {  
 .venue-card {  
 padding: 15px;  
 margin-top: 20px;  
 border-radius: 15px;  
 }  
  
 h2 {  
 font-size: 20px;  
 margin-bottom: 12px;  
 }  
  
 .info {  
 font-size: 14px;  
 padding: 6px 4px;  
 }  
  
 #imagePreview {  
 max-height: 180px;  
 border-radius: 10px;  
 margin-bottom: 15px;  
 }  
  
 .btn-back {  
 font-size: 15px;  
 padding: 10px 20px;  
 border-radius: 10px;  
 }  
}

————————————————————————————————————————

## VenueDetails\VenueDetails.html

<!DOCTYPE html>  
<html lang="en" dir="ltr">  
 <head>  
 <meta charset="UTF-8" />  
 <title>Venue Details</title>  
 <link rel="stylesheet" href="VenueDetails.css" />  
 </head>  
 <body>  
 <div class="venue-card" id="venueCard">  
 <h2>Venue Details</h2>  
  
 <!-- Dynamic Content -->  
 <div class="info" id="placeName">  
 <strong>Place Name:</strong> Loading...  
 </div>  
 <div class="info" id="location">  
 <strong>Location:</strong> Loading...  
 </div>  
 <div class="info" id="description">  
 <strong>Description:</strong> Loading...  
 </div>  
 <div class="info" id="sportCategory">  
 <strong>Sport Category:</strong> Loading...  
 </div>  
 <div class="info" id="ownerUsername">  
 <strong>Owner Username:</strong> Loading...  
 </div>  
 <div class="info" id="ownerEmail">  
 <strong>Owner Email:</strong> Loading...  
 </div>  
 <!-- Venue Image -->  
 <div class="info" id="venueImage">  
 <img id="imagePreview" src="" alt="Venue Image" style="display: none" />  
 </div>  
 </div>  
  
 <!-- Back Button Centered -->  
 <div class="back-wrapper">  
 <button class="btn-back" onclick="history.back()">← Back</button>  
 </div>  
  
 <script src="VenueDetails.js"></script>  
 </body>  
</html>

————————————————————————————————————————

## VenueDetails\VenueDetails.js

document.addEventListener("DOMContentLoaded", function () {  
 // Get the venue ID from the URL  
 const urlParams = new URLSearchParams(window.location.search);  
 const venueId = urlParams.get("id");  
  
 if (venueId) {  
 loadVenueDetails(venueId);  
 } else {  
 showError("No venue ID provided.");  
 }  
});  
  
function loadVenueDetails(venueId) {  
 fetch(`VenueDetailsController.php?action=get\_venue\_details&id=${venueId}`)  
 .then((response) => response.json())  
 .then((data) => {  
 if (data.error) {  
 showError(data.error);  
 return;  
 }  
  
 displayVenueDetails(data);  
 })  
 .catch((error) => {  
 console.error("Error loading venue details:", error);  
 showError("Failed to load venue details. Please try again.");  
 });  
}  
  
function displayVenueDetails(venue) {  
 // Update the venue card with facility details  
 document.getElementById(  
 "placeName"  
 ).innerHTML = `<strong>Place Name:</strong> ${venue.place\_name || "N/A"}`;  
 document.getElementById("location").innerHTML = `<strong>Location:</strong> ${  
 venue.location || "N/A"  
 }`;  
 document.getElementById(  
 "description"  
 ).innerHTML = `<strong>Description:</strong> ${venue.description || "N/A"}`;  
 document.getElementById(  
 "sportCategory"  
 ).innerHTML = `<strong>Sport Category:</strong> ${  
 venue.SportCategory || "N/A"  
 }`;  
 document.getElementById(  
 "ownerUsername"  
 ).innerHTML = `<strong>Owner Username:</strong> ${  
 venue.owner\_username || "N/A"  
 }`;  
  
 // For email, we don't have it directly in the facilities table, but you could fetch it from users table if needed  
 document.getElementById(  
 "ownerEmail"  
 ).innerHTML = `<strong>Owner Email:</strong> ${venue.owner\_email || "N/A"}`;  
  
 // Update the image if available  
 const imagePreview = document.getElementById("imagePreview");  
 if (venue.image\_url && venue.image\_url.trim() !== "") {  
 imagePreview.src = `../../../uploads/venues/${venue.image\_url}`;  
 imagePreview.style.display = "block";  
 } else {  
 imagePreview.src = "../../../Images/staduim\_icon.png"; // Default image  
 imagePreview.style.display = "block";  
 }  
  
 // Show the venue card  
 document.getElementById("venueCard").style.display = "block";  
}  
  
function showError(message) {  
 const venueCard = document.getElementById("venueCard");  
 venueCard.innerHTML = `  
 <h2>Error</h2>  
 <div class="info error-message">${message}</div>  
 <div class="back-wrapper" style="margin-top: 20px;">  
 <button class="btn-back" onclick="history.back()">← Back</button>  
 </div>  
 `;  
 venueCard.style.display = "block";  
}

————————————————————————————————————————

## VenueDetails\VenueDetails.php

<?php  
session\_start();  
require\_once '../../../db.php';   
// Check if user is admin by querying the database  
$user\_name = $\_SESSION['user\_id'];  
$stmt = $conn->prepare("SELECT \* FROM users WHERE username = ?");  
$stmt->bind\_param("s", $user\_name);  
$stmt->execute();  
$result = $stmt->get\_result();  
  
if (!isset($\_SESSION['user\_id']) || $result->fetch\_assoc()['is\_admin'] != 1) {  
 // Unset all session variables  
 session\_unset();  
 // Destroy the session completely  
 session\_destroy();  
 header('Location: ../../auth/Login\_Page/Login.php');  
 exit();  
}  
include '../navbar.php';  
include 'VenueDetails.html';  
exit();  
?>

————————————————————————————————————————

## VenueDetails\VenueDetailsController.php

<?php  
require\_once '../../../db.php';  
header('Content-Type: application/json');  
  
$action = $\_POST['action'] ?? $\_GET['action'] ?? '';  
  
switch ($action) {  
 case 'get\_venue\_details':  
 // Get the venue ID from the request  
 $venue\_id = isset($\_GET['id']) ? intval($\_GET['id']) : 0;  
   
 if ($venue\_id <= 0) {  
 echo json\_encode(['error' => 'Invalid venue ID']);  
 exit;  
 }  
   
 // Query to get venue details from the facilities table  
 $sql = "SELECT f.\*, u.email as owner\_email  
 FROM sportfacilities f  
 LEFT JOIN users u ON f.owner\_username = u.username  
 WHERE f.facilities\_id = ?";  
   
 $stmt = $conn->prepare($sql);  
 $stmt->bind\_param("i", $venue\_id);  
 $stmt->execute();  
 $result = $stmt->get\_result();  
   
 if ($result && $result->num\_rows > 0) {  
 $venue = $result->fetch\_assoc();  
 echo json\_encode($venue);  
 } else {  
 echo json\_encode(['error' => 'Venue not found']);  
 }  
 break;  
   
 default:  
 echo json\_encode(['error' => 'Invalid action']);  
 break;  
}  
?>

————————————————————————————————————————

## global.css

:root {  
 --primary: #1e90ff;  
 --secondary: #22c55e;  
 --bg-light: #e0f7fa;  
 --bg-secondary: #d0f8ce;  
 --card-bg: rgba(255, 255, 255, 0.95);  
 --text-main: #0f172a;  
 --text-light: #475569;  
 --radius-sm: 8px;  
 --radius-md: 14px;  
 --radius-lg: 24px;  
 --shadow-sm: 0 4px 6px rgba(0, 0, 0, 0.05);  
 --shadow-md: 0 6px 16px rgba(0, 0, 0, 0.08);  
 --shadow-lg: 0 10px 25px rgba(0, 0, 0, 0.12);  
 --transition: all 0.3s cubic-bezier(0.4, 0, 0.2, 1);  
 }  
   
 \* {  
 box-sizing: border-box;  
 margin: 0;  
 padding: 0;  
 }  
   
 body {  
 font-family: 'Outfit', sans-serif;  
 margin: 0;  
 background: linear-gradient(135deg, var(--bg-light), var(--bg-secondary));  
 background-attachment: fixed;  
 color: var(--text-main);  
 min-height: 100vh;  
 line-height: 1.4;  
 }  
   
 /\* ===== NAVIGATION ===== \*/  
 .fresh-navbar {  
 display: flex;  
 justify-content: space-between;  
 align-items: center;  
 padding: 10px 30px;  
 background: linear-gradient(to right, #e0f7fa, #d0f8ce);  
 box-shadow: 0 2px 12px rgba(0, 0, 0, 0.05);  
 position: sticky;  
 top: 0;  
 z-index: 1000;  
 border-bottom: 1px solid rgba(0, 0, 0, 0.04);  
 height: 60px;  
 }  
   
 .logo {  
 font-size: 20px;  
 font-weight: bold;  
 background: linear-gradient(to right, #0ea5e9, #22c55e);  
 -webkit-background-clip: text;  
 -webkit-text-fill-color: transparent;  
 background-clip: text;  
 }  
   
 .nav-links {  
 list-style: none;  
 display: flex;  
 gap: 12px;  
 align-items: center;  
 margin: 0;  
 padding: 0;  
 }  
   
 .nav-links li a {  
 display: block;  
 padding: 8px 14px;  
 border-radius: 20px;  
 font-size: 13px;  
 font-weight: 500;  
 text-decoration: none;  
 color: var(--text-main);  
 background: white;  
 transition: var(--transition);  
 box-shadow: 0 2px 5px rgba(0, 0, 0, 0.04);  
 }  
   
 .nav-links li a:hover {  
 transform: translateY(-1px);  
 box-shadow: 0 3px 10px rgba(0, 0, 0, 0.06);  
 background: linear-gradient(to right, #4fc3f7, #81c784);  
 color: white;  
 }  
   
 .nav-links li a.active {  
 background: linear-gradient(to right, #4fc3f7, #81c784);  
 color: white;  
 font-weight: bold;  
 box-shadow: 0 0 10px rgba(79, 195, 247, 0.3);  
 }  
   
 .nav-links li a.logout {  
 background: rgba(244, 67, 54, 0.15);  
 color: #f44336;  
 font-weight: 600;  
 }  
   
 /\* ===== LAYOUT CONTAINERS ===== \*/  
 .container {  
 max-width: 1200px;  
 margin: 0 auto;  
 padding: 15px 20px;  
 position: relative;  
 }

————————————————————————————————————————

## navbar.php

<?php  
/\*\*  
 \* Facility Owner Navbar Component  
 \* For Facility Owner only with conditions to determine active page  
 \*/  
  
// Get current page name  
$current\_page = basename($\_SERVER['PHP\_SELF'], '.php');  
$current\_folder = basename(dirname($\_SERVER['PHP\_SELF']));  
  
// Determine active page based on conditions  
$active\_page = '';  
  
if ($current\_page == 'HomePage' || $current\_folder == 'HomePage') {  
 $active\_page = 'HomePage';  
} elseif ($current\_page == 'ManageVenue' || $current\_folder == 'ManageVenue') {  
 $active\_page = 'ManageVenue';  
} elseif ($current\_page == 'Bookings' || $current\_folder == 'Bookings') {  
 $active\_page = 'Bookings';  
} elseif ($current\_page == 'Analytics' || $current\_folder == 'Analytics') {  
 $active\_page = 'Analytics';  
} elseif ($current\_page == 'Messages' || $current\_folder == 'Messages') {  
 $active\_page = 'Messages';  
}  
  
// Function to check if page is active  
function isActive($pageName) {  
 global $active\_page;  
 return ($active\_page === $pageName) ? 'active' : '';  
}  
?>  
  
<!-- Facility Owner Navbar -->  
<link rel="stylesheet" href="../global.css">  
<nav class="fresh-navbar">  
 <div class="logo">Book<span>&</span>Play</div>  
 <ul class="nav-links">  
 <li><a href="../HomePage/HomePage.php" class="<?php echo isActive('HomePage'); ?>">HomePage</a></li>  
 <li><a href="../ManageVenue/ManageVenue.php" class="<?php echo isActive('ManageVenue'); ?>">Manage Venue</a></li>  
 <li><a href="../Bookings/Bookings.php" class="<?php echo isActive('Bookings'); ?>">Bookings</a></li>  
 <li><a href="../Analytics/Analytics.php" class="<?php echo isActive('Analytics'); ?>">Analytics</a></li>  
 <li><a href="../Messages/Messages.php" class="<?php echo isActive('Messages'); ?>">Messages</a></li>  
 <li><a href="../../../logout.php" class="logout">Logout</a></li>  
 </ul>  
</nav>  
  
<!-- Debug Information -->  
<script>  
console.log('=== FACILITY OWNER NAVBAR DEBUG ===');  
console.log('Current Page:', '<?php echo $current\_page; ?>');  
console.log('Current Folder:', '<?php echo $current\_folder; ?>');  
console.log('Active Page:', '<?php echo $active\_page; ?>');  
</script>

————————————————————————————————————————

## Analytics\Analytics.css

/\* Main venue view - restrict height to prevent scrolling \*/  
#cardSection {  
 height: calc(100vh - 300px);  
 overflow: hidden;  
}  
  
.main-content {  
 flex: 1;  
 padding: 15px;  
 max-width: 1400px;  
 margin: 0 auto;  
}  
  
/\* ===== PAGE HEADER SECTION ===== \*/  
.page-header {  
 margin-bottom: 20px;  
 background-color: rgba(224, 247, 250, 0.5);  
 padding: 20px;  
 border-radius: var(--radius-md);  
 text-align: center;  
 backdrop-filter: blur(5px);  
}  
  
.page-title {  
 font-size: 26px;  
 font-weight: 700;  
 margin-bottom: 5px;  
 color: var(--text-main);  
 letter-spacing: -0.02em;  
}  
  
.page-subtitle {  
 font-size: 14px;  
 color: var(--text-light);  
 font-weight: 400;  
 margin: 0;  
 max-width: 600px;  
 margin: 0 auto;  
}  
  
/\* ===== SEARCH & FILTER SECTION ===== \*/  
.search-sort-container {  
 display: flex;  
 justify-content: space-between;  
 align-items: center;  
 gap: 15px;  
 margin-bottom: 20px;  
 padding: 15px;  
 background: var(--card-bg);  
 border-radius: var(--radius-md);  
 box-shadow: var(--shadow-sm);  
}  
  
.search-bar {  
 flex: 1;  
 position: relative;  
 max-width: 350px;  
}  
  
.search-bar input {  
 width: 100%;  
 padding: 10px 15px 10px 35px;  
 border-radius: var(--radius-md);  
 border: 2px solid rgba(15, 23, 42, 0.1);  
 background-color: white;  
 font-size: 14px;  
 font-family: 'Outfit', sans-serif;  
 box-shadow: var(--shadow-sm);  
 transition: var(--transition);  
}  
  
.search-bar input:focus {  
 outline: none;  
 border-color: var(--primary);  
 box-shadow: 0 0 0 3px rgba(30, 144, 255, 0.1);  
}  
  
.search-icon {  
 position: absolute;  
 left: 12px;  
 top: 50%;  
 transform: translateY(-50%);  
 color: var(--text-light);  
 font-size: 14px;  
}  
  
.sort-container {  
 display: flex;  
 align-items: center;  
 gap: 12px;  
 flex-wrap: wrap;  
}  
  
.sort-label {  
 font-size: 14px;  
 font-weight: 600;  
 color: var(--text-main);  
 white-space: nowrap;  
}  
  
.sort-select {  
 padding: 10px 14px;  
 border-radius: var(--radius-md);  
 border: 2px solid rgba(0, 0, 0, 0.1);  
 font-family: 'Outfit', sans-serif;  
 font-size: 13px;  
 font-weight: 500;  
 background: white;  
 cursor: pointer;  
 box-shadow: var(--shadow-sm);  
 transition: var(--transition);  
 min-width: 120px;  
}  
  
.sort-select:hover,  
.sort-select:focus {  
 border-color: var(--primary);  
 box-shadow: 0 0 0 3px rgba(30, 144, 255, 0.1);  
 outline: none;  
}  
  
/\* ===== VENUE CARDS SECTION ===== \*/  
.card-grid {  
 display: grid;  
 grid-template-columns: repeat(auto-fill, minmax(250px, 1fr));  
 gap: 20px;  
 margin-top: 15px;  
 height: calc(100vh - 300px);  
 overflow-y: auto;  
 padding-right: 10px;  
}  
  
.venue-card {  
 background: var(--card-bg);  
 border-radius: var(--radius-lg);  
 padding: 20px;  
 box-shadow: var(--shadow-md);  
 text-align: center;  
 transition: var(--transition);  
 position: relative;  
 overflow: hidden;  
 border: 1px solid rgba(255, 255, 255, 0.8);  
 animation: fadeIn 0.5s ease-out forwards;  
 animation-delay: calc(var(--animation-order) \* 0.1s);  
 opacity: 0;  
 height: fit-content;  
}  
  
.venue-card::before {  
 content: '';  
 position: absolute;  
 top: 0;  
 left: 0;  
 width: 100%;  
 height: 4px;  
 background: var(--primary);  
 opacity: 0;  
 transition: var(--transition);  
}  
  
.venue-card:hover {  
 transform: translateY(-5px);  
 box-shadow: var(--shadow-lg);  
}  
  
.venue-card:hover::before {  
 opacity: 1;  
}  
  
.venue-image {  
 width: 100%;  
 height: 150px;  
 object-fit: cover;  
 border-radius: var(--radius-md);  
 transition: var(--transition);  
 box-shadow: var(--shadow-sm);  
 margin-bottom: 15px;  
}  
  
.venue-card:hover .venue-image {  
 transform: scale(1.03);  
}  
  
.venue-name {  
 font-size: 1.2rem;  
 font-weight: 600;  
 color: var(--text-main);  
 text-transform: capitalize;  
 margin-bottom: 6px;  
 letter-spacing: -0.01em;  
}  
  
.venue-sport {  
 font-size: 0.9rem;  
 color: #0ea5e9;  
 font-weight: 500;  
 margin-bottom: 10px;  
 text-transform: capitalize;  
}  
  
.venue-rating {  
 color: #ffa726;  
 margin-bottom: 15px;  
 font-size: 1rem;  
}  
  
.view-button {  
 background: var(--primary);  
 color: white;  
 border: none;  
 padding: 10px 20px;  
 border-radius: var(--radius-md);  
 cursor: pointer;  
 font-weight: 600;  
 font-size: 0.9rem;  
 font-family: 'Outfit', sans-serif;  
 transition: var(--transition);  
 box-shadow: 0 4px 12px rgba(30, 144, 255, 0.3);  
 letter-spacing: 0.02em;  
}  
  
.view-button:hover {  
 background: var(--primary);  
 opacity: 0.9;  
 box-shadow: 0 6px 20px rgba(30, 144, 255, 0.4);  
 transform: translateY(-2px);  
}  
  
/\* ===== BOOKING PAGE SECTION ===== \*/  
.booking-page {  
 display: none;  
 padding: 20px 0;  
}  
  
.booking-container {  
 display: flex;  
 gap: 24px;  
 max-width: 1200px;  
 margin: 0 auto;  
 padding: 0 20px;  
}  
  
.booking-content {  
 flex: 1;  
 background: var(--card-bg);  
 border-radius: var(--radius-lg);  
 padding: 30px;  
 box-shadow: var(--shadow-md);  
}  
  
.sidebar {  
 width: 280px;  
 background: var(--card-bg);  
 padding: 30px 25px;  
 box-shadow: var(--shadow-md);  
 height: fit-content;  
 max-height: calc(100vh - 100px);  
 overflow-y: auto;  
 position: sticky;  
 top: 80px;  
 border-right: 1px solid rgba(255, 255, 255, 0.3);  
 display: flex;  
 flex-direction: column;  
 backdrop-filter: blur(10px);  
 border-radius: var(--radius-md) 0 0 var(--radius-md);  
}  
  
.sidebar h3 {  
 font-size: 1.3rem;  
 margin-bottom: 20px;  
 font-weight: 600;  
 color: var(--text-main);  
 text-align: center;  
 padding-bottom: 15px;  
 border-bottom: 2px solid rgba(30, 144, 255, 0.1);  
}  
  
.month-btn {  
 display: block;  
 background: none;  
 border: none;  
 font-size: 1rem;  
 text-align: left;  
 padding: 14px 18px;  
 margin-bottom: 6px;  
 cursor: pointer;  
 width: 100%;  
 border-radius: var(--radius-sm);  
 transition: var(--transition);  
 color: var(--text-light);  
 font-weight: 500;  
 font-family: 'Outfit', sans-serif;  
}  
  
.month-btn:hover {  
 background-color: var(--bg-light);  
 color: var(--primary);  
 transform: translateX(4px);  
}  
  
.month-btn.active {  
 background: var(--primary);  
 color: white;  
 font-weight: 600;  
 box-shadow: 0 4px 12px rgba(30, 144, 255, 0.3);  
}  
  
#yearSelect {  
 margin-top: 20px;  
 padding: 12px 16px;  
 border-radius: var(--radius-md);  
 border: 2px solid rgba(0, 0, 0, 0.1);  
 font-family: 'Outfit', sans-serif;  
 font-size: 1rem;  
 background: white;  
 cursor: pointer;  
 box-shadow: var(--shadow-sm);  
 transition: var(--transition);  
 width: 100%;  
}  
  
/\* ===== DATA TABLE SECTION ===== \*/  
.back-btn {  
 margin-bottom: 30px;  
 background: var(--secondary);  
 color: white;  
 border: none;  
 padding: 14px 24px;  
 border-radius: var(--radius-md);  
 cursor: pointer;  
 font-weight: 600;  
 font-size: 1rem;  
 font-family: 'Outfit', sans-serif;  
 transition: var(--transition);  
 display: inline-flex;  
 align-items: center;  
 gap: 10px;  
 box-shadow: 0 4px 12px rgba(34, 197, 94, 0.3);  
}  
  
.back-btn:hover {  
 background: var(--secondary);  
 opacity: 0.9;  
 box-shadow: 0 6px 20px rgba(34, 197, 94, 0.4);  
 transform: translateY(-2px);  
}  
  
h2 {  
 font-size: 2.2rem;  
 margin-bottom: 25px;  
 font-weight: 700;  
 color: var(--text-main);  
 letter-spacing: -0.02em;  
}  
  
h3 {  
 font-size: 1.5rem;  
 margin-bottom: 1rem;  
 font-weight: 600;  
 color: var(--primary-dark);  
}  
  
.day-button-group {  
 display: flex;  
 flex-wrap: wrap;  
 gap: 12px;  
 margin-bottom: 30px;  
 padding: 20px;  
 background: var(--card-bg);  
 border-radius: var(--radius-md);  
 box-shadow: var(--shadow-sm);  
}  
  
.day-button-group .month-btn {  
 font-size: 0.9rem;  
 padding: 10px 16px;  
 margin-bottom: 0;  
 border-radius: var(--radius-sm);  
 background: white;  
 border: 2px solid rgba(30, 144, 255, 0.2);  
}  
  
#pdfContent {  
 background: var(--card-bg);  
 padding: 30px;  
 border-radius: var(--radius-md);  
 box-shadow: var(--shadow-md);  
 margin-bottom: 20px;  
}  
  
table {  
 width: 100%;  
 border-collapse: separate;  
 border-spacing: 0;  
 box-shadow: var(--shadow-md);  
 border-radius: var(--radius-md);  
 overflow: hidden;  
 background: white;  
}  
  
th,  
td {  
 padding: 18px 24px;  
 text-align: center;  
}  
  
th {  
 background: var(--bg-light);  
 color: var(--text-main);  
 font-weight: 700;  
 text-transform: uppercase;  
 font-size: 0.85rem;  
 letter-spacing: 1px;  
 border-bottom: 3px solid var(--primary);  
}  
  
td {  
 background: white;  
 border-bottom: 1px solid rgba(0, 0, 0, 0.05);  
 font-weight: 500;  
}  
  
tr:last-child td {  
 border-bottom: none;  
}  
  
tr:hover td {  
 background-color: rgba(30, 144, 255, 0.05);  
}  
  
.summary {  
 margin-top: 30px;  
 font-weight: 600;  
 background: var(--card-bg);  
 padding: 25px 30px;  
 border-radius: var(--radius-md);  
 box-shadow: var(--shadow-md);  
 color: var(--text-main);  
 font-size: 1.1rem;  
 display: flex;  
 justify-content: space-between;  
 flex-wrap: wrap;  
 gap: 20px;  
 border-left: 4px solid var(--primary);  
}  
  
.no-data {  
 text-align: center;  
 padding: 40px 20px;  
 color: var(--text-light);  
 font-style: italic;  
 font-size: 1.1rem;  
}  
  
/\* ===== ANIMATIONS ===== \*/  
@keyframes fadeIn {  
 from {  
 opacity: 0;  
 transform: translateY(20px);  
 }  
 to {  
 opacity: 1;  
 transform: translateY(0);  
 }  
}  
  
.venue-card:nth-child(1) { --animation-order: 1; }  
.venue-card:nth-child(2) { --animation-order: 2; }  
.venue-card:nth-child(3) { --animation-order: 3; }  
.venue-card:nth-child(4) { --animation-order: 4; }  
  
/\* ===== RESPONSIVE DESIGN ===== \*/  
@media (max-width: 1024px) {  
 .card-grid {  
 grid-template-columns: repeat(auto-fill, minmax(220px, 1fr));  
 gap: 15px;  
 }  
   
 .sidebar {  
 width: 200px;  
 }  
}  
  
@media (max-width: 768px) {  
 .container {  
 padding: 10px 15px;  
 }  
  
 #cardSection {  
 height: auto;  
 overflow: visible;  
 }  
  
 .search-sort-container {  
 flex-direction: column;  
 align-items: stretch;  
 gap: 10px;  
 padding: 12px;  
 }  
  
 .search-bar {  
 max-width: none;  
 }  
  
 .sort-container {  
 justify-content: space-between;  
 width: 100%;  
 }  
  
 .sort-select {  
 flex: 1;  
 min-width: 0;  
 }  
  
 .booking-page {  
 flex-direction: column;  
 height: auto;  
 }  
  
 .sidebar {  
 width: 100%;  
 height: auto;  
 position: relative;  
 display: flex;  
 flex-direction: row;  
 overflow-x: auto;  
 padding: 20px;  
 gap: 10px;  
 align-items: center;  
 border-radius: var(--radius-md);  
 top: 0;  
 }  
  
 .sidebar h3 {  
 display: none;  
 }  
  
 .month-btn {  
 padding: 10px 16px;  
 margin-bottom: 0;  
 white-space: nowrap;  
 flex-shrink: 0;  
 }  
  
 .page-title {  
 font-size: 22px;  
 }  
  
 .page-header {  
 padding: 15px;  
 }  
  
 .card-grid {  
 height: auto;  
 overflow-y: visible;  
 }  
  
 #pdfContent {  
 height: auto;  
 overflow-y: visible;  
 }  
}  
  
@media (max-width: 576px) {  
 .main-content {  
 padding: 10px;  
 }  
  
 .card-grid {  
 grid-template-columns: 1fr;  
 }  
  
 .venue-card {  
 padding: 15px;  
 }  
  
 th,  
 td {  
 padding: 10px 8px;  
 font-size: 0.8rem;  
 }  
  
 .summary {  
 flex-direction: column;  
 align-items: flex-start;  
 gap: 10px;  
 padding: 15px;  
 }  
  
 .page-title {  
 font-size: 20px;  
 }  
  
 .nav-links {  
 gap: 6px;  
 }  
  
 .nav-links li a {  
 padding: 6px 10px;  
 font-size: 11px;  
 }  
}

————————————————————————————————————————

## Analytics\Analytics.html

<!DOCTYPE html>  
<html lang="en">  
<head>  
 <meta charset="UTF-8" />  
 <meta name="viewport" content="width=device-width, initial-scale=1.0">  
 <title>Venue Analytics | Book&Play</title>  
 <script src="https://cdnjs.cloudflare.com/ajax/libs/html2pdf.js/0.9.2/html2pdf.bundle.min.js"></script>  
 <link rel="preconnect" href="https://fonts.googleapis.com">  
 <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>  
 <link href="https://fonts.googleapis.com/css2?family=Outfit:wght@300;400;500;600;700&display=swap" rel="stylesheet">  
 <link rel="stylesheet" href="Analytics.css">  
</head>  
<body>  
  
  
<!-- Page Content -->  
<div class="container">  
 <!-- Page Header Section -->  
 <div class="page-header">  
 <h1 class="page-title">Analytics My Venues</h1>  
 <p class="page-subtitle">View and control all your venue reservations with clarity and ease</p>  
 </div>  
  
 <!-- Search and Sort Section -->  
 <div class="search-sort-container">  
 <div class="search-bar">  
 <span class="search-icon">🔍</span>  
 <input type="text" id="searchInput" placeholder="Search venues by name">  
 </div>  
   
 <div class="sort-container">  
 <span class="sort-label">Sort By:</span>  
 <select class="sort-select" id="sortSelect">  
 <option value="name">Name (A-Z)</option>  
 <option value="name-desc">Name (Z-A)</option>  
 </select>  
 <select class="sort-select" id="sportSelect">  
 <option value="all">All Sport</option>  
 </select>  
 </div>  
 </div>  
  
 <!-- Venue Cards Section -->  
 <div id="cardSection" class="main-content">  
 <div class="card-grid" id="venueCards">  
 <!-- Venue cards will be injected here by JavaScript -->  
 </div>  
 </div>  
  
 <!-- Booking Details Section -->  
 <div id="bookingPage" class="container booking-page" style="display:none;">  
 <!-- Sidebar Navigation -->  
 <div class="sidebar">  
 <h3>🗓 Months</h3>  
 <button class="month-btn" onclick="showMonth('January')">January</button>  
 <button class="month-btn" onclick="showMonth('February')">February</button>  
 <button class="month-btn" onclick="showMonth('March')">March</button>  
 <button class="month-btn" onclick="showMonth('April')">April</button>  
 <button class="month-btn" onclick="showMonth('May')">May</button>  
 <button class="month-btn" onclick="showMonth('June')">June</button>  
 <button class="month-btn" onclick="showMonth('July')">July</button>  
 <button class="month-btn" onclick="showMonth('August')">August</button>  
 <button class="month-btn" onclick="showMonth('September')">September</button>  
 <button class="month-btn" onclick="showMonth('October')">October</button>  
 <button class="month-btn" onclick="showMonth('November')">November</button>  
 <button class="month-btn" onclick="showMonth('December')">December</button>  
 <button class="month-btn" onclick="showMonth('All')">All</button>  
   
 <select id="yearSelect" onchange="showMonth(currentMonth)">  
 <option>2025</option>  
 <option>2024</option>  
 <option>2023</option>  
 </select>  
 </div>  
  
 <!-- Main Content Area -->  
 <div class="main-content">  
 <button class="back-btn" onclick="goBack()">← Back to Venues</button>  
   
 <h2 id="venueTitle" style="margin-top: 30px;">Bookings for ...</h2>  
   
 <div id="dayButtons" class="day-button-group" style="margin-top: 20px;"></div>  
  
 <!-- Data Table Container -->  
 <div id="pdfContent">  
 <table>  
 <thead>  
 <tr>  
 <th>Date 📅</th>  
 <th>Bookings 👥</th>  
 <th>Total (₪) 💰</th>  
 </tr>  
 </thead>  
 <tbody id="dailyBookingTable"></tbody>  
 </table>  
   
 <div class="summary" id="summaryRow"></div>  
 </div>  
   
 <button class="view-button" onclick="downloadPDF()" style="margin-top: 20px;">📄 Download as PDF</button>  
 </div>  
 </div>  
</div>  
  
<!-- JS -->  
<script src="Analytics.js" defer></script>  
  
</body>  
</html>

————————————————————————————————————————

## Analytics\Analytics.js

// 🌐 Global state variables  
let facilities = [];  
let currentSearch = '';  
let currentSort = 'name';  
let currentSport = 'all';  
let currentVenue = '';  
let currentMonth = '';  
let venueAnalyticsData = {};  
  
// 📥 Load all venues from AnalyticsAPI  
function loadFacilities() {  
 fetch('AnalyticsAPI.php?action=fetch\_venues')  
 .then(response => response.json())  
 .then(data => {  
 if (!Array.isArray(data.venues)) {  
 console.error('Invalid data format:', data);  
 return;  
 }  
 facilities = data.venues;  
 applyFilters();  
 })  
 .catch(error => {  
 console.error('Error loading facilities:', error);  
 const venueGrid = document.getElementById('venueCards');  
 if (venueGrid) {  
 venueGrid.innerHTML = '<div class="no-venues-message">Failed to load venues.</div>';  
 }  
 });  
}  
  
// 🖼️ Display venue cards in the UI  
function displayVenues(data) {  
 const grid = document.getElementById('venueCards');  
 grid.innerHTML = '';  
  
 if (data.length === 0) {  
 grid.innerHTML = '<div class="no-venues-message">No venues match your filters.</div>';  
 return;  
 }  
  
 data.forEach(facility => {  
 const card = document.createElement('div');  
 card.className = 'venue-card';  
 card.innerHTML = `  
 <img src="${facility.image\_url || 'default.jpg'}" class="venue-image" />  
 <div class="venue-name">${facility.place\_name}</div>  
 <div class="venue-sport">${facility.SportCategory}</div>  
 <div class="venue-rating">📍 ${facility.location || 'Unknown location'}</div>  
 <button class="view-button" onclick="openBookings('${facility.place\_name}')">View Analytics</button>  
 `;  
 grid.appendChild(card);  
 });  
}  
  
// 🧠 Apply filters and sorting on venue data  
function applyFilters() {  
 let result = facilities;  
 if (currentSearch.trim()) {  
 result = result.filter(f => f.place\_name.toLowerCase().startsWith(currentSearch.toLowerCase()));  
 }  
 if (currentSport !== 'all') {  
 result = result.filter(f => f.SportCategory.toLowerCase() === currentSport.toLowerCase());  
 }  
 if (currentSort === 'name') {  
 result.sort((a, b) => a.place\_name.localeCompare(b.place\_name));  
 } else if (currentSort === 'name-desc') {  
 result.sort((a, b) => b.place\_name.localeCompare(a.place\_name));  
 }  
 displayVenues(result);  
}  
  
// 🚀 Initialize on page load  
window.onload = () => {  
 loadFacilities();  
 loadSports();  
  
 // 🔎 Handle user input and filters  
 document.getElementById('searchInput').addEventListener('input', e => {  
 currentSearch = e.target.value;  
 applyFilters();  
 });  
 document.getElementById('sortSelect').addEventListener('change', e => {  
 currentSort = e.target.value;  
 applyFilters();  
 });  
 document.getElementById('sportSelect').addEventListener('change', e => {  
 currentSport = e.target.value;  
 applyFilters();  
 });  
};  
  
// 🎯 Load all sports categories from API  
function loadSports() {  
 fetch('AnalyticsAPI.php?action=fetch\_sports')  
 .then(response => response.json())  
 .then(data => {  
 const sportSelect = document.getElementById('sportSelect');  
 data.sports.forEach(sport => {  
 const option = document.createElement('option');  
 option.value = sport.sport\_name.toLowerCase();  
 option.textContent = sport.sport\_name;  
 sportSelect.appendChild(option);  
 });  
 })  
 .catch(error => console.error('Error loading sports:', error));  
}  
  
// 📊 Open analytics view for a selected venue  
function openBookings(venueName) {  
 const year = document.getElementById("yearSelect").value;  
 currentVenue = venueName;  
 currentMonth = 'January';  
  
 document.getElementById("venueTitle").textContent = `Bookings for ${venueName}`;  
 document.getElementById("bookingPage").style.display = "flex";  
 document.getElementById("cardSection").style.display = "none";  
  
 fetchMonthlyData(venueName, year);  
}  
  
// 📅 Fetch monthly analytics from API  
function fetchMonthlyData(venueName, year) {  
 const url = `AnalyticsAPI.php?action=fetch\_monthly\_analytics&venue=${encodeURIComponent(venueName)}&year=${year}`;  
 fetch(url)  
 .then(res => res.json())  
 .then(data => {  
 venueAnalyticsData = data.data || {};  
 populateMonthButtons(Object.keys(venueAnalyticsData));  
 showMonth("January");  
 })  
 .catch(err => {  
 console.error("❌ Error fetching monthly data:", err);  
 });  
}  
  
// 📆 Show or hide month buttons depending on available data  
function populateMonthButtons(months) {  
 document.querySelectorAll('.month-btn').forEach(btn => {  
 if (btn.textContent !== 'All') {  
 btn.style.display = months.includes(btn.textContent) ? 'block' : 'none';  
 }  
 });  
}  
  
// 📈 Show table data for selected month  
function showMonth(month) {  
 currentMonth = month;  
 const tableBody = document.getElementById("dailyBookingTable");  
 const summaryRow = document.getElementById("summaryRow");  
 tableBody.innerHTML = '';  
  
 if (month === 'All') {  
 let monthTotals = {};  
  
 Object.entries(venueAnalyticsData).forEach(([monthName, entries]) => {  
 let monthlyBookings = 0;  
 let monthlyRevenue = 0;  
 entries.forEach(entry => {  
 monthlyBookings += entry.bookings;  
 monthlyRevenue += entry.total;  
 });  
 monthTotals[monthName] = {  
 bookings: monthlyBookings,  
 total: monthlyRevenue  
 };  
 });  
  
 Object.entries(monthTotals).forEach(([monthName, summary]) => {  
 tableBody.innerHTML += `  
 <tr>  
 <td>${monthName}</td>  
 <td>${summary.bookings}</td>  
 <td>${summary.total.toFixed(2)} ₪</td>  
 </tr>`;  
 });  
  
 const totalBookings = Object.values(monthTotals).reduce((sum, m) => sum + m.bookings, 0);  
 const totalRevenue = Object.values(monthTotals).reduce((sum, m) => sum + m.total, 0);  
  
 summaryRow.innerHTML = `  
 <span>Total Bookings: ${totalBookings}</span>  
 <span>Total Revenue: ₪${totalRevenue.toFixed(2)}</span>  
 `;  
 } else {  
 const data = venueAnalyticsData[month] || [];  
 let totalBookings = 0;  
 let totalRevenue = 0;  
  
 data.forEach(entry => {  
 tableBody.innerHTML += `  
 <tr>  
 <td>${entry.date}</td>  
 <td>${entry.bookings}</td>  
 <td>${entry.total} ₪</td>  
 </tr>`;  
 totalBookings += entry.bookings;  
 totalRevenue += entry.total;  
 });  
  
 summaryRow.innerHTML = `  
 <span>Total Bookings: ${totalBookings}</span>  
 <span>Total Revenue: ₪${totalRevenue.toFixed(2)}</span>  
 `;  
 }  
  
 // Highlight selected month button  
 document.querySelectorAll('.month-btn').forEach(btn => btn.classList.remove('active'));  
 const activeBtn = Array.from(document.querySelectorAll('.month-btn')).find(b => b.textContent === month);  
 if (activeBtn) activeBtn.classList.add('active');  
}  
  
// ⬅️ Back to venue cards view  
function goBack() {  
 document.getElementById("bookingPage").style.display = "none";  
 document.getElementById("cardSection").style.display = "block";  
}  
  
// 🧾 Export bookings data to PDF  
function downloadPDF() {  
 const element = document.getElementById("pdfContent");  
 const opt = {  
 margin: 0.5,  
 filename: 'bookings-report.pdf',  
 image: { type: 'jpeg', quality: 0.98 },  
 html2canvas: { scale: 2 },  
 jsPDF: { unit: 'in', format: 'letter', orientation: 'portrait' }  
 };  
 html2pdf().set(opt).from(element).save();  
}  
  
// Expose function to global scope  
window.openBookings = openBookings;

————————————————————————————————————————

## Analytics\Analytics.php

<?php  
include '../navbar.php';  
include 'Analytics.html';  
exit;  
?>

————————————————————————————————————————

## Analytics\AnalyticsAPI.php

<?php  
session\_start();  
require\_once '../../../db.php';  
header('Content-Type: application/json');  
  
// Get the requested action from the frontend  
$action = $\_GET['action'] ?? $\_POST['action'] ?? '';  
  
switch ($action) {  
  
 // 📊 1. Fetch monthly booking analytics per facility  
 case 'fetch\_monthly\_analytics':  
 $facilities\_id = intval($\_GET['facilities\_id'] ?? 0);  
  
 if ($facilities\_id <= 0) {  
 echo json\_encode(['success' => false, 'message' => 'Invalid facility ID.']);  
 exit;  
 }  
  
 $stmt = $conn->prepare("  
 SELECT booking\_day, booking\_month, booking\_year, total\_bookings   
 FROM facilityinsights   
 WHERE facilities\_id = ?  
 ORDER BY booking\_year DESC, booking\_month DESC, booking\_day DESC  
 ");  
 $stmt->bind\_param("i", $facilities\_id);  
 $stmt->execute();  
 $result = $stmt->get\_result();  
  
 $analytics = [];  
 while ($row = $result->fetch\_assoc()) {  
 $analytics[] = $row;  
 }  
  
 echo json\_encode(['success' => true, 'data' => $analytics]);  
 break;  
  
 // 🏅 2. Fetch all accepted sports  
 case 'fetch\_sports':  
 $sports = [];  
 $sql = "SELECT \* FROM sports WHERE is\_Accepted = 1";  
 $result = $conn->query($sql);  
  
 if ($result && $result->num\_rows > 0) {  
 while ($row = $result->fetch\_assoc()) {  
 $sports[] = $row;  
 }  
 }  
  
 echo json\_encode(['success' => true, 'sports' => $sports]);  
 break;  
  
 // 🏟️ 3. Fetch all accepted venues  
 case 'fetch\_venues':  
 $venues = [];  
 $sql = "SELECT \* FROM sportfacilities WHERE is\_Accepted = 1";  
 $result = $conn->query($sql);  
  
 if ($result && $result->num\_rows > 0) {  
 while ($row = $result->fetch\_assoc()) {  
 $row['image\_url'] = explode(",", $row['image\_url'])[0]; // Return only first image  
 $venues[] = $row;  
 }  
 }  
  
 echo json\_encode(['success' => true, 'venues' => $venues]);  
 break;  
  
 // ⛔ Handle unknown action  
 default:  
 echo json\_encode(['success' => false, 'message' => 'Invalid or missing action.']);  
 break;  
}  
  
$conn->close();  
?>

————————————————————————————————————————

## Bookings\add\_booking.php

<?php  
session\_start();  
require\_once '../../../db.php';  
header('Content-Type: application/json');  
  
$username = $\_POST['username'] ?? '';  
$date = $\_POST['date'] ?? '';  
$players = intval($\_POST['players'] ?? 0);  
$start\_time = $\_POST['start\_time'] ?? '';  
$end\_time = $\_POST['end\_time'] ?? '';  
$facilityId = intval($\_POST['facility\_id'] ?? 0);  
  
if (!$username || !$date || !$players || !$start\_time || !$end\_time || !$facilityId) {  
 echo json\_encode(['success' => false, 'message' => 'Missing fields']);  
 exit;  
}  
  
  
// ✅ Check if user exists  
$userCheck = $conn->prepare("SELECT 1 FROM users WHERE username = ?");  
$userCheck->bind\_param("s", $username);  
$userCheck->execute();  
$userCheck->store\_result();  
  
if ($userCheck->num\_rows === 0) {  
 echo json\_encode(['success' => false, 'message' => 'Username not found in database.']);  
 exit;  
}  
  
// ✅ Prevent past dates  
if (strtotime($date) < strtotime(date('Y-m-d'))) {  
 echo json\_encode(['success' => false, 'message' => 'Date must be today or in the future.']);  
 exit;  
}  
  
// ✅ Prevent booking in past hours of same day  
if ($date === date('Y-m-d')) {  
 $now = date('H:i');  
 if (strtotime($end) <= strtotime($now)) {  
 echo json\_encode(['success' => false, 'message' => 'Cannot book for past hours today.']);  
 exit;  
 }  
}  
  
// ✅ Check for existing booking at same time  
$conflictCheck = $conn->prepare("  
 SELECT 1 FROM bookings  
 WHERE facilities\_id = ? AND booking\_date = ?   
 AND (  
 (start\_time < ? AND end\_time > ?) OR  
 (start\_time < ? AND end\_time > ?) OR  
 (start\_time >= ? AND end\_time <= ?)  
 )  
");  
$conflictCheck->bind\_param("isssssss", $facilityId, $date, $end\_time, $end\_time, $start\_time, $start\_time, $start\_time, $end\_time);  
$conflictCheck->execute();  
$conflictCheck->store\_result();  
  
if ($conflictCheck->num\_rows > 0) {  
 echo json\_encode(['success' => false, 'message' => 'This time slot is already booked.']);  
 exit;  
}  
$start = DateTime::createFromFormat('H:i', $start\_time);  
$end = DateTime::createFromFormat('H:i', $end\_time);  
$duration\_hours = ($end->getTimestamp() - $start->getTimestamp()) / 3600;  
  
// Extract price from database  
$facility\_id = $\_POST['facility\_id'];  
$stmt = $conn->prepare("SELECT price FROM sportfacilities WHERE facilities\_id = ?");  
$stmt->bind\_param("i", $facility\_id);  
$stmt->execute();  
$result = $stmt->get\_result();  
$price = $result->fetch\_assoc()['price'] ?? 0;  
  
$total\_price = $price \* $duration\_hours;  
// ✅ Save booking  
$stmt = $conn->prepare("  
 INSERT INTO bookings (facilities\_id, username, booking\_date, start\_time, end\_time, status,Total\_Price)   
 VALUES (?, ?, ?, ?, ?, 'pending',?)  
");  
$stmt->bind\_param("issssi", $facilityId, $username, $date, $start\_time, $end\_time, $total\_price);  
  
if ($stmt->execute()) {  
 $bookingId = $conn->insert\_id; // ✅ here you have booking\_id  
  
 if ($players > 1) {  
 $groupName = $username . "'s Group " . date("Ymd\_His");  
 $description = "Auto-created group for booking on $date at $start\_time";  
 $maxMembers = $players;  
  
 $groupStmt = $conn->prepare("  
 INSERT INTO groups (group\_name, facilities\_id, description, created\_by, max\_members, booking\_id)  
 VALUES (?, ?, ?, ?, ?, ?)  
 ");  
 $groupStmt->bind\_param("sissii", $groupName, $facilityId, $description, $username, $maxMembers, $bookingId);  
  
 if ($groupStmt->execute()) {  
 $groupId = $conn->insert\_id;  
 $memberStmt = $conn->prepare("INSERT INTO group\_members (group\_id, username) VALUES (?, ?)");  
 $memberStmt->bind\_param("is", $groupId, $username);  
 $memberStmt->execute();  
 }  
 }  
  
 echo json\_encode(['success' => true, 'message' => 'Booking and group (if needed) saved!']);  
} else {  
 echo json\_encode(['success' => false, 'message' => $conn->error]);  
}  
?>

————————————————————————————————————————

## Bookings\BookedFacilitiesFetch.php

<?php  
session\_start();  
require\_once '../../../db.php';  
header('Content-Type: application/json');  
  
if (!isset($\_SESSION['user\_id'])) {  
 echo json\_encode(['success' => false, 'message' => 'User not logged in']);  
 exit;  
}  
  
$user\_name = $\_SESSION['user\_id'];  
  
$query = "  
 SELECT   
 sf.facilities\_id,   
 sf.place\_name,   
 sf.location,   
 sf.image\_url,   
 sf.SportCategory,  
 ROUND(AVG(r.rating\_value), 1) AS avg\_rating  
FROM sportfacilities sf  
JOIN bookings b ON sf.facilities\_id = b.facilities\_id  
LEFT JOIN ratings r ON sf.facilities\_id = r.facilities\_id AND r.rating\_value > 0  
WHERE sf.owner\_username = ? AND sf.is\_Accepted = 1  
GROUP BY sf.facilities\_id  
  
";  
  
$stmt = $conn->prepare($query);  
$stmt->bind\_param("s", $user\_name);  
$stmt->execute();  
$result = $stmt->get\_result();  
  
$venues = [];  
  
while ($row = $result->fetch\_assoc()) {  
 $venues[] = [  
 'id' => $row['facilities\_id'],  
 'name' => $row['place\_name'],  
 'location' => $row['location'],  
 'image' => $row['image\_url'],  
 'sport' => $row['SportCategory'],  
 'rating' => $row['avg\_rating'] ?? 'N/A'  
 ];  
}  
  
echo json\_encode(['success' => true, 'venues' => $venues]);  
?>

————————————————————————————————————————

## Bookings\Bookings.css

/\* Global Styles & Variables \*/  
  
/\* Main Content Layout \*/  
main {  
 padding: 30px;  
 max-width: 1400px;  
 margin: 0 auto;  
}  
  
.section-header {  
 display: flex;  
 justify-content: space-between;  
 align-items: center;  
 margin-bottom: 24px;  
}  
  
h1 {  
 font-size: 24px;  
 font-weight: 700;  
 color: var(--text-main);  
}  
  
.filters {  
 display: flex;  
 gap: 12px;  
}  
  
select {  
 padding: 12px 16px;  
 border-radius: var(--radius-md);  
 border: 1px solid rgba(0, 0, 0, 0.1);  
 font-family: 'Outfit', sans-serif;  
 font-size: 1rem;  
 background: white;  
 cursor: pointer;  
 box-shadow: var(--shadow-sm);  
 transition: var(--transition);  
}  
  
select:hover, select:focus {  
 border-color: var(--primary);  
 box-shadow: 0 0 0 3px rgba(58, 134, 255, 0.2);  
 outline: none;  
}  
  
/\* Venue Cards View \*/  
  
.search-sort-container {  
 display: flex;  
 justify-content: space-between;  
 gap: 20px;  
 margin-bottom: 30px;  
 align-items: flex-end; /\* Align items to the bottom \*/  
}  
  
.search-bar {  
 flex: 1;  
 position: relative;  
 margin-top: 20px;  
}  
.search-bar input {  
 width: 100%;  
 padding: 12px 15px 12px 40px;  
 border-radius: var(--radius);  
 border: 1px solid var(--card-border);  
 background-color: white;  
 font-size: 14px;  
}  
.search-icon {  
 position: absolute;  
 left: 15px;  
 top: 50%;  
 transform: translateY(-50%);  
 color: var(--text-muted);  
}  
  
/\* Sort By section \*/  
.sort-section {  
 display: flex;  
 align-items: center;  
 gap: 10px;  
 margin-bottom: 2px; /\* Adjust this to fine-tune vertical alignment \*/  
}  
  
.sort-section label {  
 font-weight: 500;  
 color: var(--text-main);  
}  
  
.venue-card {  
 background: var(--card-bg);  
 border-radius: 20px;  
 box-shadow: var(--shadow-md);  
 padding: 20px;  
 text-align: center;  
 width: 260px;  
 margin: 0;  
 transition: var(--transition);  
 display: flex;  
 flex-direction: column;  
 align-items: center;  
}  
  
  
.venue-card::before {  
 content: '';  
 position: absolute;  
 top: 0;  
 left: 0;  
 width: 100%;  
 height: 5px;  
 background: var(--primary);  
 opacity: 0;  
 transition: var(--transition);  
}  
  
  
.venue-card:hover {  
 transform: translateY(-5px);  
 box-shadow: var(--shadow-lg);  
}  
  
.venue-card:hover::before {  
 opacity: 1;  
}  
.venue-image {  
 width: 100%;  
 height: 160px;  
 object-fit: cover;  
 border-radius: 16px;  
 margin-bottom: 15px;  
 box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);  
}  
  
.venue-card:nth-child(1) { --animation-order: 1; }  
.venue-card:nth-child(2) { --animation-order: 2; }  
.venue-card:nth-child(3) { --animation-order: 3; }  
.venue-card:nth-child(4) { --animation-order: 4; }  
  
.venue-header {  
 display: flex;  
 justify-content: space-between;  
 align-items: center;  
 margin-bottom: 16px;  
}  
  
  
  
.venue-rating {  
 font-size: 1.1rem;  
 font-weight: 500;  
 color: gold;  
 margin-bottom: 10px;  
}  
.venue-name {  
 font-size: 1.2rem;  
 font-weight: 600;  
 color: var(--text-main);  
 margin-bottom: 5px;  
 text-transform: capitalize;  
  
}  
.venue-location {  
 font-size: 0.95rem;  
 color: var(--text-light);  
 margin-bottom: 10px;  
}  
.venue-sport {  
 font-size: 0.95rem;  
 font-weight: 600;  
 color: var(--text-main);  
 margin-bottom: 20px;  
 text-transform: capitalize;  
  
}  
.venue-stats {  
 display: grid;  
 grid-template-columns: repeat(3, 1fr);  
 gap: 12px;  
 margin-bottom: 20px;  
}  
  
.stat {  
 display: flex;  
 flex-direction: column;  
}  
  
  
.stat-value {  
 font-weight: 600;  
}  
  
.btn-view-bookings {  
 background: var(--primary);  
 color: white;  
 border: none;  
 padding: 10px 20px;  
 border-radius: 12px;  
 font-size: 1rem;  
 font-weight: 500;  
 box-shadow: 0 4px 12px rgba(30, 144, 255, 0.3);  
 cursor: pointer;  
 transition: 0.3s ease;  
}  
.btn-view-bookings:hover {  
 transform: scale(1.05);  
 background: var(--primary);  
 opacity: 0.95;  
}  
.venues-container {  
 display: flex;  
 flex-wrap: wrap;  
 justify-content: flex-start; /\* or 'space-between' \*/  
 gap: 16px; /\* reduce this from 24px or more \*/  
 margin-top: 30px;  
}  
/\* Booking Detail View \*/  
.hidden {  
 display: none;  
}  
  
.booking-container {  
 display: flex;  
 gap: 24px;  
 min-height: 600px;  
}  
  
.venues-sidebar {  
 width: 240px;  
 background: var(--card-bg);  
 padding: 25px 20px;  
 box-shadow: var(--shadow-md);  
 border-radius: var(--radius-md);  
 border-right: 1px solid rgba(255, 255, 255, 0.3);  
 display: flex;  
 flex-direction: column;  
 backdrop-filter: blur(10px);  
}  
  
.venues-sidebar h3 {  
 margin-bottom: 16px;  
 font-size: 16px;  
}  
  
.venues-list {  
 list-style: none;  
 margin-bottom: 24px;  
}  
  
.venues-list li {  
 display: flex;  
 align-items: center;  
 padding: 12px 16px;  
 border-radius: var(--radius-sm);  
 margin-bottom: 8px;  
 cursor: pointer;  
 transition: var(--transition);  
 font-size: 14px;  
 color: var(--text-light);  
}  
  
.venues-list li:hover {  
 background-color: var(--bg-light);  
 color: var(--primary);  
}  
  
.venues-list li.active {  
 background: var(--primary);  
 color: white;  
 font-weight: 500;  
 box-shadow: 0 4px 10px rgba(30, 144, 255, 0.3);  
}  
  
  
  
.btn-back-to-venues {  
 margin-top: auto;  
 background: var(--secondary);  
 color: white;  
 border: none;  
 padding: 12px 20px;  
 border-radius: var(--radius-md);  
 cursor: pointer;  
 font-weight: 500;  
 font-size: 1rem;  
 transition: var(--transition);  
 display: inline-flex;  
 align-items: center;  
 justify-content: center;  
 gap: 8px;  
 box-shadow: 0 4px 10px rgba(34, 197, 94, 0.3);  
}  
  
.btn-back-to-venues:hover {  
 background: var(--secondary);  
 opacity: 0.9;  
 box-shadow: 0 7px 15px rgba(34, 197, 94, 0.4);  
 transform: translateY(-2px);  
}  
  
.booking-content {  
 flex: 1;  
 background: var(--card-bg);  
 border-radius: var(--radius-lg);  
 padding: 24px;  
 box-shadow: var(--shadow-md);  
}  
  
.venue-details {  
 margin-bottom: 24px;  
}  
  
.venue-meta {  
 display: flex;  
 gap: 12px;  
 margin-top: 4px;  
}  
  
.location-badge, .rating-badge {  
 display: inline-flex;  
 align-items: center;  
 padding: 4px 8px;  
 border-radius: var(--radius-sm);  
 font-size: 12px;  
}  
  
.location-badge {  
 background-color: var(--bg-light);  
 color: var(--text-main);  
}  
  
.rating-badge {  
 background-color: #fffaf0;  
 color: #dd6b20;  
}  
  
.booking-controls {  
 margin-bottom: 24px;  
}  
  
.month-selector {  
 display: flex;  
 align-items: center;  
 margin-bottom: 16px;  
 justify-content: center;  
}  
  
.month-nav {  
 background: none;  
 border: 1px solid rgba(0, 0, 0, 0.1);  
 color: var(--text-light);  
 width: 32px;  
 height: 32px;  
 border-radius: 50%;  
 cursor: pointer;  
 display: flex;  
 align-items: center;  
 justify-content: center;  
 transition: var(--transition);  
}  
  
.month-nav:hover {  
 background-color: var(--bg-light);  
 color: var(--primary);  
}  
  
.current-month {  
 font-weight: 700;  
 margin: 0 16px;  
 min-width: 120px;  
 text-align: center;  
}  
  
.dates-container {  
 display: flex;  
 gap: 8px;  
 overflow-x: auto;  
 padding-bottom: 8px;  
}  
  
.date-btn {  
 padding: 8px 12px;  
 background-color: white;  
 border: 1px solid rgba(0, 0, 0, 0.1);  
 border-radius: var(--radius-sm);  
 font-size: 14px;  
 cursor: pointer;  
 transition: var(--transition);  
 white-space: nowrap;  
}  
  
.date-btn.active {  
 background: var(--primary);  
 color: white;  
 font-weight: 500;  
 border-color: var(--primary);  
 box-shadow: 0 4px 10px rgba(30, 144, 255, 0.3);  
}  
  
.date-btn:hover {  
 background-color: var(--bg-light);  
}  
  
.bookings-table-container {  
 overflow-x: auto;  
}  
  
.bookings-table {  
 width: 100%;  
 border-collapse: separate;  
 border-spacing: 0;  
 margin-top: 25px;  
 box-shadow: var(--shadow-md);  
 border-radius: var(--radius-md);  
 overflow: hidden;  
}  
  
.bookings-table th {  
 background: var(--bg-light);  
 color: var(--text-main);  
 font-weight: 600;  
 text-transform: uppercase;  
 font-size: 0.9rem;  
 letter-spacing: 1px;  
 padding: 16px 20px;  
 text-align: center;  
}  
  
.bookings-table td {  
 background: var(--card-bg);  
 border-bottom: 1px solid rgba(0, 0, 0, 0.05);  
 padding: 16px 20px;  
 text-align: center;  
}  
  
.bookings-table tr:last-child td {  
 border-bottom: none;  
}  
  
.bookings-table tr:hover td {  
 background-color: rgba(58, 134, 255, 0.05);  
}  
  
.user-column {  
 display: flex;  
 align-items: center;  
 justify-content: center;  
}  
  
.user-avatar-small {  
 width: 30px;  
 height: 30px;  
 border-radius: 50%;  
 margin-right: 12px;  
}  
  
.status {  
 display: inline-block;  
 padding: 4px 8px;  
 border-radius: var(--radius-sm);  
 font-size: 12px;  
 font-weight: 600;  
}  
  
.btn-add-booking {  
 background: linear-gradient(to right, #1e90ff, #22c55e); /\* Gradient Blue to Green \*/  
 color: white;  
 border: none;  
 padding: 10px 20px;  
 font-size: 16px;  
 border-radius: 12px;  
 box-shadow: 0 4px 12px rgba(30, 144, 255, 0.2);  
 cursor: pointer;  
 transition: 0.3s ease;  
}  
/\* Modal overlay \*/  
.modal-overlay {  
 display: none; /\* ⛔ Hide on page load \*/  
 position: fixed;  
 z-index: 1000;  
 left: 0; top: 0;  
 width: 100%;  
 height: 100%;  
 background: rgba(15, 23, 42, 0.5); /\* Soft overlay \*/  
 justify-content: center;  
 align-items: center;  
}  
  
/\* Modal box \*/  
.modal-content {  
 background: #ffffff;  
 border-radius: 16px;  
 padding: 30px 25px;  
 width: 400px;  
 box-shadow: 0 10px 25px rgba(0, 0, 0, 0.2);  
 font-family: 'Outfit', sans-serif;  
}  
  
.modal-content h3 {  
 margin-top: 0;  
 margin-bottom: 20px;  
 font-size: 22px;  
 color: #0f172a;  
}  
  
.modal-content label {  
 display: block;  
 margin-top: 12px;  
 font-weight: 500;  
 color: #0f172a;  
}  
  
.modal-content input {  
 width: 100%;  
 padding: 10px;  
 border: 1px solid #cbd5e1;  
 border-radius: 10px;  
 margin-top: 5px;  
 font-size: 14px;  
 outline: none;  
}  
  
/\* Buttons \*/  
.modal-buttons {  
 margin-top: 20px;  
 display: flex;  
 justify-content: space-between;  
}  
  
.btn-confirm {  
 background-color: #22c55e;  
 color: white;  
 border: none;  
 padding: 10px 20px;  
 border-radius: 10px;  
 cursor: pointer;  
}  
  
.btn-cancel {  
 background-color: #ef4444;  
 color: white;  
 border: none;  
 padding: 10px 20px;  
 border-radius: 10px;  
 cursor: pointer;  
}  
  
.btn-add-booking:hover {  
 opacity: 0.9;  
 transform: scale(1.03);  
}  
  
  
.btn-send-message {  
 padding: 6px 12px;  
 background: white;  
 color: var(--primary);  
 border: 1px solid var(--primary);  
 border-radius: var(--radius-sm);  
 font-size: 12px;  
 font-weight: 600;  
 cursor: pointer;  
 transition: var(--transition);  
}  
  
.btn-send-message:hover {  
 background-color: var(--primary);  
 color: white;  
 box-shadow: 0 4px 10px rgba(30, 144, 255, 0.3);  
}  
  
/\* Animation Effects \*/  
@keyframes fadeIn {  
 from { opacity: 0; transform: translateY(20px); }  
 to { opacity: 1; transform: translateY(0); }  
}  
  
/\* Responsive Styles \*/  
@media (max-width: 1024px) {  
 .search-sort-container {  
 grid-template-columns: repeat(auto-fill, minmax(240px, 1fr));  
 gap: 20px;  
 }  
   
 .booking-container {  
 flex-direction: column;  
 }  
   
 .venues-sidebar {  
 width: 100%;  
 margin-bottom: 16px;  
 }  
   
 .venues-list {  
 display: flex;  
 flex-wrap: wrap;  
 gap: 8px;  
 }  
   
 .venues-list li {  
 margin-bottom: 0;  
 }  
}  
  
@media (max-width: 768px) {  
 header {  
 flex-direction: column;  
 padding: 16px;  
 }  
   
 nav ul {  
 margin: 16px 0;  
 gap: 12px;  
 overflow-x: auto;  
 padding-bottom: 8px;  
 width: 100%;  
 justify-content: flex-start;  
 }  
   
 .section-header {  
 flex-direction: column;  
 align-items: flex-start;  
 gap: 16px;  
 }  
   
 .filters {  
 width: 100%;  
 }  
   
 select {  
 flex: 1;  
 }  
   
 .search-sort-container {  
 flex-direction: column;  
 align-items: stretch;  
 }  
   
 .sort-section {  
 margin-top: 10px;  
 }  
}  
  
@media (max-width: 576px) {  
 main {  
 padding: 20px 15px;  
 }  
   
 .search-sort-container {  
 grid-template-columns: 1fr;  
 }  
   
 .venue-stats {  
 grid-template-columns: 1fr 1fr;  
 }  
   
 .bookings-table th,   
 .bookings-table td {  
 padding: 12px 10px;  
 font-size: 0.9rem;  
 }  
}  
.venue-rating {  
 color: #f5c518;  
 font-size: 16px;  
 font-weight: bold;  
}  
.form-message {  
 margin-top: 10px;  
 margin-bottom: 10px;  
 padding: 8px 12px;  
 border-radius: 6px;  
 font-size: 14px;  
 display: block;  
 text-align: center;  
}  
  
.form-message.success {  
 background-color: #d4edda;  
 color: #155724;  
 border: 1px solid #c3e6cb;  
}  
  
.form-message.error {  
 background-color: #f8d7da;  
 color: #721c24;  
 border: 1px solid #f5c6cb;  
}

————————————————————————————————————————

## Bookings\Bookings.html

<!DOCTYPE html>  
<html lang="en">  
 <head>  
 <meta charset="UTF-8" />  
 <meta name="viewport" content="width=device-width, initial-scale=1.0" />  
 <title>FreshGlow - Facility Owner Bookings</title>  
 <link  
 href="https://fonts.googleapis.com/css2?family=Outfit:wght@300;400;500;600;700&display=swap"  
 rel="stylesheet"  
 />  
 <!-- Flatpickr CSS -->  
 <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/flatpickr/dist/flatpickr.min.css" />  
  
 <link rel="stylesheet" href="Bookings.css" />  
 </head>  
 <body>  
  
 <main>  
 <!-- Main Content -->  
 <div class="container">  
 <div class="page-header">  
 <h1 class="page-title">Bookings</h1>  
 <p class="page-subtitle">  
 View and control all your venue reservations with clarity and ease  
 </p>  
 </div>  
  
 <!-- Search and Sort -->  
 <div class="search-sort-container">  
 <div class="search-bar">  
 <span class="search-icon">🔍</span>  
 <input  
 type="text"  
 id="searchInput"  
 placeholder="Search venues by name"  
 />  
 </div>  
 <div class="sort-container">  
 <span class="sort-label">Sort By:</span>  
 <select class="sort-select" id="sortSelect">  
 <option value="name">Name (A-Z)</option>  
 <option value="name-desc">Name (Z-A)</option>  
 </select>  
 </div>  
 <div class="sort-container">  
 <select class="sort-select" id="sortSelectSport">  
 <option value="">All Sports</option>  
 </select>  
 </div>  
 </div>  
  
 <div class="venues-container" id="venue-cards-view"></div>  
  
 <!-- Detailed Booking View (initially hidden) -->  
 <section id="booking-detail-view" class="hidden">  
 <div class="booking-container">  
 <!-- Left Sidebar with Venues -->  
 <div class="venues-sidebar">  
 <h3>Your Venues</h3>  
 <ul class="venues-list"></ul>  
 <button class="btn-back-to-venues">Back to All Venues</button>  
 </div>  
  
 <!-- Right Content Area -->  
 <div class="booking-content">  
 <div class="venue-details">  
 <h2 data-venue-id="">Downtown Soccer Field</h2>  
 <div class="venue-meta">  
 <span class="location-badge">Central District</span>  
 <span class="rating-badge">★ 4.8</span>  
 </div>  
 </div>  
 <div style="margin-top: 10px; margin-bottom: 20px">  
 <button class="btn-add-booking">+ Add Booking</button>  
 </div>  
 <div class="booking-controls">  
 <div id="calendar-container"></div>  
 </div>  
  
 <div class="bookings-table-container">  
 <table class="bookings-table">  
 <thead>  
 <tr>  
 <th>Username</th>  
 <th>Players</th>  
 <th>Time Range</th>  
 <th>Action</th>  
 </tr>  
 </thead>  
 <tbody>  
 <tr>  
 <td class="user-column">  
 <span>SoccerFan42</span>  
 </td>  
 <td>8/10</td>  
 <td>09:00 - 11:00</td>  
 <td>  
 <button class="btn-send-message">Send Message</button>  
 </td>  
 </tr>  
 <tr>  
 <td class="user-column">  
 <span>GoalKeeper88</span>  
 </td>  
 <td>6/10</td>  
 <td>13:00 - 15:00</td>  
 <td>  
 <button class="btn-send-message">Send Message</button>  
 </td>  
 </tr>  
 <tr>  
 <td class="user-column">  
 <span>TeamCaptainX</span>  
 </td>  
 <td>10/10</td>  
 <td>18:00 - 20:00</td>  
 <td>  
 <button class="btn-send-message">Send Message</button>  
 </td>  
 </tr>  
 </tbody>  
 </table>  
 </div>  
 </div>  
 </div>  
 </section>  
 </div>  
 </main>  
  
 <!-- Add Booking Modal -->  
 <div id="addBookingModal" class="modal-overlay">  
 <div class="modal-content">  
 <h3>Add New Booking</h3>  
 <form id="bookingForm">  
 <label>Username:</label>  
 <input type="text" name="username" required />  
  
 <label>Date:</label>  
<input type="text" name="date" id="dateInput" required />  
  
 <label>Players:</label>  
 <input type="number" name="players" min="1" max="20" required />  
  
 <label>Start Time:</label>  
<input type="text" id="startTime" name="start\_time" required />  
  
<label>End Time:</label>  
<input type="text" id="endTime" name="end\_time" required />  
  
 <!-- ✅ Error or success message will appear here -->  
 <div id="bookingMessage" class="form-message"></div>  
  
 <div class="modal-buttons">  
 <button type="submit" class="btn-confirm">Add Booking</button>  
 <button  
 type="button"  
 class="btn-cancel"  
 onclick="document.getElementById('addBookingModal').style.display='none';"  
 >  
 Cancel  
 </button>  
 </div>  
 </form>  
 </div>  
 </div>  
 <!-- Message Modal -->  
 <div id="messageModal" class="modal-overlay">  
 <div class="modal-content">  
 <h3>Send Message</h3>  
 <form id="messageForm">  
 <input type="hidden" name="receiver\_username" id="receiverInput" />  
 <label for="messageInput">Message:</label>  
 <input type="text" name="message" id="messageInput" required />  
 <div class="modal-buttons">  
 <button type="submit" class="btn-confirm">Send</button>  
 <button  
 type="button"  
 class="btn-cancel"  
 onclick="document.getElementById('messageModal').style.display='none'"  
 >  
 Cancel  
 </button>  
 </div>  
 </form>  
 </div>  
 </div>  
 <!-- Flatpickr JS -->  
<script src="https://cdn.jsdelivr.net/npm/flatpickr"></script>  
 <!-- Include the JavaScript file -->  
 <script src="bookings.js"></script>  
   
 </body>  
</html>

————————————————————————————————————————

## Bookings\Bookings.js

document.addEventListener('DOMContentLoaded', function () {  
 const searchInput = document.getElementById('searchInput');  
 const sortSelect = document.getElementById('sortSelect');  
 const sportSelect = document.getElementById('sortSelectSport');  
 const venuesContainer = document.querySelector('.venues-container');  
 const bookingDetailView = document.getElementById('booking-detail-view');  
 const venueCardsView = document.getElementById('venue-cards-view');  
 const bookingForm = document.getElementById('bookingForm');  
  
 let allBookings = [];  
 let currentMonthIndex = 0;  
 let globalCalendar = {};  
  
 // Initialize Flatpickr time pickers with improved configuration  
 function initializeTimePickers() {  
 // Start time picker  
 const startTimePicker = flatpickr("#startTime", {  
 enableTime: true,  
 noCalendar: true,  
 dateFormat: "H:i",  
 time\_24hr: true,  
 minuteIncrement: 30,  
 defaultHour: 9,  
 defaultMinute: 0,  
 minTime: "06:00",  
 maxTime: "23:30",  
 onChange: function(selectedDates, dateStr) {  
 // Update end time minimum when start time changes  
 if (endTimePicker && dateStr) {  
 const startTime = dateStr.split(':');  
 const startHour = parseInt(startTime[0]);  
 const startMinute = parseInt(startTime[1]);  
   
 // Set minimum end time to 30 minutes after start time  
 let minEndHour = startHour;  
 let minEndMinute = startMinute + 30;  
   
 if (minEndMinute >= 60) {  
 minEndHour += 1;  
 minEndMinute = 0;  
 }  
   
 const minEndTime = `${minEndHour.toString().padStart(2, '0')}:${minEndMinute.toString().padStart(2, '0')}`;  
 endTimePicker.set('minTime', minEndTime);  
   
 // Clear end time if it's now invalid  
 const currentEndTime = document.getElementById('endTime').value;  
 if (currentEndTime && currentEndTime <= dateStr) {  
 endTimePicker.clear();  
 }  
 }  
 }  
 });  
  
 // End time picker  
 const endTimePicker = flatpickr("#endTime", {  
 enableTime: true,  
 noCalendar: true,  
 dateFormat: "H:i",  
 time\_24hr: true,  
 minuteIncrement: 30,  
 defaultHour: 10,  
 defaultMinute: 0,  
 minTime: "06:30",  
 maxTime: "23:59"  
 });  
  
 return { startTimePicker, endTimePicker };  
 }  
  
 // Initialize date picker  
 function initializeDatePicker() {  
 return flatpickr("#dateInput", {  
 dateFormat: "Y-m-d",  
 minDate: new Date().fp\_incr(1), // Tomorrow onwards  
 disableMobile: true,  
 locale: {  
 firstDayOfWeek: 1 // Start week on Monday  
 }  
 });  
 }  
  
 // Initialize time and date pickers  
 const timePickers = initializeTimePickers();  
 const datePicker = initializeDatePicker();  
  
 // Fetch venues data  
 fetch("BookedFacilitiesFetch.php")  
 .then(res => res.json())  
 .then(data => {  
 if (data.success) {  
 venuesContainer.innerHTML = "";  
 const venuesList = document.querySelector('.venues-list');  
 venuesList.innerHTML = '';  
  
 // Remove duplicates based on ID  
 const uniqueVenues = [];  
 const seenIds = new Set();  
  
 data.venues.forEach(venue => {  
 if (seenIds.has(venue.id)) return;  
 seenIds.add(venue.id);  
 uniqueVenues.push(venue);  
 });  
  
 uniqueVenues.forEach(venue => {  
 const card = document.createElement("div");  
 card.className = "venue-card";  
  
 const ratingValue = parseFloat(venue.rating) || 0;  
 const fullStars = Math.floor(ratingValue);  
 const halfStar = (ratingValue - fullStars) >= 0.5 ? 1 : 0;  
 const emptyStars = 5 - fullStars - halfStar;  
 const starsHTML = '★'.repeat(fullStars) + (halfStar ? '½' : '') + '✩'.repeat(emptyStars);  
  
 card.innerHTML = `  
 <div class="venue-header">  
 <div class="venue-rating">${starsHTML} (${venue.rating ?? 'N/A'})</div>  
 </div>  
 <img src="${venue.image}" class="venue-image" />  
 <h2 class="venue-name">${venue.name}</h2>  
 <p class="venue-location">${venue.location}</p>  
 <div class="venue-stats">  
 <div class="stat">  
 <span class="stat-label">Sport</span>  
 <span class="stat-value venue-sport">${venue.sport}</span>  
 </div>  
 </div>  
 <button class="btn-view-bookings">View Bookings</button>  
 `;  
 venuesContainer.appendChild(card);  
  
 const li = document.createElement('li');  
 li.textContent = venue.name;  
 li.dataset.venueId = venue.id;  
  
 const showBookings = () => {  
 venueCardsView.style.display = 'none';  
 bookingDetailView.classList.remove('hidden');  
  
 const venueTitle = document.querySelector('.booking-content .venue-details h2');  
 const venueLocation = document.querySelector('.location-badge');  
 const venueRating = document.querySelector('.rating-badge');  
  
 venueTitle.textContent = venue.name;  
 venueTitle.dataset.venueId = venue.id;  
 venueLocation.textContent = venue.location;  
 venueRating.textContent = `${starsHTML} (${venue.rating ?? 'N/A'})`;  
  
 fetch(`get\_bookings.php?facilities\_id=${venue.id}`)  
 .then(res => res.json())  
 .then(data => {  
 allBookings = data.bookings || [];  
 renderBookingsForDate('\_\_\_EMPTY\_\_\_');  
   
 if (data.calendar) {  
 globalCalendar = data.calendar;  
 currentMonthIndex = 0;  
 renderCalendar(data.calendar);  
 }  
 });  
 };  
  
 card.querySelector('.btn-view-bookings').addEventListener('click', showBookings);  
 li.addEventListener('click', showBookings);  
 venuesList.appendChild(li);  
 });  
 }  
 });  
  
 // Render bookings for selected date or all bookings  
 function renderBookingsForDate(selectedDate = null) {  
 const tbody = document.querySelector('.bookings-table tbody');  
 tbody.innerHTML = '';  
  
 if (selectedDate === '\_\_\_EMPTY\_\_\_') {  
 tbody.innerHTML = '<tr><td colspan="4">Select a date to view bookings.</td></tr>';  
 return;  
 }  
  
 const filtered = selectedDate  
 ? allBookings.filter(b => b.date === selectedDate)  
 : allBookings;  
  
 if (filtered.length > 0) {  
 filtered.forEach(b => {  
 const row = document.createElement('tr');  
 row.innerHTML = `  
 <td class="user-column"><span>${b.username}</span></td>  
 <td>${b.players}</td>  
 <td>${b.time}</td>  
 <td><button class="btn-send-message" data-receiver="${b.username}">Send Message</button></td>  
 `;  
 tbody.appendChild(row);  
  
 row.querySelector('.btn-send-message').addEventListener('click', function () {  
 const receiver = this.dataset.receiver;  
 document.getElementById('receiverInput').value = receiver;  
 document.getElementById('messageInput').value = '';  
 document.getElementById('messageModal').style.display = 'flex';  
 });  
 });  
 } else {  
 tbody.innerHTML = '<tr><td colspan="4">No bookings found for this date.</td></tr>';  
 }  
 }  
  
 // Calendar rendering and navigation  
 function renderCalendar(calendarData) {  
 globalCalendar = calendarData;  
 showMonth(currentMonthIndex);  
 }  
  
 function showMonth(index) {  
 const container = document.getElementById('calendar-container');  
 container.innerHTML = '';  
  
 const monthNames = Object.keys(globalCalendar);  
 const month = monthNames[index];  
 const dates = globalCalendar[month];  
  
 const nav = document.createElement('div');  
 nav.className = 'month-selector';  
  
 const prevBtn = document.createElement('button');  
 prevBtn.className = 'month-nav prev';  
 prevBtn.innerHTML = '&lt;';  
 prevBtn.onclick = () => {  
 currentMonthIndex = (currentMonthIndex - 1 + monthNames.length) % monthNames.length;  
 showMonth(currentMonthIndex);  
 };  
  
 const nextBtn = document.createElement('button');  
 nextBtn.className = 'month-nav next';  
 nextBtn.innerHTML = '&gt;';  
 nextBtn.onclick = () => {  
 currentMonthIndex = (currentMonthIndex + 1) % monthNames.length;  
 showMonth(currentMonthIndex);  
 };  
  
 const label = document.createElement('span');  
 label.className = 'current-month';  
 label.textContent = month;  
  
 nav.appendChild(prevBtn);  
 nav.appendChild(label);  
 nav.appendChild(nextBtn);  
  
 const daysRow = document.createElement('div');  
 daysRow.className = 'dates-container';  
  
 dates.forEach(date => {  
 const dayBtn = document.createElement('button');  
 dayBtn.className = 'date-btn';  
 dayBtn.textContent = date;  
  
 dayBtn.addEventListener('click', () => {  
 document.querySelectorAll('.date-btn').forEach(btn => btn.classList.remove('active'));  
 dayBtn.classList.add('active');  
 renderBookingsForDate(date);  
 console.log('Selected date:', date);  
 });  
  
 daysRow.appendChild(dayBtn);  
 });  
  
 container.appendChild(nav);  
 container.appendChild(daysRow);  
 }  
  
 // Enhanced form validation function  
 function validateBookingForm() {  
 const startTime = document.getElementById('startTime').value;  
 const endTime = document.getElementById('endTime').value;  
 const dateInput = document.getElementById('dateInput').value;  
 const playersInput = document.getElementById('playersInput')?.value;  
  
 // Clear previous error messages  
 clearErrorMessages();  
  
 let isValid = true;  
 const errors = [];  
  
 // Validate date  
 if (!dateInput) {  
 errors.push("Please select a booking date.");  
 isValid = false;  
 }  
  
 // Validate start time  
 if (!startTime) {  
 errors.push("Please select a start time.");  
 isValid = false;  
 }  
  
 // Validate end time  
 if (!endTime) {  
 errors.push("Please select an end time.");  
 isValid = false;  
 }  
  
 // Validate time logic  
 if (startTime && endTime) {  
 if (startTime >= endTime) {  
 errors.push("End time must be after start time.");  
 isValid = false;  
 }  
  
 // Check minimum duration (30 minutes)  
 const start = new Date(`1970-01-01T${startTime}:00`);  
 const end = new Date(`1970-01-01T${endTime}:00`);  
 const diffMinutes = (end - start) / (1000 \* 60);  
  
 if (diffMinutes < 30) {  
 errors.push("Minimum booking duration is 30 minutes.");  
 isValid = false;  
 }  
 }  
  
 // Validate players (if field exists)  
 if (playersInput !== undefined) {  
 const players = parseInt(playersInput);  
 if (!players || players < 1) {  
 errors.push("Please enter a valid number of players.");  
 isValid = false;  
 }  
 }  
  
 // Display errors  
 if (!isValid) {  
 showErrors(errors);  
 }  
  
 return isValid;  
 }  
  
 function showErrors(errors) {  
 const messageBox = document.getElementById('bookingMessage');  
 if (messageBox) {  
 messageBox.className = 'form-message error';  
 messageBox.innerHTML = errors.join('<br>');  
 }  
 }  
  
 function clearErrorMessages() {  
 const messageBox = document.getElementById('bookingMessage');  
 if (messageBox) {  
 messageBox.textContent = '';  
 messageBox.className = '';  
 }  
 }  
  
 // Booking form submission with enhanced validation  
 if (bookingForm) {  
 bookingForm.onsubmit = function (e) {  
 e.preventDefault();  
  
 // Validate form before submission  
 if (!validateBookingForm()) {  
 return;  
 }  
  
 const formData = new FormData(bookingForm);  
 const selectedVenueId = document.querySelector('.booking-content h2')?.dataset.venueId;  
  
 if (!selectedVenueId) {  
 showErrors(["Unable to identify selected venue. Please try again."]);  
 return;  
 }  
  
 formData.append('facility\_id', selectedVenueId);  
  
 fetch('add\_booking.php', {  
 method: 'POST',  
 body: formData  
 })  
 .then(res => res.json())  
 .then(data => {  
 const messageBox = document.getElementById('bookingMessage');  
  
 if (messageBox) {  
 messageBox.className = data.success ? 'form-message success' : 'form-message error';  
 messageBox.textContent = data.message;  
 }  
  
 if (data.success) {  
 setTimeout(() => {  
 bookingForm.reset();  
 // Reset time pickers  
 timePickers.startTimePicker.clear();  
 timePickers.endTimePicker.clear();  
 datePicker.clear();  
   
 clearErrorMessages();  
 document.getElementById('addBookingModal').style.display = 'none';  
  
 // Refresh bookings  
 fetch(`get\_bookings.php?facilities\_id=${selectedVenueId}`)  
 .then(res => res.json())  
 .then(data => {  
 allBookings = data.bookings || [];  
 renderBookingsForDate('\_\_\_EMPTY\_\_\_');  
   
 if (data.calendar) {  
 globalCalendar = data.calendar;  
 renderCalendar(data.calendar);  
 }  
 });  
 }, 1500);  
 }  
 })  
 .catch(error => {  
 console.error('Booking submission error:', error);  
 showErrors(["Failed to submit booking. Please try again."]);  
 });  
 };  
 }  
  
 // Search functionality  
 if (searchInput) {  
 searchInput.addEventListener('input', function () {  
 const query = searchInput.value.trim().toLowerCase();  
 const venueCards = document.querySelectorAll('.venue-card');  
 venueCards.forEach(card => {  
 const name = card.querySelector('.venue-name')?.textContent.toLowerCase() || '';  
 card.style.display = name.startsWith(query) ? 'flex' : 'none';  
 });  
 });  
 }  
  
 // Sort functionality  
 if (sortSelect) {  
 sortSelect.addEventListener('change', function () {  
 const cards = Array.from(document.querySelectorAll('.venue-card'));  
 cards.sort((a, b) => {  
 const nameA = a.querySelector('.venue-name').textContent.toLowerCase();  
 const nameB = b.querySelector('.venue-name').textContent.toLowerCase();  
 return sortSelect.value === 'name' ? nameA.localeCompare(nameB) : nameB.localeCompare(nameA);  
 });  
 venuesContainer.innerHTML = '';  
 cards.forEach(card => venuesContainer.appendChild(card));  
 });  
 }  
  
 // Sport filter functionality  
 if (sportSelect) {  
 sportSelect.addEventListener('change', function () {  
 const selectedSport = sportSelect.value.toLowerCase();  
 const venueCards = document.querySelectorAll('.venue-card');  
 venueCards.forEach(card => {  
 const sport = card.querySelector('.venue-sport')?.textContent.toLowerCase() || '';  
 card.style.display = selectedSport === '' || sport === selectedSport ? 'flex' : 'none';  
 });  
 });  
 }  
  
 // Back button functionality  
 const backBtn = document.querySelector('.btn-back-to-venues');  
 if (backBtn) {  
 backBtn.addEventListener('click', function () {  
 if (bookingDetailView) bookingDetailView.classList.add('hidden');  
 if (venueCardsView) venueCardsView.style.display = 'flex';  
 });  
 }  
  
 // Load sports for filter dropdown  
 fetch("get\_sports.php")  
 .then(res => res.json())  
 .then(sports => {  
 if (sportSelect) {  
 sports.forEach(sport => {  
 const option = document.createElement("option");  
 option.value = sport;  
 option.textContent = sport;  
 sportSelect.appendChild(option);  
 });  
 }  
 })  
 .catch(err => console.error("Failed to load sports:", err));  
  
 // Modal functionality  
 const addBookingBtn = document.querySelector('.btn-add-booking');  
 const addBookingModal = document.getElementById('addBookingModal');  
 const cancelBtn = document.querySelector('.btn-cancel');  
  
 if (addBookingBtn) {  
 addBookingBtn.onclick = function () {  
 if (addBookingModal) {  
 addBookingModal.style.display = 'flex';  
 clearErrorMessages(); // Clear any previous error messages  
 }  
 };  
 }  
  
 if (cancelBtn) {  
 cancelBtn.onclick = function () {  
 if (addBookingModal) {  
 addBookingModal.style.display = 'none';  
 clearErrorMessages();  
 }  
 };  
 }  
  
 // Close modal when clicking outside  
 window.onclick = function (e) {  
 if (e.target === addBookingModal) {  
 addBookingModal.style.display = 'none';  
 clearErrorMessages();  
 }  
 };  
  
 // Message form submission  
 const messageForm = document.getElementById('messageForm');  
 if (messageForm) {  
 messageForm.addEventListener('submit', function (e) {  
 e.preventDefault();  
 const receiver = document.getElementById('receiverInput').value;  
 const message = document.getElementById('messageInput').value;  
  
 if (!message.trim()) {  
 alert("Please enter a message.");  
 return;  
 }  
  
 fetch('send\_message.php', {  
 method: 'POST',  
 headers: { 'Content-Type': 'application/x-www-form-urlencoded' },  
 body: new URLSearchParams({ receiver\_username: receiver, message })  
 })  
 .then(res => res.json())  
 .then(data => {  
 alert(data.message);  
 if (data.success) {  
 document.getElementById('messageModal').style.display = 'none';  
 }  
 })  
 .catch(err => {  
 console.error('Error sending message:', err);  
 alert('Failed to send message. Please try again.');  
 });  
 });  
 }  
});

————————————————————————————————————————

## Bookings\Bookings.php

<?php  
session\_start();  
if (!isset($\_SESSION['user\_id'])) {  
 // Unset all session variables  
 session\_unset();  
 // Destroy the session completely  
 session\_destroy();  
 header('Location: ../../auth/Login\_Page/Login.php');  
 exit();  
}  
include '../navbar.php';  
include 'Bookings.html';  
exit();  
?>

————————————————————————————————————————

## Bookings\get\_bookings.php

<?php  
require\_once '../../../db.php';  
header('Content-Type: application/json');  
  
$facilityId = $\_GET['facilities\_id'] ?? null;  
  
if (!$facilityId) {  
 echo json\_encode(['success' => false, 'message' => 'Facility ID missing']);  
 exit;  
}  
  
$currentYear = date("Y");  
$calendar = [];  
  
// Prepare all months empty  
$months = [  
 "January", "February", "March", "April", "May", "June",  
 "July", "August", "September", "October", "November", "December"  
];  
  
foreach ($months as $month) {  
 $calendar[$month] = [];  
}  
  
// Fill dates that have bookings  
$query = "  
 SELECT   
 DATE\_FORMAT(booking\_date, '%M') AS month\_name,  
 DATE\_FORMAT(booking\_date, '%Y-%m-%d') AS full\_date  
 FROM bookings  
 WHERE facilities\_id = ? AND YEAR(booking\_date) = ?  
 GROUP BY full\_date  
 ORDER BY full\_date ASC  
";  
$stmt = $conn->prepare($query);  
$stmt->bind\_param("ii", $facilityId, $currentYear);  
$stmt->execute();  
$result = $stmt->get\_result();  
  
while ($row = $result->fetch\_assoc()) {  
 $month = $row['month\_name'];  
 $calendar[$month][] = $row['full\_date'];  
}  
  
// Display table bookings  
$query2 = "  
 SELECT   
 b.booking\_id,  
 b.username,  
 b.start\_time,  
 b.end\_time,  
 DATE(b.booking\_date) AS booking\_date,  
 COUNT(gm.username) AS member\_count,  
 g.max\_members  
 FROM bookings b  
 LEFT JOIN groups g ON g.booking\_id = b.booking\_id  
 LEFT JOIN group\_members gm ON gm.group\_id = g.group\_id  
 WHERE b.facilities\_id = ?  
 GROUP BY b.booking\_id  
 ORDER BY b.booking\_date DESC, b.start\_time ASC  
";  
$stmt2 = $conn->prepare($query2);  
$stmt2->bind\_param("i", $facilityId);  
$stmt2->execute();  
$result2 = $stmt2->get\_result();  
  
$bookings = [];  
while ($row = $result2->fetch\_assoc()) {  
 $playersDisplay = isset($row['max\_members'])   
 ? $row['member\_count'] . '/' . $row['max\_members']   
 : '0/0'; // in case there's no group  
  
 $bookings[] = [  
 'username' => $row['username'],  
 'time' => $row['start\_time'] . ' - ' . $row['end\_time'],  
 'players' => $playersDisplay,  
 'date' => $row['booking\_date']  
 ];  
}  
  
  
echo json\_encode([  
 'success' => true,  
 'calendar' => $calendar,  
 'bookings' => $bookings  
]);  
?>

————————————————————————————————————————

## Bookings\get\_sports.php

<?php  
require\_once '../../../db.php';  
header('Content-Type: application/json');  
  
$query = "SELECT DISTINCT SportCategory as sport\_name FROM sportfacilities WHERE is\_Accepted = 1";  
$result = $conn->query($query);  
  
$sports = [];  
while ($row = $result->fetch\_assoc()) {  
 $sports[] = $row['sport\_name'];  
}  
  
echo json\_encode($sports);  
?>

————————————————————————————————————————

## Bookings\send\_message.php

<?php  
session\_start();  
require\_once '../../../db.php';  
header('Content-Type: application/json');  
  
$sender = $\_SESSION['user\_id'] ?? null;  
$receiver = $\_POST['receiver\_username'] ?? '';  
$message = trim($\_POST['message'] ?? '');  
  
if (!$sender || !$receiver || !$message) {  
 echo json\_encode(['success' => false, 'message' => 'Missing fields']);  
 exit;  
}  
  
$stmt = $conn->prepare("INSERT INTO messages (sender\_username, receiver\_username, message\_text) VALUES (?, ?, ?)");  
$stmt->bind\_param("sss", $sender, $receiver, $message);  
  
if ($stmt->execute()) {  
 echo json\_encode(['success' => true, 'message' => 'Message sent successfully.']);  
} else {  
 echo json\_encode(['success' => false, 'message' => 'Failed to send message.']);  
}  
?>

————————————————————————————————————————

## HomePage\HomePage.css

/\* Light Elegant Theme - Updated Version \*/  
:root {  
 --primary: #1e90ff;  
 --secondary: #22c55e;  
 --light-bg: #f8f6f0;  
 --card-bg: #ffffff;  
 --card-border: #d1d5db;  
 --text-main: #1e293b;  
 --muted-text: #6b7280;  
 --accent-glow: rgba(30, 144, 255, 0.3);  
 --radius: 0.75rem;  
}  
body {  
 margin: 0;  
 font-family: "Outfit", sans-serif;  
 background: linear-gradient(135deg, #e0f7fa, #d0f8ce);  
 color: var(--text-main);  
 min-height: 100vh;  
 display: flex;  
 flex-direction: column;  
 align-items: center;  
 overflow-x: hidden;  
}  
/\* Header styles \*/  
.header-bar {  
 width: 100%;  
 padding: 20px 40px;  
 display: flex;  
 justify-content: space-between;  
 align-items: center;  
 box-sizing: border-box;  
}  
.logo-title {  
 font-size: 32px;  
 font-weight: bold;  
 background: linear-gradient(to right, #1e90ff, #22c55e);  
 -webkit-background-clip: text;  
 background-clip: text;  
 -webkit-text-fill-color: transparent;  
}  
.logo-title span {  
 color: #f472b6;  
}  
/\* Profile dropdown \*/  
.profile-container {  
 position: relative;  
 cursor: pointer;  
 z-index: 100;  
}  
.profile-container {  
 position: absolute;  
 top: 20px;  
 right: 20px;  
 display: flex;  
 gap: 10px;  
 align-items: center;  
}  
  
.admin-request-icon {  
 width: 30px;  
 height: 30px;  
 background-color: #facc15;  
 color: black;  
 border-radius: 50%;  
 display: flex;  
 align-items: center;  
 justify-content: center;  
 font-weight: bold;  
 cursor: pointer;  
 box-shadow: 0 0 6px #facc15;  
 transition: all 0.3s ease;  
}  
  
.admin-request-icon:hover {  
 transform: scale(1.1);  
}  
  
.admin-banner {  
 width: 100%;  
 background: linear-gradient(90deg, #fff8e1, #fef3c7);  
 color: var(--text-main);  
 display: flex;  
 justify-content: center;  
 align-items: center;  
 padding: 12px 20px;  
 font-size: 15px;  
 font-weight: 500;  
 border-bottom: 1px solid #fcd34d;  
 box-shadow: 0 2px 8px rgba(252, 211, 77, 0.2);  
 z-index: 99;  
 position: relative;  
}  
  
.floating-admin-btn {  
 position: fixed;  
 bottom: 20px;  
 right: 20px;  
 background: linear-gradient(135deg, #facc15, #fef08a);  
 color: #1e293b;  
 border: none;  
 border-radius: 12px;  
 padding: 12px 20px;  
 font-size: 14px;  
 font-weight: 600;  
 box-shadow: 0 4px 12px rgba(250, 204, 21, 0.4);  
 cursor: pointer;  
 transition: all 0.3s ease;  
 z-index: 999;  
}  
  
.floating-admin-btn:hover {  
 background: linear-gradient(135deg, #eab308, #fde047);  
 box-shadow: 0 6px 16px rgba(250, 204, 21, 0.6);  
}  
  
.profile-icon {  
 width: 60px;  
 height: 60px;  
 border-radius: 50%;  
 object-fit: cover;  
 box-shadow: 0 0 6px var(--accent-glow);  
}  
  
.profile-menu {  
 position: absolute;  
 top: 50px;  
 right: 0;  
 background-color: white;  
 border: 1px solid var(--card-border);  
 border-radius: var(--radius);  
 box-shadow: 0 4px 16px rgba(0, 0, 0, 0.06);  
 overflow: hidden;  
 display: none;  
 flex-direction: column;  
 min-width: 180px;  
 z-index: 101;  
}  
  
.profile-menu a {  
 padding: 12px 18px;  
 text-decoration: none;  
 color: var(--text-main);  
 font-size: 14px;  
}  
  
.profile-menu a:hover {  
 background-color: #f3f4f6;  
}  
  
.profile-menu.active {  
 display: flex;  
}  
.profile-menu {  
 min-width: 160px;  
 right: -10px;  
}  
/\* Main Buttons \*/  
.main-buttons {  
 display: flex;  
 gap: 30px;  
 margin-top: 60px;  
 flex-wrap: wrap;  
 justify-content: center;  
 width: 90%;  
 max-width: 800px;  
}  
/\* Secondary buttons \*/  
.secondary-buttons {  
 display: flex;  
 gap: 30px;  
 flex-wrap: wrap;  
 justify-content: center;  
 width: 90%;  
 max-width: 800px;  
 margin: 60px auto;  
}  
.glow-card {  
 width: 200px;  
 height: 200px;  
 background: var(--card-bg);  
 border: 1px solid var(--card-border);  
 border-radius: var(--radius);  
 padding: 20px;  
 text-align: center;  
 color: var(--text-main);  
 box-shadow: 0 4px 12px rgba(30, 144, 255, 0.05);  
 transition: all 0.3s ease;  
 cursor: pointer;  
 display: flex;  
 flex-direction: column;  
 justify-content: center;  
 align-items: center;  
}  
  
.glow-card img.icon {  
 width: 80px;  
 margin-bottom: 12px;  
 max-width: 100%;  
 height: auto;  
}  
  
.glow-card span {  
 font-size: 15px;  
 font-weight: 500;  
 color: var(--text-main);  
 text-align: center;  
}  
  
.glow-card:hover {  
 transform: translateY(-5px);  
 box-shadow: 0 6px 20px rgba(30, 144, 255, 0.15);  
}  
/\* Button special glows \*/  
.btn-venue {  
 box-shadow: 0 0 12px #38bdf8;  
}  
  
.btn-bookings {  
 box-shadow: 0 0 12px #22c55e;  
  
}  
.btn-messages {  
 box-shadow: 0 0 12px #3b82f6;  
}  
.btn-analytics {  
 box-shadow: 0 0 12px #fb923c;  
  
}  
.btn-switch {  
 box-shadow: 0 0 12px #f43f5e;  
}  
/\* Footer styles \*/  
footer {  
 background-color: transparent;  
 padding: 20px;  
 width: 100%;  
 text-align: center;  
 font-size: 14px;  
 color: var(--muted-text);  
 margin-top: auto;  
 box-sizing: border-box;  
}  
.footer-content {  
 max-width: 1200px;  
 margin: auto;  
}  
/\* Animations \*/  
@keyframes fadeInUp {  
 from {  
 opacity: 0;  
 transform: translateY(20px);  
 }  
 to {  
 opacity: 1;  
 transform: translateY(0);  
 }  
}  
.main-buttons,  
.secondary-buttons {  
 animation: fadeInUp 0.6s ease-out forwards;  
}  
.username-display {  
 margin-left: 20px;  
 font-weight: 600;  
 color: var(--text-main);  
 font-size: 15px;  
}

————————————————————————————————————————

## HomePage\HomePage.html

<!DOCTYPE html>  
<html lang="en">  
 <head>  
 <meta charset="UTF-8" />  
 <meta name="viewport" content="width=device-width, initial-scale=1.0" />  
 <title>Book&Play - Facility Owner</title>  
 <link  
 href="https://fonts.googleapis.com/css2?family=Outfit:wght@300;400;500;600;700&display=swap"  
 rel="stylesheet"  
 />  
 <link rel="stylesheet" href="HomePage.css" />  
 </head>  
 <body>  
 <!-- Header -->  
 <div class="header-bar">  
 <div class="logo-title">BOOK&PLAY</div>  
 <div class="profile-container" onclick="toggleProfileMenu()">  
 <span id="usernameDisplay" class="username-display"></span>  
 <img  
 id="userProfileImage"  
 class="profile-icon"  
 src="../../../uploads/users/default.jpg"  
 alt="Profile Image"  
 />  
  
 <div class="profile-menu" id="profileMenu">  
 <a href="../.././auth/EditProfile/EditProfile.php">Edit Profile</a>  
 <a href="../../../logout.php">Logout</a>  
 </div>  
 </div>  
 </div>  
  
 <!-- Main Buttons (Top Row - 3 cards) -->  
 <div class="main-buttons">  
 <div  
 class="glow-card btn-venue"  
 onclick="window.location.href='../ManageVenue/ManageVenue.php'"  
 >  
 <img  
 src="../../../Images/Venue\_icon (2).png"  
 alt="Venue Icon"  
 class="icon"  
 width="32"  
 height="32"  
 />  
 <span>Manage Venue</span>  
 </div>  
 <div  
 class="glow-card btn-bookings"  
 onclick="window.location.href='../Bookings/Bookings.php'"  
 >  
 <img  
 src="../../../Images/MangeVenue\_icon.png"  
 alt="Bookings Icon"  
 class="icon"  
 width="32"  
 height="32"  
 />  
 <span>Bookings</span>  
 </div>  
 <div  
 class="glow-card btn-analytics"  
 onclick="window.location.href='../Analytics/Analytics.php'"  
 >  
 <img  
 src="../../../Images/cash.png"  
 alt="Analytics Icon"  
 class="icon"  
 width="32"  
 height="32"  
 />  
 <span>Analytics</span>  
 </div>  
 </div>  
  
 <!-- Secondary Buttons (Bottom Row - 2 cards) -->  
 <div class="secondary-buttons">  
 <div  
 class="glow-card btn-messages"  
 onclick="window.location.href='../Messages/Messages.php'"  
 >  
 <img  
 src="../../../Images/chat\_icon.png"  
 alt="Messages Icon"  
 class="icon"  
 width="32"  
 height="32"  
 />  
 <span>Messages</span>  
 </div>  
  
 <div  
 class="glow-card btn-switch"  
 onclick="window.location.href='../../auth/User\_Selection\_Page/user-selection.php'"  
 >  
 <img  
 src="../../../Images/BackToUserSelecionRed.png"  
 alt="Enter Icon"  
 class="icon"  
 width="32"  
 height="32"  
 />  
 <span>Back To User Selection Page</span>  
 </div>  
 </div>  
  
 <!-- Footer -->  
 <footer>  
 <div class="footer-content">© 2025 Book&Play. All rights reserved.</div>  
 </footer>  
  
 <!-- Floating Admin Button -->  
 <button class="floating-admin-btn" id="adminRequestBtn">  
 Request Admin Access  
 </button>  
  
 <script src="HomePage.js"></script>  
 </body>  
</html>

————————————————————————————————————————

## HomePage\HomePage.js

// Toggle the profile menu visibility  
function toggleProfileMenu() {  
 const menu = document.getElementById("profileMenu");  
 menu.classList.toggle("active");  
}  
  
// Close the profile menu when clicking outside of it  
document.addEventListener("click", function (e) {  
 const profileContainer = document.querySelector(".profile-container");  
 const menu = document.getElementById("profileMenu");  
 if (!profileContainer.contains(e.target)) {  
 menu.classList.remove("active");  
 }  
});  
  
document.addEventListener("DOMContentLoaded", () => {  
 const adminBtn = document.getElementById("adminRequestBtn");  
 if (!adminBtn) return; // Exit if the button doesn't exist  
  
 // Check admin status and if a request has already been submitted  
 fetch("../../admin\_actions.php?action=check")  
 .then((res) => res.json())  
 .then((data) => {  
 // Hide the button if the user is an admin  
 if (data.is\_admin) {  
 adminBtn.style.display = "none";  
 return;  
 }  
  
 // Handle button click  
 adminBtn.addEventListener("click", () => {  
 if (data.already\_requested) {  
 // Show waiting message if request already submitted  
 showTempMessage(  
 "🕐 Your admin request is under review. Thank you for your patience.",  
 4000  
 );  
 } else {  
 // Send admin request  
 fetch("../../admin\_actions.php", {  
 method: "POST",  
 headers: {  
 "Content-Type": "application/x-www-form-urlencoded",  
 },  
 body: "action=submit",  
 })  
 .then((res) => res.json())  
 .then((response) => {  
 if (response.success) {  
 showTempMessage(  
 "✅ Your admin request has been submitted. We'll review it as soon as possible.",  
 4000  
 );  
 data.already\_requested = true; // Prevent re-submission  
 } else {  
 showTempMessage(  
 "❌ " + (response.message || "Failed to submit request."),  
 4000  
 );  
 }  
 })  
 .catch((error) => {  
 console.error("Error submitting request:", error);  
 showTempMessage(  
 "❌ Could not connect to server. Try again later.",  
 4000  
 );  
 });  
 }  
 });  
 })  
 .catch((error) => {  
 console.error("Error checking admin status:", error);  
 showTempMessage(  
 "❌ Could not check admin status. Try again later.",  
 4000  
 );  
 });  
});  
  
// Helper function to show temporary messages on screen  
function showTempMessage(message, duration) {  
 const msg = document.createElement("div");  
 msg.textContent = message;  
 msg.style.position = "fixed";  
 msg.style.bottom = "80px";  
 msg.style.right = "20px";  
 msg.style.background = "#1e90ff";  
 msg.style.color = "white";  
 msg.style.padding = "12px 20px";  
 msg.style.borderRadius = "10px";  
 msg.style.boxShadow = "0 4px 12px rgba(0,0,0,0.2)";  
 msg.style.zIndex = 1000;  
 msg.style.fontSize = "14px";  
 document.body.appendChild(msg);  
 setTimeout(() => msg.remove(), duration);  
}  
document.addEventListener("DOMContentLoaded", () => {  
 fetch("HomePage.php?action=get\_user\_image")  
 .then((res) => res.json())  
 .then((data) => {  
 const imgElement = document.getElementById("userProfileImage");  
 const usernameSpan = document.getElementById("usernameDisplay");  
 if (imgElement && data.image) {  
 imgElement.src = `../../uploads/users/${data.image}`;  
 }  
 if (usernameSpan && data.username) {  
 usernameSpan.textContent = data.username;  
 }  
 })  
 .catch((err) => {  
 console.error("Failed to load user image or username", err);  
 });  
});

————————————————————————————————————————

## HomePage\HomePage.php

<?php  
session\_start();  
require\_once '../../../db.php';  
$\_SESSION['previous\_page'] = $\_SERVER['PHP\_SELF'];  
  
// ✅ 1. Check request to get user image  
if (isset($\_GET['action']) && $\_GET['action'] === 'get\_user\_image') {  
 $username = $\_SESSION['user\_id'] ?? '';  
  
 if (!$username) {  
 echo json\_encode(['error' => 'Not logged in']);  
 exit;  
 }  
  
 $stmt = $conn->prepare("SELECT user\_image, username FROM users WHERE username = ?");  
 $stmt->bind\_param("s", $username);  
 $stmt->execute();  
 $result = $stmt->get\_result();  
 if ($user = $result->fetch\_assoc()) {  
 echo json\_encode(['image' => $user['user\_image'], 'username' => $user['username']]);  
 } else {  
 echo json\_encode(['error' => 'User not found']);  
 }  
 exit;  
}  
  
// ✅ 2. Check login and load page  
if (!isset($\_SESSION['user\_id'])) {  
 session\_unset();  
 session\_destroy();  
 header('Location: ../../auth/Login\_Page/Login.php');  
 exit();  
}  
  
include 'HomePage.html';  
exit();  
?>

————————————————————————————————————————

## ManageVenue\fetch\_venues.php

<?php  
session\_start();  
require\_once '../../../db.php';  
require\_once 'upload\_images.php';  
  
header('Content-Type: application/json');  
  
$action = $\_GET['action'] ?? $\_POST['action'] ?? '';  
  
switch ($action) {  
 case 'get\_facilities':  
 getFacilities();  
 break;  
  
 case 'update\_availability':  
 updateAvailability();  
 break;  
  
 case 'get\_sports':  
 getSports();  
 break;  
  
 case 'add\_facility':  
 addFacility();  
 break;  
  
 case 'update\_facility':  
 updateFacility();  
 break;  
  
 default:  
 echo json\_encode(["success" => false, "message" => "⛔ Invalid action"]);  
 break;  
}  
  
function getFacilities() {  
 global $conn;  
  
 if (!isset($\_SESSION['user\_id'])) {  
 echo json\_encode(['success' => false, 'message' => 'Not logged in']);  
 return;  
 }  
  
 $username = $\_SESSION['user\_id'];  
 $stmt = $conn->prepare("SELECT \* FROM sportfacilities WHERE owner\_username = ? AND is\_Accepted = 1");  
 $stmt->bind\_param("s", $username);  
 $stmt->execute();  
 $result = $stmt->get\_result();  
  
 $facilities = [];  
 while ($row = $result->fetch\_assoc()) {  
 $facilities[] = $row;  
 }  
  
 echo json\_encode($facilities);  
}  
  
function updateAvailability() {  
 global $conn;  
  
 if (!isset($\_SESSION['user\_id'])) {  
 echo json\_encode(['success' => false, 'message' => 'Not logged in']);  
 return;  
 }  
  
 $username = $\_SESSION['user\_id'];  
 $facility\_id = $\_POST['facility\_id'] ?? null;  
 $is\_available = $\_POST['is\_available'] ?? null;  
  
 $check\_stmt = $conn->prepare("SELECT \* FROM sportfacilities WHERE facilities\_id = ? AND owner\_username = ?");  
 $check\_stmt->bind\_param("is", $facility\_id, $username);  
 $check\_stmt->execute();  
 $result = $check\_stmt->get\_result();  
  
 if ($result->num\_rows === 0) {  
 echo json\_encode(['success' => false, 'message' => 'Unauthorized access']);  
 return;  
 }  
  
 $stmt = $conn->prepare("UPDATE sportfacilities SET is\_available = ? WHERE facilities\_id = ? AND owner\_username = ?");  
 $stmt->bind\_param("iis", $is\_available, $facility\_id, $username);  
  
 if ($stmt->execute()) {  
 echo json\_encode(["success" => true, "message" => "Availability updated"]);  
 } else {  
 echo json\_encode(["success" => false, "message" => "Failed to update"]);  
 }  
}  
  
function getSports() {  
 global $conn;  
 $sports = [];  
 $result = $conn->query("SELECT DISTINCT SportCategory as sport\_name FROM sportfacilities WHERE is\_Accepted = 1");  
  
 while ($row = $result->fetch\_assoc()) {  
 $sports[] = $row;  
 }  
  
 echo json\_encode($sports);  
}  
  
function addFacility() {  
 global $conn;  
  
 if (!isset($\_SESSION['user\_id'])) {  
 echo json\_encode(['success' => false, 'message' => 'User not logged in']);  
 return;  
 }  
  
 $username = $\_SESSION['user\_id'];  
 $place\_name = $\_POST['place\_name'] ?? '';  
 $sport\_type = $\_POST['sport\_type'] ?? '';  
 $description = $\_POST['description'] ?? '';  
 $price = intval($\_POST['price'] ?? 0);  
 $location = $\_POST['location'] ?? '';  
 $latitude = $\_POST['latitude'] ?? null; // receive latitude  
 $longitude = $\_POST['longitude'] ?? null; // receive longitude  
 $is\_available = isset($\_POST['is\_available']) ? 1 : 0;  
 $image\_urls = [];  
  
 // Check image upload  
 if (isset($\_FILES['venueImages']) && is\_array($\_FILES['venueImages']['name']) && $\_FILES['venueImages']['error'][0] !== UPLOAD\_ERR\_NO\_FILE) {  
 $image\_urls = upload\_images($\_FILES['venueImages']);  
 }  
  
 // Convert array of image links to comma-separated text  
 $image\_url\_string = implode(',', $image\_urls);  
   
 // Add coordinates to location if available  
 if ($latitude && $longitude) {  
 // Add coordinates to location (can use JSON format or any other format)  
 $coordinates = json\_encode(['lat' => $latitude, 'lng' => $longitude]);  
 } else {  
 $coordinates = null;  
 }  
   
 $conn->begin\_transaction();  
  
 try {  
 // Modify query to add coordinates field  
 $stmt = $conn->prepare("INSERT INTO sportfacilities (place\_name, location, description, image\_url, owner\_username, SportCategory, price, latitude, longitude, is\_Accepted, is\_available) VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, 0, ?)");  
 $stmt->bind\_param("ssssssiddi", $place\_name, $location, $description, $image\_url\_string, $username, $sport\_type, $price, $latitude, $longitude, $is\_available);  
  
 if (!$stmt->execute()) {  
 throw new Exception("Failed to add facility: " . $conn->error);  
 }  
  
 $facility\_id = $conn->insert\_id;  
 $report\_stmt = $conn->prepare("INSERT INTO reports (username, type, suggested\_place\_name, facilities\_id, message, created\_at) VALUES (?, 'suggest\_place', ?, ?, ?, NOW())");  
 $report\_msg = "Sport type: $sport\_type\nLocation: $location\nPrice: $price\nDescription: $description\nImages: $image\_url\_string\nCoordinates: $coordinates";  
 $report\_stmt->bind\_param("ssis", $username, $place\_name, $facility\_id, $report\_msg);  
  
 if (!$report\_stmt->execute()) {  
 throw new Exception("Failed to insert report: " . $conn->error);  
 }  
  
 $conn->commit();  
 echo json\_encode(["success" => true, "message" => "Facility added successfully", "facility\_id" => $facility\_id]);  
  
 } catch (Exception $e) {  
 $conn->rollback();  
 echo json\_encode(["success" => false, "message" => $e->getMessage()]);  
 }  
}  
function updateFacility() {  
 global $conn;  
  
 if (!isset($\_SESSION['user\_id'])) {  
 echo json\_encode(['success' => false, 'message' => 'User not logged in']);  
 return;  
 }  
  
 $username = $\_SESSION['user\_id'];  
 $facility\_id = $\_POST['facility\_id'] ?? null;  
 $place\_name = $\_POST['place\_name'] ?? '';  
 $sport\_type = $\_POST['sport\_type'] ?? '';  
 $description = $\_POST['description'] ?? '';  
 $price = intval($\_POST['price'] ?? 0);  
 $location = $\_POST['location'] ?? '';  
 $latitude = $\_POST['latitude'] ?? null;  
 $longitude = $\_POST['longitude'] ?? null;  
 $is\_available = isset($\_POST['is\_available']) ? 1 : 0;  
  
 // Check if user owns this facility  
 $check\_stmt = $conn->prepare("SELECT \* FROM sportfacilities WHERE facilities\_id = ? AND owner\_username = ?");  
 $check\_stmt->bind\_param("is", $facility\_id, $username);  
 $check\_stmt->execute();  
 $result = $check\_stmt->get\_result();  
  
 if ($result->num\_rows === 0) {  
 echo json\_encode(['success' => false, 'message' => 'Unauthorized access']);  
 return;  
 }  
  
 // Handle image upload if new images are provided  
 $image\_url\_string = '';  
 if (isset($\_FILES['venueImages']) && is\_array($\_FILES['venueImages']['name']) && $\_FILES['venueImages']['error'][0] !== UPLOAD\_ERR\_NO\_FILE) {  
 $image\_urls = upload\_images($\_FILES['venueImages']);  
 $image\_url\_string = implode(',', $image\_urls);  
 }  
  
 // Update query - only update image\_url if new images were uploaded  
 if ($image\_url\_string) {  
 $stmt = $conn->prepare("UPDATE sportfacilities SET place\_name = ?, location = ?, description = ?, image\_url = ?, SportCategory = ?, price = ?, latitude = ?, longitude = ?, is\_available = ? WHERE facilities\_id = ? AND owner\_username = ?");  
 $stmt->bind\_param("ssssssiddis", $place\_name, $location, $description, $image\_url\_string, $sport\_type, $price, $latitude, $longitude, $is\_available, $facility\_id, $username);  
 } else {  
 $stmt = $conn->prepare("UPDATE sportfacilities SET place\_name = ?, location = ?, description = ?, SportCategory = ?, price = ?, latitude = ?, longitude = ?, is\_available = ? WHERE facilities\_id = ? AND owner\_username = ?");  
 $stmt->bind\_param("sssssiddis", $place\_name, $location, $description, $sport\_type, $price, $latitude, $longitude, $is\_available, $facility\_id, $username);  
 }  
  
 if ($stmt->execute()) {  
 echo json\_encode(["success" => true, "message" => "Facility updated successfully"]);  
 } else {  
 echo json\_encode(["success" => false, "message" => "Failed to update facility"]);  
 }  
}  
?>

————————————————————————————————————————

## ManageVenue\ManageVenue.css

.page-header {  
 margin-bottom: 30px;  
 position: relative;  
   
}  
  
.page-title {  
 font-size: 26px;  
 font-weight: bold;  
 margin-bottom: 5px;  
}  
  
.page-subtitle {  
 font-size: 14px;  
 color: var(--text-muted);  
}  
  
/\* Search & Sort \*/  
.search-sort-container {  
 display: flex;  
 justify-content: space-between;  
 gap: 20px;  
 margin-bottom: 30px;  
}  
  
.search-bar {  
 flex: 1;  
 position: relative;  
}  
  
.search-bar input {  
 width: 100%;  
 padding: 12px 15px 12px 40px;  
 border-radius: var(--radius);  
 border: 1px solid var(--card-border);  
 background-color: white;  
 font-size: 14px;  
}  
  
.search-icon {  
 position: absolute;  
 left: 15px;  
 top: 50%;  
 transform: translateY(-50%);  
 color: var(--text-muted);  
}  
  
.sort-container {  
 display: flex;  
 align-items: center;  
 gap: 10px;  
}  
  
.sort-label {  
 font-size: 14px;  
 font-weight: 500;  
 color: var(--text-muted);  
}  
  
.sort-select {  
 padding: 10px 15px;  
 border-radius: var(--radius);  
 border: 1px solid var(--card-border);  
 background-color: white;  
 font-size: 14px;  
 font-weight: 500;  
}  
  
/\* Cards - Grid Layout \*/  
.venue-grid {  
 display: grid;  
 grid-template-columns: repeat(3, 1fr); /\* Fixed 3 columns \*/  
 gap: 20px;  
}  
  
.venue-card {  
 background: var(--card-bg);  
 border-radius: var(--radius);  
 box-shadow: 0 4px 12px var(--shadow);  
 border: 1px solid var(--card-border);  
 overflow: hidden;  
 transition: transform 0.3s ease, box-shadow 0.3s ease;  
}  
  
.venue-card:hover {  
 transform: translateY(-5px);  
 box-shadow: 0 8px 20px rgba(30, 144, 255, 0.15);  
}  
  
.venue-image {  
 width: 100%;  
 height: 160px;  
 overflow: hidden;  
}  
  
.venue-image img {  
 width: 100%;  
 height: 100%;  
 object-fit: cover;  
 transition: transform 0.5s ease;  
}  
  
.venue-card:hover .venue-image img {  
 transform: scale(1.05);  
}  
  
.venue-details {  
 padding: 16px;  
}  
  
.venue-name {  
 font-size: 18px;  
 font-weight: 600;  
 margin-bottom: 5px;  
 display: flex;  
 justify-content: space-between;  
 align-items: center;  
 text-transform: capitalize;  
  
}  
  
.venue-status {  
 padding: 4px 10px;  
 border-radius: 20px;  
 font-size: 12px;  
 font-weight: 500;  
 display: inline-block;  
}  
  
.status-available {  
 background-color: rgba(34, 197, 94, 0.1);  
 color: var(--secondary);  
}  
  
.status-unavailable {  
 background-color: rgba(239, 68, 68, 0.1);  
 color: #ef4444;  
}  
  
.venue-info {  
 margin: 12px 0;  
 font-size: 14px;  
 color: var(--text-main);  
}  
  
.venue-info p {  
 margin-bottom: 6px;  
 display: flex;  
 align-items: center;  
}  
  
.venue-info i {  
 margin-right: 8px;  
 color: var(--primary);  
}  
  
.venue-rating {  
 display: flex;  
 align-items: center;  
 margin-bottom: 6px;  
}  
  
.star {  
 color: #fbbf24;  
 margin-right: 5px;  
}  
  
.venue-price {  
 font-weight: 600;  
 color: var(--primary);  
}  
  
.venue-actions {  
 display: flex;  
 justify-content: space-between;  
 align-items: center;  
 margin-top: 15px;  
}  
  
.action-buttons {  
 display: flex;  
 gap: 10px;  
}  
  
.action-btn {  
 padding: 8px 16px;  
 border-radius: var(--radius);  
 font-size: 13px;  
 font-weight: 500;  
 border: none;  
 cursor: pointer;  
 transition: all 0.3s ease;  
}  
  
.action-btn.edit {  
 background-color: rgba(30, 144, 255, 0.1);  
 color: var(--primary);  
}  
  
.action-btn.edit:hover {  
 background-color: rgba(30, 144, 255, 0.2);  
}  
  
.action-btn.delete {  
 background-color: rgba(239, 68, 68, 0.1);  
 color: #ef4444;  
}  
  
.action-btn.delete:hover {  
 background-color: rgba(239, 68, 68, 0.2);  
}  
  
.toggle-switch {  
 position: relative;  
 display: inline-block;  
 width: 46px;  
 height: 24px;  
}  
  
.toggle-switch input {  
 opacity: 0;  
 width: 0;  
 height: 0;  
}  
  
.toggle-slider {  
 position: absolute;  
 top: 0;  
 left: 0;  
 right: 0;  
 bottom: 0;  
 background-color: #ccc;  
 transition: .4s;  
 border-radius: 34px;  
 cursor: pointer;  
}  
  
.toggle-slider:before {  
 content: "";  
 position: absolute;  
 height: 18px;  
 width: 18px;  
 left: 3px;  
 bottom: 3px;  
 background-color: white;  
 transition: .4s;  
 border-radius: 50%;  
}  
  
input:checked + .toggle-slider {  
 background: linear-gradient(to right, var(--primary), var(--secondary));  
}  
  
input:checked + .toggle-slider:before {  
 transform: translateX(22px);  
}  
  
/\* Button \*/  
.add-venue-btn {  
 padding: 12px 24px;  
 border-radius: var(--radius);  
 background: linear-gradient(to right, var(--primary), var(--secondary));  
 color: white;  
 border: none;  
 cursor: pointer;  
 font-weight: 600;  
 font-size: 14px;  
 box-shadow: 0 4px 12px var(--shadow);  
 transition: 0.3s;  
 position: absolute;  
 right: 0;  
 top: 0;  
}  
  
.add-venue-btn:hover {  
 transform: translateY(-3px);  
 box-shadow: 0 6px 15px rgba(30, 144, 255, 0.2);  
}  
  
/\* Responsive adjustments \*/  
@media (max-width: 1024px) {  
 .venue-grid {  
 grid-template-columns: repeat(2, 1fr); /\* 2 columns on medium screens \*/  
 }  
   
 .search-sort-container {  
 flex-direction: column;  
 }  
   
 .add-venue-btn {  
 position: relative;  
 margin-top: 15px;  
 width: 100%;  
 }  
   
 .page-header {  
 text-align: center;  
 }  
}  
  
@media (max-width: 600px) {  
 .venue-grid {  
 grid-template-columns: 1fr; /\* 1 column on small screens \*/  
 }  
   
 .fresh-navbar {  
 flex-direction: column;  
 gap: 15px;  
 padding: 15px;  
 }  
   
 .nav-links {  
 flex-wrap: wrap;  
 justify-content: center;  
 }  
   
}.modal-overlay {  
 position: fixed;  
 inset: 0;  
 background: rgba(15, 23, 42, 0.5);  
 display: none;  
 justify-content: center;  
 align-items: center;  
 z-index: 9999;  
}  
.form-group {  
 margin-bottom: 20px;  
}  
  
  
.modal-content {  
 margin-bottom: 20px; /\* distance between title and next tag \*/  
  
 background: white;  
 padding: 30px;  
 border-radius: var(--radius);  
 width: 90%;  
 max-width: 650px;  
 position: relative;  
 box-shadow: 0 10px 40px rgba(0, 0, 0, 0.2);  
}  
.modal-content h2 {  
 margin-bottom: 30px; /\* distance between title and "Venue Name" \*/  
 font-size: 24px;  
 font-weight: 700;  
 color: var(--text-main);  
}  
  
  
.close-modal {  
 position: absolute;  
 top: 16px;  
 right: 24px;  
 font-size: 24px;  
 cursor: pointer;  
 color: var(--text-main);  
}  
  
.modal-content form input,  
.modal-content form select,  
.modal-content form textarea {  
 width: 100%;  
 padding: 12px;  
 border: 1px solid var(--card-border);  
 border-radius: var(--radius);  
 margin-top: 5px;  
 margin-bottom: 15px;  
}  
  
.image-upload {  
 background: #f9fafb;  
 padding: 16px;  
 text-align: center;  
 border: 1px dashed #ccc;  
 border-radius: var(--radius);  
}  
  
.map-placeholder {  
 background: #e2e8f0;  
 height: 200px;  
 border-radius: var(--radius);  
 margin-top: 10px;  
}.form-group {  
 margin-bottom: 15px;  
}  
  
.modal-buttons {  
 display: flex;  
 justify-content: flex-end;  
 gap: 10px;  
 margin-top: 20px;  
}  
  
.save-btn, .cancel-btn {  
 padding: 10px 20px;  
 border-radius: var(--radius);  
 font-weight: 600;  
 font-size: 14px;  
 border: none;  
 cursor: pointer;  
 box-shadow: 0 4px 12px var(--shadow);  
 transition: 0.3s;  
}  
  
.save-btn {  
 background: linear-gradient(to right, var(--primary), var(--secondary));  
 color: white;  
}  
  
.save-btn:hover {  
 transform: translateY(-2px);  
 box-shadow: 0 6px 15px rgba(30, 144, 255, 0.2);  
}  
.cancel-btn {  
 background: linear-gradient(to right, #fca5a5, #f87171);  
 color: white;  
 border: none;  
 padding: 10px 20px;  
 border-radius: var(--radius);  
 font-weight: 600;  
 font-size: 14px;  
 cursor: pointer;  
 transition: 0.3s ease;  
 box-shadow: 0 2px 10px rgba(248, 113, 113, 0.25);  
}  
  
.cancel-btn:hover {  
 background: linear-gradient(to right, #f87171, #ef4444);  
 transform: translateY(-2px);  
}  
.image-preview-container {  
 display: flex;  
 gap: 10px;  
 margin-top: 10px;  
 flex-wrap: wrap;  
}  
  
.image-preview-container img {  
 width: 100px;  
 height: 100px;  
 object-fit: cover;  
 border-radius: 10px;  
 border: 1px solid #ddd;  
}  
/\* Add these new styles to your existing CSS file \*/  
  
.message {  
 background-color: #f8f9fa;  
 border-left: 4px solid #1e90ff;  
 margin-bottom: 20px;  
 padding: 10px 15px;  
 border-radius: 0 5px 5px 0;  
 font-weight: 500;  
}  
  
.no-venues-message {  
 text-align: center;  
 padding: 40px;  
 background: rgba(255, 255, 255, 0.7);  
 border-radius: 12px;  
 font-size: 18px;  
 color: #64748b;  
 grid-column: 1 / -1;  
}  
  
.image-preview-container {  
 display: flex;  
 gap: 10px;  
 margin-top: 10px;  
 flex-wrap: wrap;  
}  
  
.image-preview-container img {  
 width: 100px;  
 height: 100px;  
 object-fit: cover;  
 border-radius: 10px;  
 border: 1px solid #ddd;  
}  
/\* Add these styles to your ManageVenue.css file \*/  
  
/\* Modal scrolling fixes \*/  
.modal-overlay {  
 position: fixed;  
 inset: 0;  
 background: rgba(15, 23, 42, 0.5);  
 display: none;  
 justify-content: center;  
 align-items: flex-start; /\* Align to top instead of center \*/  
 z-index: 9999;  
 overflow-y: auto; /\* Allow vertical scrolling \*/  
 padding: 20px;  
}  
  
.modal-content {  
 background: white;  
 padding: 30px;  
 border-radius: var(--radius);  
 width: 90%;  
 max-width: 650px;  
 max-height: 85vh; /\* Set maximum height to 85% of viewport height \*/  
 position: relative;  
 box-shadow: 0 10px 40px rgba(0, 0, 0, 0.2);  
 overflow-y: auto; /\* Allow the content to scroll \*/  
 margin: 30px auto; /\* Center the modal with some top margin \*/  
}  
  
/\* Make sure buttons stay visible at the bottom \*/  
.modal-buttons {  
 position: sticky;  
 bottom: 0;  
 background: white;  
 padding-top: 15px;  
 margin-top: 15px;  
 border-top: 1px solid #f0f0f0;  
 z-index: 1;  
}  
  
/\* Better spacing for form fields \*/  
.modal-content form input,  
.modal-content form select,  
.modal-content form textarea {  
 margin-bottom: 10px;  
}  
  
/\* Label spacing \*/  
.modal-content form label {  
 display: block;  
 margin-top: 10px;  
 margin-bottom: 5px;  
 font-weight: 500;  
}  
  
/\* Add more bottom space to last form element before buttons \*/  
.form-group:last-of-type {  
 margin-bottom: 20px;  
}  
  
/\* Responsive adjustment for smaller screens \*/  
@media (max-height: 700px) {  
 .modal-content {  
 max-height: 90vh;  
 margin-top: 10px;  
 }  
}  
#map {  
 height: 300px;  
 border-radius: var(--radius);  
 margin-top: 10px;  
 border: 1px solid var(--card-border);  
 box-shadow: 0 2px 8px rgba(0, 0, 0, 0.05);  
}  
  
.map-help {  
 font-size: 12px;  
 color: var(--text-muted);  
 margin-top: 5px;  
 text-align: center;  
 font-style: italic;  
}  
  
#locationInput {  
 margin-bottom: 10px;  
}  
  
/\* Styling for map buttons \*/  
.gm-style .gm-style-iw {  
 font-family: 'Outfit', sans-serif;  
}  
  
/\* Improve positioning on small screens \*/  
@media (max-width: 768px) {  
 #map {  
 height: 250px;  
 }  
}

————————————————————————————————————————

## ManageVenue\ManageVenue.html

<!DOCTYPE html>  
<html lang="en">  
<head>  
 <meta charset="UTF-8">  
 <meta name="viewport" content="width=device-width, initial-scale=1.0">  
 <title>Book&Play - Venue Management</title>  
 <link rel="preconnect" href="https://fonts.googleapis.com">  
 <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>  
 <link rel="stylesheet" href="ManageVenue.css" />  
 <link href="https://fonts.googleapis.com/css2?family=Outfit:wght@300;400;500;600;700&display=swap" rel="stylesheet">  
 <script async defer src="https://maps.googleapis.com/maps/api/js?key=AIzaSyCdDZbnpmM7I-vaVXq6N0\_hiJQmHNL5hpk&libraries=places,marker&callback=initializeApp"></script>  
  
</head>  
<body>  
  
  
 <!-- Main Content -->  
 <div class="container">  
 <div class="page-header">  
 <h1 class="page-title">Manage Your Venues</h1>  
 <p class="page-subtitle">Control and optimize your sports facilities with ease</p>  
 <button class="add-venue-btn">Add New Venue +</button>  
 </div>  
  
 <!-- Search and Sort -->  
 <div class="search-sort-container">  
 <div class="search-bar">  
 <span class="search-icon">🔍</span>  
 <input type="text" id="searchInput" placeholder="Search venues by name">  
 </div>  
 <div class="sort-container">  
 <span class="sort-label">Sort By:</span>  
 <select class="sort-select" id="sortSelect">  
 <option value="name">Name (A-Z)</option>  
 <option value="name-desc">Name (Z-A)</option>  
  
 </select>  
 </div>  
 <div class="sort-container">  
 <select class="sort-select" id="sortbysport">  
   
 </select>  
 </div>  
 </div>  
  
 <!-- Venue Grid -->  
 <div class="venue-grid" id="venueGrid">  
 <!-- Venues will be dynamically populated here via JavaScript -->  
 </div>  
 </div>  
  
 <!-- Modal -->  
 <div id="venueModal" class="modal-overlay">  
 <div class="modal-content">  
 <span class="close-modal" id="closeModal">&times;</span>  
 <h2 id="modalTitle">Add New Venue</h2>  
 <form method="POST" enctype="multipart/form-data">  
 <input type="hidden" name="facility\_id" id="facilityId" value="">  
 <input type="hidden" name="save\_facility" value="1">  
 <!-- Hidden fields to store coordinates -->  
 <input type="hidden" name="latitude" id="latitude" value="">  
 <input type="hidden" name="longitude" id="longitude" value="">  
   
 <label>Venue Name</label>  
 <input type="text" name="place\_name" id="placeName" placeholder="Enter venue name" required>  
   
 <label>Sport Type</label>  
 <select name="sport\_type" id="sportType" required>  
 <!-- Sports will be populated via JavaScript -->  
 </select>  
   
 <label>Venue Images</label>  
 <input type="file" id="venueImages" name="venueImages[]" accept="image/\*" multiple>  
 <div id="imagePreview" class="image-preview-container"></div>  
   
 <label>Description</label>  
 <textarea name="description" id="description" placeholder="Enter venue description"></textarea>  
   
 <label>Hourly Price ($)</label>  
 <input type="number" name="price" id="price" placeholder="50" min="1" required>  
   
 <div class="form-group">  
 <label>Availability</label>  
 <label class="toggle-switch">  
 <input type="checkbox" name="is\_available" id="isAvailable" checked>  
 <span class="toggle-slider"></span>  
 </label>  
 </div>  
   
 <div class="form-group">  
 <label for="locationInput">Location</label>  
 <input type="text" id="locationInput" name="location" placeholder="Enter address or select on map">  
 <div id="map" style="height: 300px; border-radius: 12px; margin-top: 10px;"></div>  
 <p class="map-help">You can click on the map to set location or drag the marker for precise positioning</p>  
 </div>  
   
 <div class="modal-buttons">  
 <button type="button" class="cancel-btn">Cancel</button>  
 <button type="submit" class="save-btn">Save</button>  
 </div>  
 </form>  
 </div>  
 </div>  
   
 <!-- Load the JavaScript file -->  
 <script src="ManageVenue.js"></script>  
</body>  
</html>

————————————————————————————————————————

## ManageVenue\ManageVenue.js

// Global variables  
let venuesData = [];  
let sportsData = [];  
let map;  
let marker;  
let geocoder;  
let currentEditingId = null;  
  
// DOM elements  
const venueGrid = document.getElementById('venueGrid');  
const venueModal = document.getElementById('venueModal');  
const addVenueBtn = document.querySelector('.add-venue-btn');  
const closeModalBtn = document.getElementById('closeModal');  
const cancelBtn = document.querySelector('.cancel-btn');  
const searchInput = document.getElementById('searchInput');  
const sortSelect = document.getElementById('sortSelect');  
const sortBySportSelect = document.getElementById('sortbysport');  
const modalTitle = document.getElementById('modalTitle');  
const venueForm = document.querySelector('form');  
const imageInput = document.getElementById('venueImages');  
const imagePreview = document.getElementById('imagePreview');  
  
async function initializeApp() {  
 window.initializeApp = initializeApp;  
 try {  
 await loadSports();  
 await loadVenues();  
 initializeMap();  
 setupEventListeners();  
 populateSportsFilter();  
 } catch (error) {  
 console.error('Error initializing app:', error);  
 showMessage('Error loading data. Please refresh the page.', 'error');  
 }  
}  
async function initMap() {  
 await google.maps.importLibrary("places");  
 const placeAutocomplete = new google.maps.places.PlaceAutocompleteElement();  
 const container = document.getElementById("autocompleteElement");  
 container.replaceWith(placeAutocomplete);  
  
 const selectedLatInput = document.getElementById("latitude");  
 const selectedLngInput = document.getElementById("longitude");  
  
 placeAutocomplete.addEventListener("gmp-select", async ({ placePrediction }) => {  
 const place = placePrediction.toPlace();  
 await place.fetchFields({ fields: ["displayName", "formattedAddress", "location"] });  
  
 const location = place.location;  
 const lat = location.lat;  
 const lng = location.lng;  
  
 if (lat && lng) {  
 const latLng = { lat, lng };  
  
 map.setCenter(latLng);  
 marker.position = latLng;  
  
 selectedLatInput.value = lat;  
 selectedLngInput.value = lng;  
 }  
 });  
}  
  
// Load sports data from server  
async function loadSports() {  
 try {  
 const response = await fetch('fetch\_venues.php?action=get\_sports');  
 if (!response.ok) throw new Error('Failed to fetch sports');  
   
 sportsData = await response.json();  
 populateSportsDropdown();  
 } catch (error) {  
 console.error('Error loading sports:', error);  
 showMessage('Error loading sports data', 'error');  
 }  
}  
  
// Load venues data from server  
async function loadVenues() {  
 try {  
 const response = await fetch('fetch\_venues.php?action=get\_facilities');  
 if (!response.ok) throw new Error('Failed to fetch venues');  
   
 venuesData = await response.json();  
 displayVenues(venuesData);  
 } catch (error) {  
 console.error('Error loading venues:', error);  
 showMessage('Error loading venues data', 'error');  
 }  
}  
  
// Populate sports dropdown in modal  
function populateSportsDropdown() {  
 const sportSelect = document.getElementById('sportType');  
 sportSelect.innerHTML = '<option value="">Select Sport Type</option>';  
   
 sportsData.forEach(sport => {  
 const option = document.createElement('option');  
 option.value = sport.sport\_name;  
 option.textContent = sport.sport\_name;  
 sportSelect.appendChild(option);  
 });  
}  
  
// Populate sports filter dropdown  
function populateSportsFilter() {  
 sortBySportSelect.innerHTML = '<option value="">All Sports</option>';  
   
 sportsData.forEach(sport => {  
 const option = document.createElement('option');  
 option.value = sport.sport\_name;  
 option.textContent = sport.sport\_name;  
 sortBySportSelect.appendChild(option);  
 });  
}  
  
// Initialize Google Maps  
function initializeMap() {  
 const defaultLocation = { lat: 32.0853, lng: 34.7818 };  
  
 map = new google.maps.Map(document.getElementById('map'), {  
 zoom: 13,  
 center: defaultLocation,  
 mapId: 'bec4fd1eef54f71c3e99df68',  
 mapTypeControl: false,  
 streetViewControl: false,  
 fullscreenControl: false  
 });  
  
 marker = new google.maps.marker.AdvancedMarkerElement({  
 position: defaultLocation,  
 map: map,  
 title: 'Venue Location'  
 });  
  
 geocoder = new google.maps.Geocoder();  
  
 const locationInput = document.getElementById('locationInput');  
 const autocomplete = new google.maps.places.Autocomplete(locationInput);  
 autocomplete.bindTo('bounds', map);  
  
 autocomplete.addListener('place\_changed', function () {  
 const place = autocomplete.getPlace();  
   
 // ✅ Protection against manual input without selecting suggestion  
 if (!place.geometry || !place.geometry.location) {  
 alert("❌ Please select the location from the list that appears, don't just type the address manually.");  
 document.getElementById('locationInput').value = '';  
 return;  
 }  
  
 const location = place.geometry.location;  
 map.setCenter(location);  
 marker.position = location;  
 updateCoordinates(location.lat(), location.lng());  
});  
  
  
 map.addListener('click', function(event) {  
 updateMarkerPosition(event.latLng);  
 });  
}  
  
// Update marker position and coordinates  
function updateMarkerPosition(latLng) {  
 marker.position = latLng;  
 updateCoordinates(latLng.lat(), latLng.lng());  
 geocoder.geocode({ location: latLng }, function(results, status) {  
 if (status === 'OK' && results[0]) {  
 document.getElementById('locationInput').value = results[0].formatted\_address;  
 }  
 });  
}  
  
// Update hidden coordinate fields  
function updateCoordinates(lat, lng) {  
 document.getElementById('latitude').value = lat;  
 document.getElementById('longitude').value = lng;  
}  
  
// Setup event listeners  
function setupEventListeners() {  
 // Modal controls  
 addVenueBtn.addEventListener('click', () => openModal());  
 closeModalBtn.addEventListener('click', () => closeModal());  
 cancelBtn.addEventListener('click', () => closeModal());  
   
 // Click outside modal to close  
 venueModal.addEventListener('click', function(e) {  
 if (e.target === venueModal) {  
 closeModal();  
 }  
 });  
   
 // Form submission  
 venueForm.addEventListener('submit', handleFormSubmit);  
   
 // Search and filter  
 searchInput.addEventListener('input', handleSearch);  
 sortSelect.addEventListener('change', handleSort);  
 sortBySportSelect.addEventListener('change', handleSportFilter);  
   
 // Image preview  
 imageInput.addEventListener('change', handleImagePreview);  
}  
  
// Display venues in grid  
function displayVenues(venues) {  
 if (venues.length === 0) {  
 venueGrid.innerHTML = `  
 <div class="no-venues-message">  
 <p>No venues found. Click "Add New Venue" to get started!</p>  
 </div>  
 `;  
 return;  
 }  
   
 venueGrid.innerHTML = venues.map(venue => createVenueCard(venue)).join('');  
   
 // Add event listeners to toggle switches and action buttons  
 setupVenueCardEventListeners();  
}  
  
// Create venue card HTML  
function createVenueCard(venue) {  
 const images = venue.image\_url ? venue.image\_url.split(',') : [];  
 const firstImage = images.length > 0 ? images[0] : 'placeholder.jpg';  
 const isAvailable = venue.is\_available == 1;  
   
 return `  
 <div class="venue-card" data-venue-id="${venue.facilities\_id}">  
 <div class="venue-image">  
 <img src="${firstImage}" alt="${venue.place\_name}" onerror="this.src='../../../Images/default.jpg'">  
 </div>  
 <div class="venue-details">  
 <div class="venue-name">  
 <span>${venue.place\_name}</span>  
 <span class="venue-status ${isAvailable ? 'status-available' : 'status-unavailable'}">  
 ${isAvailable ? 'Available' : 'Unavailable'}  
 </span>  
 </div>  
 <div class="venue-info">  
 <p><i>🏃</i><strong>Sport:</strong> ${venue.SportCategory}</p>  
 <p><i>📍</i><strong>Location:</strong> ${venue.location}</p>  
 <p><i>💰</i><strong>Price:</strong> ${venue.price}/hour</p>  
 </div>  
 <div class="venue-actions">  
 <div class="action-buttons">  
 <button class="action-btn edit" onclick="editVenue(${venue.facilities\_id})">✏️ Edit</button>  
 </div>  
 <label class="toggle-switch">  
 <input type="checkbox" ${isAvailable ? 'checked' : ''}   
 onchange="toggleAvailability(${venue.facilities\_id}, this.checked)">  
 <span class="toggle-slider"></span>  
 </label>  
 </div>  
 </div>  
 </div>  
 `;  
}  
  
// Setup event listeners for venue cards  
function setupVenueCardEventListeners() {  
 // Event listeners are handled inline in the HTML for simplicity  
 // This function can be expanded if needed for more complex event handling  
}  
  
// Open modal for adding/editing venue  
function openModal(venue = null) {  
 currentEditingId = venue ? venue.facilities\_id : null;  
 modalTitle.textContent = venue ? 'Edit Venue' : 'Add New Venue';  
   
 // Reset form  
 venueForm.reset();  
 imagePreview.innerHTML = '';  
   
 if (venue) {  
 // Populate form with venue data  
 document.getElementById('facilityId').value = venue.facilities\_id;  
 document.getElementById('placeName').value = venue.place\_name;  
 document.getElementById('sportType').value = venue.SportCategory;  
 document.getElementById('description').value = venue.description || '';  
 document.getElementById('price').value = venue.price;  
 document.getElementById('locationInput').value = venue.location;  
 document.getElementById('isAvailable').checked = venue.is\_available == 1;  
   
 // Set coordinates if available  
 if (venue.latitude && venue.longitude) {  
 const lat = parseFloat(venue.latitude);  
 const lng = parseFloat(venue.longitude);  
 document.getElementById('latitude').value = lat;  
 document.getElementById('longitude').value = lng;  
   
 // Update map  
 const location = new google.maps.LatLng(lat, lng);  
 map.setCenter(location);  
 marker.position = location;  
 }  
   
 // Show existing images  
 if (venue.image\_url) {  
 const images = venue.image\_url.split(',');  
 images.forEach(imageUrl => {  
 if (imageUrl.trim()) {  
 const img = document.createElement('img');  
 img.src = imageUrl.trim();  
 img.style.width = '100px';  
 img.style.height = '100px';  
 img.style.objectFit = 'cover';  
 img.style.borderRadius = '10px';  
 imagePreview.appendChild(img);  
 }  
 });  
 }  
 } else {  
 // Reset for new venue  
 document.getElementById('facilityId').value = '';  
 // Reset map to default location for new venue  
 const defaultLocation = { lat: 32.0853, lng: 34.7818 };  
 map.setCenter(defaultLocation);  
 marker.position = defaultLocation;  
 updateCoordinates(defaultLocation.lat, defaultLocation.lng);  
 }  
   
 venueModal.style.display = 'flex';  
 document.body.style.overflow = 'hidden'; // Prevent background scrolling  
}  
  
// Close modal  
function closeModal() {  
 venueModal.style.display = 'none';  
 document.body.style.overflow = 'auto'; // Restore scrolling  
 currentEditingId = null;  
}  
  
// Handle form submission  
async function handleFormSubmit(e) {  
 e.preventDefault();  
   
 const formData = new FormData(venueForm);  
   
 // Determine if we're editing or adding  
 if (currentEditingId) {  
 formData.append('action', 'update\_facility');  
 formData.append('facility\_id', currentEditingId);  
 } else {  
 formData.append('action', 'add\_facility');  
 }  
   
 try {  
 const response = await fetch('fetch\_venues.php', {  
 method: 'POST',  
 body: formData  
 });  
   
 if (!response.ok) {  
 throw new Error('Network response was not ok');  
 }  
   
 const result = await response.json();  
   
 if (result.success) {  
 showMessage(result.message, 'success');  
 closeModal();  
 await loadVenues(); // Reload venues  
 } else {  
 showMessage(result.message, 'error');  
 }  
 } catch (error) {  
 console.error('Error saving venue:', error);  
 showMessage('Error saving venue. Please try again.', 'error');  
 }  
}  
  
// Handle image preview  
function handleImagePreview(e) {  
 const files = Array.from(e.target.files);  
 imagePreview.innerHTML = '';  
   
 // Limit to 3 images  
 const limitedFiles = files.slice(0, 3);  
   
 limitedFiles.forEach(file => {  
 if (file.type.startsWith('image/')) {  
 const reader = new FileReader();  
 reader.onload = function(e) {  
 const img = document.createElement('img');  
 img.src = e.target.result;  
 img.style.width = '100px';  
 img.style.height = '100px';  
 img.style.objectFit = 'cover';  
 img.style.borderRadius = '10px';  
 img.style.border = '1px solid #ddd';  
 imagePreview.appendChild(img);  
 };  
 reader.readAsDataURL(file);  
 }  
 });  
   
 if (files.length > 3) {  
 showMessage('Only the first 3 images will be uploaded.', 'warning');  
 }  
}  
  
// Edit venue function  
async function editVenue(venueId) {  
 const venue = venuesData.find(v => v.facilities\_id == venueId);  
 if (venue) {  
 openModal(venue);  
 }  
}  
  
// Delete venue function  
async function deleteVenue(venueId) {  
 if (!confirm('Are you sure you want to delete this venue? This action cannot be undone.')) {  
 return;  
 }  
   
 try {  
 // Note: You'll need to implement delete functionality in your PHP  
 // For now, we'll just remove it from the display  
 venuesData = venuesData.filter(v => v.facilities\_id != venueId);  
 displayVenues(venuesData);  
 showMessage('Venue deleted successfully', 'success');  
 } catch (error) {  
 console.error('Error deleting venue:', error);  
 showMessage('Error deleting venue', 'error');  
 }  
}  
  
// Toggle venue availability  
async function toggleAvailability(venueId, isAvailable) {  
 try {  
 const formData = new FormData();  
 formData.append('action', 'update\_availability');  
 formData.append('facility\_id', venueId);  
 formData.append('is\_available', isAvailable ? 1 : 0);  
   
 const response = await fetch('fetch\_venues.php', {  
 method: 'POST',  
 body: formData  
 });  
   
 const result = await response.json();  
   
 if (result.success) {  
 // Update local data without reloading  
 const venue = venuesData.find(v => v.facilities\_id == venueId);  
 if (venue) {  
 venue.is\_available = isAvailable ? 1 : 0;  
   
 // Update only the specific venue card's status display  
 updateVenueCardStatus(venueId, isAvailable);  
 }  
 showMessage(result.message, 'success');  
 } else {  
 showMessage(result.message, 'error');  
 // Revert the toggle  
 const checkbox = document.querySelector(`input[onchange\*="${venueId}"]`);  
 if (checkbox) {  
 checkbox.checked = !isAvailable;  
 }  
 }  
 } catch (error) {  
 console.error('Error updating availability:', error);  
 showMessage('Error updating availability', 'error');  
 // Revert the toggle  
 const checkbox = document.querySelector(`input[onchange\*="${venueId}"]`);  
 if (checkbox) {  
 checkbox.checked = !isAvailable;  
 }  
 }  
}  
  
// Update specific venue card status without reloading all cards  
function updateVenueCardStatus(venueId, isAvailable) {  
 const venueCard = document.querySelector(`[data-venue-id="${venueId}"]`);  
 if (venueCard) {  
 const statusElement = venueCard.querySelector('.venue-status');  
 if (statusElement) {  
 // Update status text and class  
 statusElement.textContent = isAvailable ? 'Available' : 'Unavailable';  
 statusElement.className = `venue-status ${isAvailable ? 'status-available' : 'status-unavailable'}`;  
 }  
 }  
}  
  
// Handle search  
function handleSearch() {  
 displayVenues(getFilteredVenues());  
}  
  
// Handle sorting  
function handleSort() {  
 displayVenues(getFilteredVenues());  
}  
  
// Handle sport filter  
function handleSportFilter() {  
 displayVenues(getFilteredVenues());  
}  
  
// Get filtered and sorted venues  
function getFilteredVenues() {  
 let filteredVenues = [...venuesData];  
  
 // Apply search filter (startsWith only for place\_name)  
 const searchTerm = searchInput.value.trim().toLowerCase();  
 if (searchTerm !== '') {  
 filteredVenues = filteredVenues.filter(venue =>  
 venue.place\_name.toLowerCase().startsWith(searchTerm)  
 );  
 }  
  
 // Apply sport filter  
 const selectedSport = sortBySportSelect.value;  
 if (selectedSport) {  
 filteredVenues = filteredVenues.filter(venue =>  
 venue.SportCategory === selectedSport  
 );  
 }  
  
 return applyFiltersAndSort(filteredVenues);  
}  
  
  
  
// Apply filters and sorting  
function applyFiltersAndSort(venues) {  
 const sortBy = sortSelect.value;  
   
 return venues.sort((a, b) => {  
 switch (sortBy) {  
 case 'name':  
 return a.place\_name.localeCompare(b.place\_name);  
 case 'name-desc':  
 return b.place\_name.localeCompare(a.place\_name);  
 case 'price':  
 return parseFloat(a.price) - parseFloat(b.price);  
 case 'price-desc':  
 return parseFloat(b.price) - parseFloat(a.price);  
 default:  
 return 0;  
 }  
 });  
}  
  
// Show message to user  
function showMessage(message, type = 'info') {  
 // Remove existing messages  
 const existingMessage = document.querySelector('.message');  
 if (existingMessage) {  
 existingMessage.remove();  
 }  
   
 // Create new message  
 const messageDiv = document.createElement('div');  
 messageDiv.className = `message ${type}`;  
 messageDiv.textContent = message;  
   
 // Style based on type  
 switch (type) {  
 case 'success':  
 messageDiv.style.borderLeftColor = '#22c55e';  
 messageDiv.style.backgroundColor = '#f0f9ff';  
 break;  
 case 'error':  
 messageDiv.style.borderLeftColor = '#ef4444';  
 messageDiv.style.backgroundColor = '#fef2f2';  
 break;  
 case 'warning':  
 messageDiv.style.borderLeftColor = '#f59e0b';  
 messageDiv.style.backgroundColor = '#fffbeb';  
 break;  
 default:  
 messageDiv.style.borderLeftColor = '#1e90ff';  
 messageDiv.style.backgroundColor = '#f8f9fa';  
 }  
   
 // Insert at the top of the container  
 const container = document.querySelector('.container');  
 container.insertBefore(messageDiv, container.firstChild);  
   
 // Auto-remove after 5 seconds  
 setTimeout(() => {  
 if (messageDiv.parentNode) {  
 messageDiv.remove();  
 }  
 }, 5000);  
}  
  
// Utility function to handle errors  
function handleError(error, context = '') {  
 console.error(`Error ${context}:`, error);  
 showMessage(`An error occurred ${context}. Please try again.`, 'error');  
}  
  
// Initialize tooltips or other UI enhancements  
function initializeUIEnhancements() {  
 // Add any additional UI enhancements here  
 // For example, tooltips, animations, etc.  
}  
  
// Call UI enhancements after DOM is loaded  
document.addEventListener('DOMContentLoaded', function() {  
 initializeUIEnhancements();  
});

————————————————————————————————————————

## ManageVenue\ManageVenue.php

<?php  
session\_start();  
if (!isset($\_SESSION['user\_id'])) {  
 // Unset all session variables  
 session\_unset();  
 // Destroy the session completely  
 session\_destroy();  
 header('Location: ../../auth/Login\_Page/Login.php');  
 exit();  
}  
include '../navbar.php';  
include 'ManageVenue.html';  
exit();  
?>

————————————————————————————————————————

## ManageVenue\upload\_images.php

<?php  
/\*\*  
 \* Function to handle multiple image uploads for sport facilities  
 \* Allows uploading up to 3 images and stores them in the uploads/venues directory  
 \*/  
function upload\_images($files) {  
 // Define the upload directory path (server path)  
 $upload\_dir = '../../../uploads/venues/';  
  
 // Public URL path to return (relative to web root)  
 $public\_path = '../../../uploads/venues/';  
  
 // Create the uploads directory if it doesn't exist  
 if (!file\_exists($upload\_dir)) {  
 mkdir($upload\_dir, 0777, true);  
 }  
  
 $uploaded\_files = [];  
 $allowed\_types = ['image/jpeg', 'image/png', 'image/gif', 'image/jpg', 'image/webp'];  
 $max\_size = 5 \* 1024 \* 1024; // 5MB  
  
 $file\_count = min(count($files['name']), 3); // Limit to 3 files  
  
 for ($i = 0; $i < $file\_count; $i++) {  
 if ($files['error'][$i] !== UPLOAD\_ERR\_OK) {  
 continue;  
 }  
  
 $file\_type = $files['type'][$i];  
 if (!in\_array($file\_type, $allowed\_types)) {  
 continue;  
 }  
  
 if ($files['size'][$i] > $max\_size) {  
 continue;  
 }  
  
 $new\_filename = uniqid() . '\_' . basename($files['name'][$i]);  
 $destination = $upload\_dir . $new\_filename;  
  
 if (move\_uploaded\_file($files['tmp\_name'][$i], $destination)) {  
 // ✅ Store relative web path, not full server path  
 $uploaded\_files[] = $public\_path . $new\_filename;  
 }  
 }  
  
 return $uploaded\_files;  
}  
?>

————————————————————————————————————————

## Messages\fetch\_conversations.php

<?php  
session\_start();  
require\_once '../../../db.php'; // change path according to your project  
  
header('Content-Type: application/json');  
  
if (!isset($\_SESSION['user\_id'])) {  
 echo json\_encode(["success" => false, "message" => "Not logged in"]);  
 exit;  
}  
  
$currentUser = $\_SESSION['user\_id'];  
  
$sql = "SELECT DISTINCT   
 CASE   
 WHEN sender\_username = ? THEN receiver\_username   
 ELSE sender\_username   
 END AS chat\_partner  
 FROM messages  
 WHERE sender\_username = ? OR receiver\_username = ?";  
  
$stmt = $conn->prepare($sql);  
$stmt->bind\_param("sss", $currentUser, $currentUser, $currentUser);  
$stmt->execute();  
$result = $stmt->get\_result();  
  
$chatUsers = [];  
while ($row = $result->fetch\_assoc()) {  
 $chatUsers[] = $row['chat\_partner'];  
}  
  
echo json\_encode(["success" => true, "users" => $chatUsers]);

————————————————————————————————————————

## Messages\fetch\_messages.php

<?php  
session\_start();  
require\_once '../../../db.php'; // adjust according to your project  
  
header('Content-Type: application/json');  
  
if (!isset($\_SESSION['user\_id'])) {  
 echo json\_encode(["success" => false, "message" => "Not logged in"]);  
 exit;  
}  
  
$currentUser = $\_SESSION['user\_id'];  
$chatPartner = $\_GET['chat\_with'] ?? '';  
  
if (!$chatPartner) {  
 echo json\_encode(["success" => false, "message" => "No user selected"]);  
 exit;  
}  
  
$sql = "SELECT sender\_username, receiver\_username, message\_text, sent\_at   
 FROM messages  
 WHERE   
 (sender\_username = ? AND receiver\_username = ?)  
 OR (sender\_username = ? AND receiver\_username = ?)  
 ORDER BY sent\_at ASC";  
  
$stmt = $conn->prepare($sql);  
$stmt->bind\_param("ssss", $currentUser, $chatPartner, $chatPartner, $currentUser);  
$stmt->execute();  
$result = $stmt->get\_result();  
  
$messages = [];  
while ($row = $result->fetch\_assoc()) {  
 $messages[] = $row;  
}  
  
echo json\_encode(["success" => true, "messages" => $messages]);  
?>

————————————————————————————————————————

## Messages\Messages.css

.top-buttons {  
 display: flex;  
 justify-content: center;  
 margin: 20px 0;  
 flex-wrap: wrap;  
 gap: 10px;  
}  
  
.top-buttons button {  
 background: white;  
 border: none;  
 padding: 10px 22px;  
 margin: 0;  
 border-radius: var(--radius-md);  
 cursor: pointer;  
 box-shadow: var(--shadow-sm);  
 font-weight: 600;  
 color: var(--text-main);  
 transition: var(--transition);  
 min-width: 120px;  
}  
  
.top-buttons .active {  
 background: linear-gradient(to right, #4fc3f7, #81c784);  
 color: white;  
 box-shadow: var(--shadow-md);  
}  
  
/\* Layout \*/  
.main-container {  
 backdrop-filter: blur(8px);  
 display: flex;  
 height: 75vh;  
 margin: 0 5%;  
 border-radius: var(--radius-lg);  
 overflow: hidden;  
 box-shadow: var(--shadow-md);  
 background: var(--card-bg);  
}  
  
.sidebar {  
 width: 30%;  
 padding: 20px;  
 overflow-y: auto;  
 border-top-left-radius: var(--radius-lg);  
 border-bottom-left-radius: var(--radius-lg);  
 box-shadow: var(--shadow-md);  
 backdrop-filter: blur(8px);  
 border-right: none;  
}  
  
.chat-section {  
 width: 70%;  
 display: flex;  
 flex-direction: column;  
 justify-content: space-between;  
 background: linear-gradient(135deg, #e0f7fa 0%, #d0f8ce 100%);  
 backdrop-filter: blur(10px);  
}  
  
.chat-header {  
 background: linear-gradient(to right, #4fc3f7, #81c784);  
 padding: 20px;  
 color: white;  
 font-weight: 600;  
 font-size: 1.1rem;  
 border-top-right-radius: var(--radius-lg);  
 box-shadow: var(--shadow-md);  
}  
  
.chat-body {  
 flex-grow: 1;  
 padding: 20px;  
 overflow-y: auto;  
 display: flex;  
 flex-direction: column;  
}  
  
.message {  
 max-width: 65%;  
 padding: 12px 18px;  
 border-radius: var(--radius-lg);  
 margin-bottom: 12px;  
 font-size: 1rem;  
 box-shadow: var(--shadow-sm);  
 background: linear-gradient(to right, #4fc3f7, #1e90ff);  
 color: white;  
 align-self: flex-start;  
 backdrop-filter: blur(8px);  
 transition: var(--transition);  
 word-wrap: break-word;  
 overflow-wrap: break-word;  
}  
  
.message.right {  
 background: linear-gradient(to right, #34d399, #22c55e);  
 color: white;  
 align-self: flex-end;  
 text-align: right;  
}  
  
.message:hover {  
 transform: translateY(-3px);  
 box-shadow: var(--shadow-md);  
 opacity: 0.95;  
}  
  
.chat-input {  
 display: flex;  
 padding: 14px 20px;  
 border-top: 1px solid #ccc;  
 background: linear-gradient(to right, #e0f7fa, #d0f8ce);  
 backdrop-filter: blur(10px);  
 box-shadow: var(--shadow-sm);  
 border-bottom-left-radius: var(--radius-lg);  
 border-bottom-right-radius: var(--radius-lg);  
 box-shadow: 0 8px 16px rgba(30, 144, 255, 0.1);  
 flex-wrap: wrap;  
 gap: 10px;  
}  
  
.chat-input input {  
 flex: 1;  
 padding: 12px 16px;  
 border-radius: var(--radius-md);  
 border: none;  
 font-size: 1rem;  
 margin-right: 10px;  
 box-shadow: var(--shadow-sm);  
 background: linear-gradient(to right, #ccf1ff, #c5f2d3);  
 color: var(--text-main);  
 outline: none;  
 min-width: 150px;  
}  
  
.chat-input button {  
 background: linear-gradient(to right, var(--primary), var(--secondary));  
 color: white;  
 border: none;  
 padding: 10px 18px;  
 border-radius: var(--radius-md);  
 font-weight: 600;  
 cursor: pointer;  
 transition: var(--transition);  
 box-shadow: var(--shadow-md);  
 white-space: nowrap;  
}  
  
.chat-input input::placeholder {  
 color: #475569;  
 opacity: 1;  
}  
  
.chat-input button:hover {  
 background: linear-gradient(to right, var(--primary), var(--secondary));  
 transform: scale(1.05);  
 opacity: 0.9;  
}  
  
.user-entry {  
 background: white;  
 padding: 14px 18px;  
 margin-bottom: 12px;  
 border-radius: var(--radius-md);  
 cursor: pointer;  
 box-shadow: var(--shadow-sm);  
 transition: var(--transition);  
 position: relative;  
 word-wrap: break-word;  
}  
  
.user-entry:hover {  
 background: linear-gradient(to right, #bae6fd, #bbf7d0);  
 color: var(--text-main);  
 transform: scale(1.02);  
 box-shadow: var(--shadow-md);  
}  
  
.user-entry.active {  
 background: linear-gradient(to right, #4fc3f7, #81c784);  
 color: white;  
 font-weight: 600;  
 box-shadow: 0 4px 12px rgba(30, 144, 255, 0.15);  
}  
  
  
  
.my-freinds-title {  
 font-size: 20px;  
 text-align: center;  
 font-weight: 700;  
 margin-right: 1000px;  
 background: linear-gradient(to right, #000000, #000000);  
 -webkit-background-clip: text;  
 -webkit-text-fill-color: transparent;  
 text-shadow: 2px 2px 8px rgba(0, 0, 0, 0.05);  
 font-family: 'Outfit', sans-serif;  
 letter-spacing: 0.5px;  
}  
  
  
  
  
  
  
/\* Friends Section \*/  
.friends-section {  
 display: flex;  
 flex-wrap: wrap;  
 gap: 20px;  
 margin: 0 5%;  
 padding: 25px;  
 background: linear-gradient(135deg, #e0f7fa, #d0f8ce);  
 border-radius: var(--radius-lg);  
 box-shadow: var(--shadow-md);  
 justify-content: center;  
 backdrop-filter: blur(8px);  
}  
  
.friend-card {  
 width: 160px;  
 text-align: center;  
 background: rgba(255, 255, 255, 0.85);  
 border-radius: var(--radius-md);  
 padding: 16px;  
 box-shadow: var(--shadow-sm);  
 transition: var(--transition);  
 cursor: pointer;  
 backdrop-filter: blur(6px);  
}  
  
.friend-card img {  
 width: 60px;  
 height: 60px;  
 border-radius: 50%;  
 margin-bottom: 10px;  
 object-fit: cover;  
 box-shadow: var(--shadow-sm);  
}  
  
.friend-search {  
 display: flex;  
 justify-content: center;  
 margin: 20px 0;  
 padding: 0 5%;  
}  
  
  
.my-friends-title {  
 font-size: 20px;  
 text-align: center;  
 font-weight: 700;  
 margin-right: 1000px;  
 background: linear-gradient(to right, #000000, #000000);  
 -webkit-background-clip: text;  
 -webkit-text-fill-color: transparent;  
 text-shadow: 2px 2px 8px rgba(0, 0, 0, 0.05);  
 font-family: 'Outfit', sans-serif;  
 letter-spacing: 0.5px;  
}  
  
  
/\* Friends Table Style \*/  
.friends-table {  
 display: flex;  
 flex-direction: column;  
 gap: 16px;  
 padding: 0 5% 40px 5%;  
}  
  
.friend-row {  
 display: flex;  
 align-items: center;  
 justify-content: space-between;  
 background: rgba(255, 255, 255, 0.85);  
 border-radius: var(--radius-md);  
 padding: 14px 20px;  
 box-shadow: var(--shadow-sm);  
 backdrop-filter: blur(6px);  
 transition: var(--transition);  
 flex-wrap: wrap;  
 gap: 10px;  
}  
  
.friend-row:hover {  
 box-shadow: var(--shadow-md);  
 transform: scale(1.01);  
}  
  
.friend-img {  
 border-radius: 50%;  
 width: 50px;  
 height: 50px;  
 object-fit: cover;  
 margin-right: 16px;  
 box-shadow: var(--shadow-sm);  
 flex-shrink: 0;  
}  
  
.friend-name {  
 flex: 1;  
 font-weight: 600;  
 font-size: 1.1rem;  
 color: var(--text-main);  
 min-width: 120px;  
}  
  
.friend-actions {  
 display: flex;  
 flex-wrap: wrap;  
 gap: 10px;  
}  
  
.friend-actions button {  
 padding: 8px 14px;  
 border: none;  
 border-radius: var(--radius-sm);  
 cursor: pointer;  
 font-weight: 500;  
 color: white;  
 background: linear-gradient(to right, #4fc3f7, #1e90ff);  
 box-shadow: var(--shadow-sm);  
 transition: var(--transition);  
 white-space: nowrap;  
}  
  
.friend-actions button:hover {  
 opacity: 0.9;  
}  
  
/\* Responsive Media Queries \*/  
@media (max-width: 1200px) {  
 .main-container {  
 margin: 0 3%;  
 }  
   
 .friends-section,   
 .friends-table {  
 padding: 20px 3%;  
 margin: 0 3%;  
 }  
   
 .friend-card {  
 width: 140px;  
 }  
}  
  
@media (max-width: 992px) {  
 .sidebar {  
 width: 40%;  
 }  
   
 .chat-section {  
 width: 60%;  
 }  
   
 .message {  
 max-width: 75%;  
 }  
   
 .friend-card {  
 width: 130px;  
 padding: 12px;  
 }  
}  
  
@media (max-width: 768px) {  
 .navbar,  
 .fresh-navbar {  
 padding: 14px 20px;  
 justify-content: center;  
 }  
   
 .logo {  
 margin-right: 0;  
 margin-bottom: 10px;  
 width: 100%;  
 text-align: center;  
 }  
   
 .main-container {  
 flex-direction: column;  
 height: auto;  
 margin: 0 20px 20px;  
 }  
   
 .sidebar,   
 .chat-section {  
 width: 100%;  
 border-radius: 0;  
 }  
   
 .sidebar {  
 border-top-left-radius: var(--radius-lg);  
 border-top-right-radius: var(--radius-lg);  
 max-height: 300px;  
 }  
   
 .chat-section {  
 border-bottom-left-radius: var(--radius-lg);  
 border-bottom-right-radius: var(--radius-lg);  
 }  
   
 .chat-header {  
 border-radius: 0;  
 }  
   
 .message {  
 max-width: 85%;  
 }  
   
 .friends-section {  
 margin: 20px;  
 }  
   
 .friend-card {  
 width: calc(50% - 20px);  
 min-width: 120px;  
 }  
   
 .friend-row {  
 padding: 12px 15px;  
 }  
   
 .friend-name {  
 font-size: 1rem;  
 }  
   
 .friend-actions button {  
 padding: 6px 12px;  
 font-size: 0.9rem;  
 }  
}  
  
@media (max-width: 576px) {  
 .nav-links {  
 gap: 8px;  
 width: 100%;  
 justify-content: center;  
 }  
   
 .nav-links li a {  
 padding: 8px 14px;  
 font-size: 0.9rem;  
 }  
   
 .main-container {  
 margin: 0 10px 20px;  
 }  
   
 .chat-input {  
 padding: 10px 15px;  
 }  
   
 .chat-input input {  
 padding: 10px;  
 margin-right: 0;  
 width: 100%;  
 }  
   
 .chat-input button {  
 width: 100%;  
 margin-top: 10px;  
 }  
   
 .message {  
 max-width: 90%;  
 font-size: 0.95rem;  
 }  
   
 .friend-card {  
 width: 100%;  
 }  
   
 .friend-img {  
 width: 40px;  
 height: 40px;  
 margin-right: 10px;  
 }  
   
 .friend-row {  
 justify-content: center;  
 text-align: center;  
 }  
   
 .friend-name {  
 width: 100%;  
 text-align: center;  
 margin: 8px 0;  
 }  
   
 .friend-actions {  
 width: 100%;  
 justify-content: center;  
 }  
}  
  
@media (max-width: 375px) {  
 .navbar,  
 .fresh-navbar {  
 padding: 10px;  
 }  
   
 .top-buttons button {  
 width: 100%;  
 }  
   
 .search-box {  
 padding: 8px 12px;  
 }  
}

————————————————————————————————————————

## Messages\Messages.html

<!DOCTYPE html>  
<html lang="en">  
<head>  
 <meta charset="UTF-8">  
 <title>Messages | Book&Play</title>  
 <link rel="stylesheet" href="Messages.css">  
 <link href="https://fonts.googleapis.com/css2?family=Outfit:wght@400;600&display=swap" rel="stylesheet">  
</head>  
<body>  
  
  
<div>  
 <h1 class="my-friends-title">My Friends :</h1>  
</div>  
  
<div class="main-container" id="chatSection">  
 <!-- Sidebar -->  
 <div class="sidebar" id="sidebar">  
   
 </div>  
  
 <!-- Chat Area -->  
 <div class="chat-section">  
 <div class="chat-header" id="chatHeader">  
 Select a user  
 </div>  
 <div class="chat-body" id="chatBody">  
 <p class="message left" style="display: none;"></p>  
 <p class="message right" style="display: none;"></p>  
 </div>  
 <div class="chat-input">  
 <input type="text" id="messageInput" placeholder="Type a message...">  
 <button id="sendButton">Send</button>  
</div>  
 </div>  
</div>  
  
<script>  
 const sessionUsername = "<?php echo $\_SESSION['user\_id']; ?>";  
</script>  
<script src="Messages.js" defer></script>  
</body>  
</html>

————————————————————————————————————————

## Messages\Messages.js

let currentChatUser = null; // Current user in chat  
  
// When page loads, fetch conversations  
window.addEventListener('DOMContentLoaded', () => {  
 fetch("fetch\_conversations.php")  
 .then(response => response.json())  
 .then(data => {  
 if (data.success) {  
 const sidebar = document.getElementById("sidebar");  
 sidebar.innerHTML = "";  
  
 data.users.forEach(username => {  
 const userDiv = document.createElement("div");  
 userDiv.className = "user-entry";  
 userDiv.textContent = username;  
 userDiv.onclick = () => loadChat(username);  
 sidebar.appendChild(userDiv);  
 });  
 }  
 });  
});  
  
// Load conversation when clicked  
function loadChat(username) {  
 currentChatUser = username;  
 document.getElementById("chatHeader").textContent = username;  
  
 fetch(`fetch\_messages.php?chat\_with=${encodeURIComponent(username)}`)  
 .then(res => res.json())  
 .then(data => {  
 const chatBody = document.getElementById("chatBody");  
 chatBody.innerHTML = "";  
  
 if (data.success) {  
 data.messages.forEach(msg => {  
 const messageElement = document.createElement("p");  
 messageElement.classList.add("message");  
  
 if (msg.sender\_username === sessionUsername) {  
 messageElement.classList.add("right");  
 }  
  
 messageElement.textContent = msg.message\_text;  
 chatBody.appendChild(messageElement);  
 });  
 }  
 });  
}  
  
// Send message when pressing "Send" button or Enter  
document.addEventListener("DOMContentLoaded", () => {  
 const sendButton = document.getElementById("sendButton");  
 const messageInput = document.getElementById("messageInput");  
  
 function sendMessage() {  
 const message = messageInput.value.trim();  
 if (!message || !currentChatUser) return;  
  
 fetch("send\_message.php", {  
 method: "POST",  
 headers: { "Content-Type": "application/x-www-form-urlencoded" },  
 body: new URLSearchParams({  
 receiver: currentChatUser,  
 message: message  
 })  
 })  
 .then(res => res.json())  
 .then(data => {  
 if (data.success) {  
 const chatBody = document.getElementById("chatBody");  
 const msg = document.createElement("p");  
 msg.className = "message right";  
 msg.textContent = message;  
 chatBody.appendChild(msg);  
 messageInput.value = "";  
 } else {  
 alert("❌ Failed to send message.");  
 }  
 });  
 }  
  
 // When pressing Send button  
 sendButton.addEventListener("click", sendMessage);  
  
 // When pressing Enter  
 messageInput.addEventListener("keypress", (e) => {  
 if (e.key === "Enter") {  
 e.preventDefault();  
 sendMessage();  
 }  
 });  
});

————————————————————————————————————————

## Messages\Messages.php

<?php  
session\_start();  
if (!isset($\_SESSION['user\_id'])) {  
 // Unset all session variables  
 session\_unset();  
 // Destroy the session completely  
 session\_destroy();  
 header('Location: ../../auth/Login\_Page/Login.php');  
 exit();  
}  
include '../navbar.php';  
include 'Messages.html';  
exit();  
?>

————————————————————————————————————————

## Messages\send\_message.php

<?php  
session\_start();  
require\_once '../../../db.php';  
  
header('Content-Type: application/json');  
  
$sender = $\_SESSION['user\_id'] ?? '';  
$receiver = $\_POST['receiver'] ?? '';  
$message = trim($\_POST['message'] ?? '');  
  
if (!$sender || !$receiver || !$message) {  
 echo json\_encode(["success" => false, "message" => "Invalid input"]);  
 exit;  
}  
  
$stmt = $conn->prepare("INSERT INTO messages (sender\_username, receiver\_username, message\_text) VALUES (?, ?, ?)");  
$stmt->bind\_param("sss", $sender, $receiver, $message);  
  
if ($stmt->execute()) {  
 echo json\_encode(["success" => true]);  
} else {  
 echo json\_encode(["success" => false, "message" => "Database error"]);  
}  
?>

————————————————————————————————————————

## EditProfile\EditProfile.css

/\* ===================== EDIT PROFILE PAGE SPECIFIC STYLES ===================== \*/  
  
/\* Override body styles for edit profile page (no navbar padding needed) \*/  
body {  
 padding-top: 0; /\* Remove navbar padding for edit profile page \*/  
 display: flex; /\* Enable flexbox \*/  
 align-items: center; /\* Center vertically \*/  
 justify-content: center; /\* Center horizontally \*/  
 min-height: 100vh; /\* Full screen height \*/  
 padding: 20px; /\* Add padding \*/  
 overflow-x: hidden; /\* Prevent horizontal scroll \*/  
}  
  
/\* ===================== EDIT PROFILE-SPECIFIC LAYOUT ===================== \*/  
  
/\* Profile form container - unique to this page \*/  
.profile-form {  
 background-color: var(--card-bg); /\* Card background \*/  
 border: 1px solid var(--card-border); /\* Border color \*/  
 border-radius: var(--radius); /\* Rounded corners \*/  
 box-shadow: var(--shadow); /\* Apply shadow \*/  
 width: 90%; /\* Responsive width \*/  
 max-width: 600px; /\* Max width \*/  
 max-height: 90vh; /\* Max height of form \*/  
 padding: 30px 40px; /\* Internal padding \*/  
 text-align: center; /\* Center text \*/  
 animation: fadeInUp 0.6s ease forwards; /\* Fade in effect \*/  
 overflow-y: auto; /\* Enable vertical scroll \*/  
}  
  
.profile-form h3 {  
 background: linear-gradient(to right, var(--primary), var(--secondary)); /\* Gradient text \*/  
 -webkit-background-clip: text; /\* For Safari \*/  
 background-clip: text; /\* Clip background to text \*/  
 -webkit-text-fill-color: transparent; /\* Transparent fill \*/  
 margin-bottom: 20px; /\* Space below heading \*/  
 font-size: clamp(1.5rem, 5vw, 2rem); /\* Responsive font \*/  
}  
  
/\* ===================== EDIT PROFILE-SPECIFIC COMPONENTS ===================== \*/  
  
/\* Input group styling \*/  
.input-group {  
 margin-bottom: 18px; /\* Space between fields \*/  
 text-align: left; /\* Left align labels \*/  
}  
  
.input-group label {  
 display: block; /\* Make label take full width \*/  
 margin-bottom: 6px; /\* Space below label \*/  
 font-weight: 500; /\* Medium weight \*/  
 color: var(--text-main); /\* Label color \*/  
 font-size: clamp(0.875rem, 2vw, 1rem); /\* Responsive font \*/  
}  
  
/\* Form input overrides for edit profile \*/  
input,  
button,  
select,  
textarea {  
 width: 100%; /\* Full width \*/  
 padding: 12px; /\* Internal padding \*/  
 font-size: clamp(0.875rem, 2vw, 0.9375rem); /\* Responsive font \*/  
 border-radius: 8px; /\* Rounded edges \*/  
 border: 1px solid var(--card-border); /\* Border color \*/  
 background-color: #f8fafc; /\* Light background \*/  
 color: var(--text-main); /\* Text color \*/  
 box-sizing: border-box; /\* Include padding in size \*/  
}  
  
/\* Number input specific styles \*/  
input[type="number"]::-webkit-inner-spin-button,  
input[type="number"]::-webkit-outer-spin-button {  
 -webkit-appearance: none; /\* Remove spin button \*/  
 margin: 0; /\* Remove margin \*/  
}  
  
input[type="number"] {  
 -moz-appearance: textfield; /\* Firefox fix \*/  
}  
  
/\* Placeholder styling \*/  
input::placeholder,  
textarea::placeholder {  
 color: var(--text-muted); /\* Placeholder color \*/  
}  
  
/\* Button styling for edit profile \*/  
.submit-btn,  
#backBtn {  
 background: linear-gradient(to right, var(--primary), var(--secondary)); /\* Gradient button \*/  
 color: white; /\* White text \*/  
 border: none; /\* Remove border \*/  
 cursor: pointer; /\* Pointer cursor \*/  
 font-weight: bold; /\* Bold text \*/  
 margin-top: 10px; /\* Space above \*/  
 transition: 0.3s; /\* Smooth transition \*/  
 padding: clamp(10px, 3vw, 14px); /\* Responsive padding \*/  
 font-size: clamp(0.875rem, 2vw, 1rem); /\* Responsive font \*/  
}  
  
.submit-btn:hover,  
#backBtn:hover {  
 transform: scale(1.03); /\* Slight zoom on hover \*/  
 box-shadow: 0 0 10px rgba(30, 144, 255, 0.3); /\* Glow effect \*/  
}  
  
#backBtn {  
 background: transparent; /\* Transparent button \*/  
 border: 2px solid var(--primary); /\* Blue border \*/  
 color: var(--primary); /\* Blue text \*/  
}  
  
/\* Textarea specific styling \*/  
textarea {  
 resize: none; /\* Disable resize \*/  
 font-family: 'Outfit', sans-serif; /\* Set font \*/  
 min-height: 100px; /\* Minimum height \*/  
}  
  
/\* File input specific styling \*/  
input[type="file"] {  
 background-color: #f0f9ff; /\* Light background \*/  
 color: var(--text-muted); /\* Text color \*/  
 border: 1px solid var(--card-border); /\* Border \*/  
 border-radius: 8px; /\* Rounded \*/  
 padding: 8px; /\* Padding \*/  
 cursor: pointer; /\* Pointer cursor \*/  
}  
  
input[type="file"]::-webkit-file-upload-button {  
 background-color: var(--primary); /\* Blue background \*/  
 color: white; /\* White text \*/  
 border: none; /\* No border \*/  
 padding: 8px 14px; /\* Padding \*/  
 border-radius: 5px; /\* Rounded button \*/  
 cursor: pointer; /\* Pointer cursor \*/  
 margin-right: 8px; /\* Space to the right \*/  
 font-size: clamp(0.75rem, 1.5vw, 0.875rem); /\* Responsive font \*/  
}  
  
input[type="file"]::-webkit-file-upload-button:hover {  
 background-color: var(--secondary); /\* Change to green on hover \*/  
 color: white; /\* Keep text white \*/  
}  
  
/\* Readonly input styling \*/  
input[readonly] {  
 background-color: #f1f5f9; /\* Gray background \*/  
 cursor: not-allowed; /\* Disable cursor \*/  
}  
  
/\* Favorite sports styling \*/  
.favorite-sports {  
 display: flex; /\* Use flexbox \*/  
 flex-wrap: wrap; /\* Allow wrapping \*/  
 gap: 10px; /\* Spacing between items \*/  
 margin-top: 5px; /\* Space on top \*/  
}  
  
.sport-option input[type="checkbox"] {  
 display: none; /\* Hide actual checkbox \*/  
}  
  
.sport-label {  
 display: inline-block; /\* Allow padding \*/  
 padding: 8px 15px; /\* Internal spacing \*/  
 background: transparent; /\* Transparent bg \*/  
 color: var(--primary); /\* Blue text \*/  
 border: 2px solid var(--primary); /\* Border \*/  
 border-radius: 20px; /\* Pill shape \*/  
 cursor: pointer; /\* Pointer cursor \*/  
 transition: all 0.3s ease; /\* Smooth transition \*/  
 font-size: clamp(0.75rem, 1.5vw, 0.875rem); /\* Responsive font \*/  
 white-space: nowrap; /\* Prevent wrapping \*/  
}  
  
.sport-label:hover {  
 transform: scale(1.05); /\* Slight zoom on hover \*/  
}  
  
.sport-option input[type="checkbox"]:checked + .sport-label {  
 background: linear-gradient(to right, var(--primary), var(--secondary)); /\* Highlight \*/  
 color: white; /\* White text \*/  
 border: none; /\* Remove border \*/  
 box-shadow: 0 0 10px rgba(30, 144, 255, 0.3); /\* Glow effect \*/  
}  
  
/\* ===================== EDIT PROFILE-SPECIFIC ANIMATIONS ===================== \*/  
  
@keyframes fadeInUp {  
 from {  
 opacity: 0; /\* Start invisible \*/  
 transform: translateY(20px); /\* Move up \*/  
 }  
 to {  
 opacity: 1; /\* Fully visible \*/  
 transform: translateY(0); /\* Normal position \*/  
 }  
}  
  
/\* ===================== EDIT PROFILE-SPECIFIC RESPONSIVE ===================== \*/  
  
/\* Edit profile-specific responsive adjustments \*/  
@media screen and (max-width: 768px) {  
 .profile-form {  
 padding: 20px 25px; /\* Reduce padding \*/  
 width: 95%; /\* Adjust width \*/  
 }  
  
 .favorite-sports {  
 justify-content: center; /\* Center content \*/  
 }  
}  
  
@media screen and (max-width: 480px) {  
 .profile-form {  
 padding: 15px 20px; /\* Smaller padding \*/  
 width: 100%; /\* Full width \*/  
 }  
  
 input,  
 button,  
 select,  
 textarea {  
 padding: 10px; /\* Smaller padding \*/  
 }  
  
 .sport-label {  
 padding: 6px 12px; /\* Smaller buttons \*/  
 font-size: 0.75rem; /\* Smaller font \*/  
 }  
}  
  
/\* Fix for iOS devices to prevent zooming when focusing on inputs \*/  
@media screen and (max-width: 768px) {  
 input, select, textarea {  
 font-size: 16px !important; /\* Prevent zoom \*/  
 }  
}  
  
/\* Ensure form stays on screen on smaller viewports \*/  
@media screen and (max-height: 700px) {  
 body {  
 align-items: flex-start; /\* Align top \*/  
 padding-top: 20px; /\* Add top padding \*/  
 }  
  
 .profile-form {  
 max-height: calc(100vh - 40px); /\* Adjust height \*/  
 }  
}

————————————————————————————————————————

## EditProfile\EditProfile.html

<!DOCTYPE html>  
<html lang="en">  
<head>  
 <meta charset="UTF-8" />  
 <meta name="viewport" content="width=device-width, initial-scale=1.0" />  
 <title>Create Profile | Book&Play</title>  
  
 <!-- Shared CSS files -->  
 <link rel="stylesheet" href="../../../assets/css/global.css" />  
 <link rel="stylesheet" href="../../../assets/css/components.css" />  
 <link rel="stylesheet" href="../../../assets/css/layouts.css" />  
 <link rel="stylesheet" href="../../../assets/css/responsive.css" />  
 <!-- Edit Profile-specific CSS -->  
 <link rel="stylesheet" href="EditProfile.css" />  
 <!-- Shared JavaScript -->  
 <script src="../../../assets/js/shared.js"></script>  
 <!-- Edit Profile-specific JavaScript -->  
 <script src="EditProfile.js" defer></script>  
</head>  
<body>  
 <!-- ========== START: Profile Form ========== -->  
 <form id="profileForm" class="profile-form" action="EditProfile.php" method="POST" enctype="multipart/form-data">  
 <h3>Edit Your User Profile</h3>  
  
 <!-- Hidden email and password fields (already known from DB) -->  
 <div class="input-group" style="display: none;">  
 <input type="email" id="profileEmail" name="email" />  
 </div>  
 <div class="input-group" style="display: none;">  
 <input type="password" id="profilePassword" name="password" />  
 </div>  
  
 <!-- First name input -->  
 <div class="input-group">  
 <label for="firstName">First Name <span style="color:red">\*</span></label>  
 <input   
 type="text"   
 id="firstName"   
 name="firstName"   
 maxlength="25"   
 pattern="[A-Za-z]+"   
 title="Only letters allowed"   
 required   
 oninput="this.value = this.value.replace(/[^a-zA-Z]/g, '')" />  
 </div>  
  
 <!-- Last name input -->  
 <div class="input-group">  
 <label for="lastName">Last Name <span style="color:red">\*</span></label>  
 <input   
 type="text"   
 id="lastName"   
 name="lastName"   
 maxlength="25"   
 pattern="[A-Za-z]+"   
 title="Only letters allowed"   
 required   
 oninput="this.value = this.value.replace(/[^a-zA-Z]/g, '')" />  
 </div>  
  
 <!-- Username (readonly) -->  
 <div class="input-group">  
 <label for="username">  
 Username <span style="color:red">\*</span>  
 <span style="color: red; font-size: 18px; margin-left: 8px;" title="You can't change your username">🔒</span>  
 </label>  
 <input   
 type="text"   
 id="username"   
 name="username"   
 maxlength="25"   
 required   
 readonly   
 style="background-color: #f1f5f9; cursor: not-allowed;"   
 placeholder="You Can't Change Your Username!" />  
 </div>  
  
 <!-- Age input: number between 10 and 99 -->  
 <div class="input-group">  
 <label for="age">Age <span style="color:red">\*</span></label>  
 <input type="number" id="age" name="age" min="10" max="99" required oninput="this.value = this.value.slice(0,2)" />  
 </div>  
  
 <!-- Gender selection dropdown -->  
 <div class="input-group">  
 <label for="gender">Gender <span style="color:red">\*</span></label>  
 <select id="gender" name="gender" required>  
 <option value="" disabled selected>Select Gender</option>  
 <option value="male">Male</option>  
 <option value="female">Female</option>  
 </select>  
 </div>  
  
 <!-- Phone number input: 10 digits only -->  
 <div class="input-group">  
 <label for="phone">Phone Number</label>  
 <input   
 type="tel"   
 id="phone"   
 name="phone"   
 pattern="^\d{10}$"   
 maxlength="10"   
 inputmode="numeric"   
 oninput="this.value = this.value.replace(/[^0-9]/g, '').slice(0,10);"   
 placeholder="1234567890"   
 title="Enter exactly 10 digits" />  
 </div>  
  
 <!-- Favorite sports (dynamically included via PHP) -->  
 <div class="input-group">  
 <label for="favoriteSport">Favorite Sports</label>  
 <div class="favorite-sports" id="favoriteSportsContainer">  
 <!-- Load sport checkboxes from backend -->  
 </div>  
 </div>  
  
 <!-- Personal description -->  
 <div class="input-group">  
 <label for="description">Personal Description</label>  
 <textarea id="description" name="description" rows="4" placeholder="Tell us about yourself..."></textarea>  
 </div>  
  
 <!-- Profile image upload with preview -->  
 <div class="input-group">  
 <label for="profileImage">Profile Picture</label>  
 <input type="file" id="profileImage" name="user\_image" accept="image/\*" onchange="previewImage()" />  
 <div id="imagePreviewContainer" style="margin-top: 10px;">  
 <img id="profileImagePreview" src="#" alt="Image Preview" style="display: none; width: 120px; height: 120px; border-radius: 50%; object-fit: cover; box-shadow: 0 2px 10px rgba(0,0,0,0.4);" />  
 <br>  
 <button type="button" id="removeImageBtn" style="margin-top: 10px; display: none;" onclick="removeImage()">❌ Remove Image</button>  
 </div>  
 </div>  
  
 <!-- Submit and back buttons -->  
 <button type="submit" class="submit-btn">Save Profile</button>  
 <button onclick="goBack()" class="back-btn">← Back</button>  
 </form>  
 <!-- ========== END: Profile Form ========== -->  
 <!-- ✅ Confirmation Modal -->  
<div id="confirmationModal" style="display:none; position:fixed; top:0; left:0; width:100vw; height:100vh; background:rgba(0,0,0,0.6); z-index:9999; justify-content:center; align-items:center;">  
 <div style="background:white; padding:30px; border-radius:15px; text-align:center; max-width:90%; box-shadow:0 4px 20px rgba(0,0,0,0.2);">  
 <h3 style="margin-bottom: 15px; color:#22c55e;">✅ Your profile has been updated!</h3>  
 <button onclick="closeModal()" style="padding:10px 20px; background:#1e90ff; color:white; border:none; border-radius:8px; cursor:pointer;">OK</button>  
 </div>  
</div>  
</body>  
</html>

————————————————————————————————————————

## EditProfile\EditProfile.js

// Preview profile image on load  
window.previewImage = function () {  
 const input = document.getElementById("profileImage");  
 const preview = document.getElementById("profileImagePreview");  
 const removeBtn = document.getElementById("removeImageBtn");  
  
 const file = input.files[0];  
 if (file) {  
 preview.src = URL.createObjectURL(file);  
 preview.style.display = "block";  
 removeBtn.style.display = "inline-block";  
 }  
};  
  
window.removeImage = function () {  
 const input = document.getElementById("profileImage");  
 const preview = document.getElementById("profileImagePreview");  
 const removeBtn = document.getElementById("removeImageBtn");  
  
 // Clean interface  
 input.value = "";  
 preview.src = "#";  
 preview.style.display = "none";  
 removeBtn.style.display = "none";  
  
 // Send request to delete image from database  
 fetch("EditProfile.php?action=remove\_image")  
 .then((res) => res.json())  
 .then((data) => {  
 if (!data.success) {  
 alert("⚠️ Failed to remove image from database.");  
 }  
 })  
 .catch(() => {  
 alert("⚠️ Error occurred while removing the image.");  
 });  
};  
  
  
// Check that phone contains only numbers (also in oninput)  
document.addEventListener("DOMContentLoaded", function () {  
 const phoneInput = document.getElementById("phone");  
  
 phoneInput.addEventListener("input", function () {  
 this.value = this.value.replace(/[^0-9]/g, "").slice(0, 10);  
 });  
  
 // You can add any additional validation here later  
});  
document.addEventListener("DOMContentLoaded", function () {  
 fetch("EditProfile.php?action=get\_user\_data")  
 .then((res) => res.json())  
 .then((data) => {  
 if (data.success && data.user) {  
 const u = data.user;  
 document.getElementById("firstName").value = u.first\_name || "";  
 document.getElementById("lastName").value = u.last\_name || "";  
 document.getElementById("username").value = u.username || "";  
 document.getElementById("age").value = u.age || "";  
 document.getElementById("gender").value = u.Gender || "";  
 document.getElementById("phone").value = u.phone\_number || "";  
 document.getElementById("description").value = u.description || "";  
  
 if (u.user\_image) {  
 const imgPath = `../../../uploads/users/${u.user\_image}`;  
 const preview = document.getElementById("profileImagePreview");  
 const removeBtn = document.getElementById("removeImageBtn");  
 preview.src = imgPath;  
 preview.style.display = "block";  
 removeBtn.style.display = "inline-block";  
 }  
 }  
 })  
 .catch((err) => console.error("❌ Error loading profile:", err));  
});  
// Show window after saving  
function showModal() {  
 const modal = document.getElementById("confirmationModal");  
 if (modal) modal.style.display = "flex";  
}  
  
// Close window when pressing OK  
window.closeModal = function () {  
 const modal = document.getElementById("confirmationModal");  
 if (modal) {  
 modal.style.display = "none";  
 window.location.href = "EditProfile.php"; // redirect after closing  
 }  
};  
function goBack() {  
 fetch("EditProfile.php?action=prev")  
 .then((res) => res.json())  
 .then((data) => {  
 if (data.url) {  
 window.location.href = data.url;  
 } else {  
 history.back();  
 }  
 });  
}

————————————————————————————————————————

## EditProfile\EditProfile.php

<?php  
session\_start();  
require\_once '../../../db.php';  
  
$username = $\_SESSION['user\_id'] ?? '';  
  
if (isset($\_GET['action']) && $\_GET['action'] === 'remove\_image') {  
 $username = $\_SESSION['user\_id'] ?? '';  
 if (!$username) {  
 echo json\_encode(['success' => false, 'message' => 'Not logged in']);  
 exit;  
 }  
  
 // Delete image name from database  
 $stmt = $conn->prepare("UPDATE users SET user\_image = NULL WHERE username = ?");  
 $stmt->bind\_param("s", $username);  
 if ($stmt->execute()) {  
 echo json\_encode(['success' => true]);  
 } else {  
 echo json\_encode(['success' => false, 'message' => 'Failed to remove image']);  
 }  
 exit;  
}  
  
// ✅ API to get user data (for JavaScript)  
if (isset($\_GET['action']) && $\_GET['action'] === 'get\_user\_data') {  
 header('Content-Type: application/json');  
  
 if (!$username) {  
 echo json\_encode(['error' => 'Not logged in']);  
 exit;  
 }  
  
 $stmt = $conn->prepare("SELECT \* FROM users WHERE username = ?");  
 $stmt->bind\_param("s", $username);  
 $stmt->execute();  
 $result = $stmt->get\_result();  
 if ($user = $result->fetch\_assoc()) {  
 echo json\_encode(['success' => true, 'user' => $user]);  
 } else {  
 echo json\_encode(['error' => 'User not found']);  
 }  
 exit;  
}  
  
// ✅ Check login when opening page directly  
if (!$username) {  
 header("Location: ../Login\_Page/Login.php");  
 exit;  
}  
  
// ✅ Handle saving changes (POST)  
if ($\_SERVER['REQUEST\_METHOD'] === 'POST') {  
 $firstName = $\_POST['firstName'];  
 $lastName = $\_POST['lastName'];  
 $age = $\_POST['age'];  
 $gender = $\_POST['gender'];  
 $phone = $\_POST['phone'];  
 $description = $\_POST['description'];  
 $sports = isset($\_POST['favoriteSport']) ? $\_POST['favoriteSport'] : [];  
  
 // ✅ Handle profile image upload  
 $user\_image = null;  
 if (isset($\_FILES['user\_image']) && $\_FILES['user\_image']['error'] === UPLOAD\_ERR\_OK) {  
 $image\_tmp = $\_FILES['user\_image']['tmp\_name'];  
 $original\_name = basename($\_FILES['user\_image']['name']);  
 $filename = time() . "\_" . $original\_name;  
 $upload\_folder = "../../../uploads/users/";  
  
 if (!is\_dir($upload\_folder)) {  
 mkdir($upload\_folder, 0775, true);  
 }  
  
 $target\_path = $upload\_folder . $filename;  
  
 if (move\_uploaded\_file($image\_tmp, $target\_path)) {  
 $user\_image = $filename;  
 }  
 }  
  
 // ✅ Update users table  
 if ($user\_image) {  
 $stmt = $conn->prepare("UPDATE users SET first\_name=?, last\_name=?, age=?, Gender=?, phone\_number=?, description=?, user\_image=? WHERE username=?");  
 $stmt->bind\_param("ssisssss", $firstName, $lastName, $age, $gender, $phone, $description, $user\_image, $username);  
 } else {  
 $stmt = $conn->prepare("UPDATE users SET first\_name=?, last\_name=?, age=?, Gender=?, phone\_number=?, description=? WHERE username=?");  
 $stmt->bind\_param("ssissss", $firstName, $lastName, $age, $gender, $phone, $description, $username);  
 }  
  
 $stmt->execute();  
 $stmt->close();  
  
 // ✅ Delete previous sports  
 $deleteStmt = $conn->prepare("DELETE FROM user\_favorite\_sports WHERE username = ?");  
 if ($deleteStmt) {  
 $deleteStmt->bind\_param("s", $username);  
 $deleteStmt->execute();  
 $deleteStmt->close();  
 }  
  
 // ✅ Insert new favorite sports  
 $insertStmt = $conn->prepare("INSERT INTO user\_favorite\_sports (username, sport\_id) VALUES (?, ?)");  
 foreach ($sports as $sportName) {  
 $sportQuery = $conn->prepare("SELECT sport\_id FROM sports WHERE sport\_name = ?");  
 $sportQuery->bind\_param("s", $sportName);  
 $sportQuery->execute();  
 $sportQuery->bind\_result($sportId);  
 if ($sportQuery->fetch()) {  
 $insertStmt->bind\_param("si", $username, $sportId);  
 $insertStmt->execute();  
 }  
 $sportQuery->close();  
 }  
 $insertStmt->close();  
  
 echo "<script>window.addEventListener('DOMContentLoaded', () => showModal());</script>";  
include 'EditProfile.html';  
exit;  
 $conn->close();  
 exit;  
}  
if (isset($\_GET['action']) && $\_GET['action'] === 'prev') {  
 header('Content-Type: application/json');  
 echo json\_encode(['url' => $\_SESSION['previous\_page'] ?? '']);  
 exit;  
}  
  
  
// ✅ Display HTML page  
require\_once '../../../components/sports-scroll.php';  
include 'EditProfile.html';  
exit;  
?>

————————————————————————————————————————

## Login\_Page\Login.css

/\*\*  
 \* Login Page CSS Styles  
 \*   
 \* This file contains login-specific styles that override global styles  
 \* and provide custom styling for the login form and page layout.  
 \*   
 \* @author BOOK-PLAY Development Team  
 \* @version 1.0  
 \* @since 2025  
 \*/  
  
/\* ===== LOGIN PAGE LAYOUT OVERRIDES ===== \*/  
  
/\* Override body styles for login page (no navbar padding needed) \*/  
body {  
 padding-top: 0; /\* Remove navbar padding for login page \*/  
 display: flex; /\* Flex layout for centering \*/  
 justify-content: center; /\* Center horizontally \*/  
 align-items: center; /\* Center vertically \*/  
 overflow-x: hidden; /\* Hide horizontal scroll \*/  
 position: relative; /\* For absolute positioned children \*/  
 padding: 10px; /\* Outer padding for mobile \*/  
}  
  
/\* ===== LOGIN FORM STYLES ===== \*/  
  
/\* Error message styling for login form \*/  
.error-message {  
 color: red; /\* Red color for errors \*/  
 font-weight: bold; /\* Bold text for emphasis \*/  
 margin-top: 10px; /\* Top margin for spacing \*/  
 text-align: center; /\* Center align text \*/  
}  
  
/\* Form validation visual feedback \*/  
form input:invalid {  
 border-color: #ef4444; /\* Red border for invalid input \*/  
}  
  
form input:valid {  
 border-color: #10b981; /\* Green border for valid input \*/  
}  
  
/\* Loading state for submit button \*/  
form button:disabled {  
 opacity: 0.7; /\* Reduce opacity when disabled \*/  
 cursor: not-allowed; /\* Show disabled cursor \*/  
 transform: none; /\* Remove hover effects \*/  
}  
  
/\* Focus states for form inputs \*/  
form input:focus {  
 border-color: #1e90ff; /\* Blue border on focus \*/  
 outline: none; /\* Remove default outline \*/  
 box-shadow: 0 0 5px rgba(30, 144, 255, 0.2); /\* Blue glow effect \*/  
}  
  
/\* Hover effects for submit button \*/  
form button:hover:not(:disabled) {  
 transform: scale(1.03); /\* Slight scale up on hover \*/  
 box-shadow: 0 0 15px rgba(30, 144, 255, 0.3); /\* Enhanced glow \*/  
}  
  
/\* Active state for submit button \*/  
form button:active:not(:disabled) {  
 transform: scale(0.98); /\* Slight scale down when clicked \*/  
}

————————————————————————————————————————

## Login\_Page\Login.html

<!DOCTYPE html>  
<html lang="en">  
 <head>  
 <meta charset="UTF-8" />  
 <meta name="viewport" content="width=device-width, initial-scale=1.0">  
 <title>Login - Book&Play</title>  
   
 <!-- ===== STYLESHEETS ===== -->  
 <!-- Shared CSS files for consistent styling -->  
 <link rel="stylesheet" href="../../../assets/css/global.css" />  
 <link rel="stylesheet" href="../../../assets/css/components.css" />  
 <link rel="stylesheet" href="../../../assets/css/layouts.css" />  
 <link rel="stylesheet" href="../../../assets/css/responsive.css" />  
 <!-- Login-specific CSS for custom styling -->  
 <link rel="stylesheet" href="Login.css" />  
   
 <!-- ===== JAVASCRIPT FILES ===== -->  
 <!-- Shared JavaScript for common functionality -->  
 <script src="../../../assets/js/shared.js"></script>  
 <!-- Login-specific JavaScript for form handling -->  
 <script src="Login.js"></script>  
 </head>  
 <body>  
 <!-- ===== BACKGROUND DECORATIVE ICONS ===== -->  
 <div class="bg-icons">  
 <!-- Decorative sports-themed background icons for visual appeal -->  
 <img src="../../../Images/TennisPlayer\_icon.png" class="icon1" alt="Tennis Player Icon" />  
 <img src="../../../Images/GroupPlayers\_icon.png" class="icon2" alt="Group Players Icon" />  
 <img src="../../../Images/FootBallPlayer\_icon.png" class="icon3" alt="Football Player Icon" />  
 <img src="../../../Images/Facility\_icon.png" class="icon4" alt="Sports Facility Icon" />  
 <img src="../../../Images/SearchPlayer\_icon.png" class="icon5" alt="Search Player Icon" />  
 <img src="../../../Images/Booking\_icon.png" class="icon6" alt="Booking Icon" />  
 </div>  
  
 <!-- ===== MAIN LOGIN FORM CONTAINER ===== -->  
 <div class="container">  
 <!-- Glass-style container for login form -->  
 <h2 class="logo">BOOK&PLAY</h2>  
  
 <!-- Login form with username and password fields -->  
 <form method="POST" id="loginForm">  
 <input type="text" name="username" placeholder="Username" required />  
 <input type="password" name="password" placeholder="Password" required />  
 <button type="submit">Login</button>  
   
 <!-- Error message display area -->  
 <p id="errorMessage" class="error-message" style="color: red; font-weight: bold"></p>  
 </form>  
  
 <!-- Registration link for new users -->  
 <div class="slogan">  
 Not registered?  
 <a href="../Register\_Page/Register.php">Create an account</a>  
 </div>  
  
 <!-- Password reset link for existing users -->  
 <div class="slogan">  
 Forgot Your Password?  
 <a href="../Reset\_Password/Reset\_Password.php">Reset Password</a>  
 </div>  
 </div>  
 </body>  
</html>

————————————————————————————————————————

## Login\_Page\Login.js

/\*\*  
 \* Login Page JavaScript  
 \*   
 \* This file handles the client-side login form functionality including:  
 \* - Form submission handling  
 \* - AJAX authentication requests  
 \* - Error message display  
 \* - Success redirects  
 \*   
 \* @author BOOK-PLAY Development Team  
 \* @version 1.0  
 \* @since 2025  
 \*/  
  
/\*\*  
 \* Initialize login form functionality when DOM is loaded  
 \* Sets up form submission handler and AJAX authentication  
 \*/  
document.addEventListener('DOMContentLoaded', function () {  
 // Get login form element  
 const form = document.getElementById('loginForm');  
  
 /\*\*  
 \* Handle form submission with AJAX  
 \* Prevents default form submission and sends data via fetch API  
 \*/  
 form.addEventListener('submit', function (e) {  
 e.preventDefault(); // Prevent default form submission  
   
 // Get form input values  
 const identifier = form.querySelector('input[name="username"]').value;  
 const password = form.querySelector('input[name="password"]').value;  
 const errorMessage = document.getElementById('errorMessage');  
   
 // Show loading message  
 errorMessage.textContent = '⏳ Verifying...';  
  
 // Prepare form data for submission  
 const formData = new FormData();  
 formData.append('username', identifier);  
 formData.append('password', password);  
  
 // Send authentication request to server  
 fetch('login\_process.php', {  
 method: 'POST',  
 body: formData,  
 })  
 .then((res) => res.json()) // Parse JSON response  
 .then((data) => {  
 if (data.success) {  
 // Redirect to appropriate dashboard on success  
 window.location.href = data.redirect;  
 } else {  
 // Display error message on failure  
 errorMessage.textContent = data.message;  
 }  
 })  
 .catch((err) => {  
 // Handle network or server errors  
 errorMessage.textContent = '❌ Login error occurred, please try again.';  
 console.error(err);  
 });  
 });  
});

————————————————————————————————————————

## Login\_Page\Login.php

<?php  
/\*\*  
 \* Login Page Controller  
 \*   
 \* This file serves as the main entry point for the login page.  
 \* It starts a session and includes the login HTML template.  
 \*   
 \* @author BOOK-PLAY Development Team  
 \* @version 1.0  
 \* @since 2025  
 \*/  
  
// Start session for user authentication  
session\_start();  
  
// Include the login page HTML template  
include 'Login.html';  
exit();  
?>

————————————————————————————————————————

## Login\_Page\login\_process.php

<?php  
/\*\*  
 \* Login Process Handler  
 \*   
 \* This file handles user authentication by processing login form submissions.  
 \* It supports two user types: regular users and facility owners.  
 \*   
 \* Authentication Flow:  
 \* 1. Validates form data  
 \* 2. Checks owner table first (by email)  
 \* 3. Checks users table (by username)  
 \* 4. Verifies password and sets session  
 \* 5. Redirects to appropriate dashboard  
 \*   
 \* @author BOOK-PLAY Development Team  
 \* @version 1.0  
 \* @since 2025  
 \*/  
  
// Start session and include database connection  
session\_start();  
require\_once '../../../db.php';   
  
// Initialize response array  
$response = ['success' => false, 'message' => ''];  
  
// Process POST request for login  
if ($\_SERVER['REQUEST\_METHOD'] === 'POST') {  
 $identifier = $\_POST['username']; // Can be username or email  
 $password = $\_POST['password'];  
  
 // Validate input fields  
 if (empty($identifier) || empty($password)) {  
 $response['message'] = 'Please fill in all required fields';  
 } else {  
 $user = null;  
 $role = null;  
  
 // 🔎 First check in the owner table by owner\_email  
 $stmt = $conn->prepare("SELECT \* FROM owner WHERE owner\_email = ? LIMIT 1");  
 $stmt->bind\_param("s", $identifier);  
 $stmt->execute();  
 $result = $stmt->get\_result();  
 if ($result->num\_rows === 1) {  
 $user = $result->fetch\_assoc();  
 $role = 'owner';  
 }  
 $stmt->close();  
  
 // 🔎 If not found, check in the users table by username only  
 if (!$user) {  
 $stmt = $conn->prepare("SELECT \* FROM users WHERE username = ? LIMIT 1");  
 $stmt->bind\_param("s", $identifier);  
 $stmt->execute();  
 $result = $stmt->get\_result();  
 if ($result->num\_rows === 1) {  
 $user = $result->fetch\_assoc();  
 $role = 'user';  
 }  
 $stmt->close();  
 }  
  
 // Process authentication if user found  
 if ($user) {  
 if (password\_verify($password, $user['password'])) {  
 // Set session variables based on user role  
 $\_SESSION['role'] = $role;  
  
 if ($role === 'owner') {  
 // Owner authentication - redirect to owner dashboard  
 $\_SESSION['user\_id'] = $user['owner\_email'];  
 $\_SESSION['user\_name'] = 'Administrator';  
 $response['redirect'] = '../../Owner/Owner.php';  
 } else {  
 // Regular user authentication - redirect to user selection  
 $\_SESSION['user\_id'] = $user['username'];  
 $\_SESSION['user\_name'] = $user['first\_name'] . ' ' . $user['last\_name'];  
 $response['redirect'] = '../User\_Selection\_Page/user-selection.php';  
 }  
  
 $response['success'] = true;  
 $response['message'] = '✅ Login successful!';  
 } else {  
 $response['message'] = '❌ Incorrect password';  
 }  
 } else {  
 $response['message'] = '❌ User not found';  
 }  
 }  
}  
  
// Return JSON response  
header('Content-Type: application/json');  
echo json\_encode($response);  
$conn->close();  
?>

————————————————————————————————————————

## Register\_Page\Register.css

/\* ===================== REGISTER PAGE SPECIFIC STYLES ===================== \*/  
  
/\* Override body styles for register page (no navbar padding needed) \*/  
body {  
 padding-top: 0; /\* Remove navbar padding for register page \*/  
 display: flex; /\* Flex layout \*/  
 align-items: center; /\* Center items vertically \*/  
 justify-content: center; /\* Center items horizontally \*/  
 min-height: 100vh; /\* Full viewport height \*/  
 overflow-x: hidden; /\* Hide horizontal overflow \*/  
 padding: 15px; /\* Inner padding \*/  
}  
  
/\* ===================== REGISTER-SPECIFIC LAYOUT ===================== \*/  
  
/\* Register container - unique to this page \*/  
.register-container {  
 background-color: var(--card-bg);  
 backdrop-filter: blur(10px);  
 display: flex;  
 border-radius: 20px;  
 box-shadow: var(--shadow);  
 overflow: hidden;  
 width: 980px;  
 max-width: 100%;  
 transition: all 0.3s ease;  
 flex-direction: row;  
}  
  
.left-section,  
.right-section {  
 flex: 1;  
 padding: 40px;  
 display: flex;  
 flex-direction: column;  
 justify-content: center;  
}  
  
.right-section {  
 background: rgba(255, 255, 255, 0.15);  
 backdrop-filter: blur(10px);  
 text-align: center;  
 pointer-events: none;  
 opacity: 0.6;  
 transition: opacity 0.3s ease;  
 border-left: 1px solid rgba(255, 255, 255, 0.2);  
 box-shadow: inset 0 0 40px rgba(30, 144, 255, 0.1);  
}  
  
.right-section.active {  
 pointer-events: auto;  
 opacity: 1;  
}  
  
.right-section img {  
 width: 160px;  
 margin: 25px auto;  
 display: block;  
 filter: drop-shadow(0 0 10px rgba(30, 144, 255, 0.3));  
}  
  
/\* ===================== REGISTER-SPECIFIC FORM STYLES ===================== \*/  
  
/\* Input groups - specific to register forms \*/  
.input-group {  
 margin-bottom: 18px;  
 text-align: left;  
}  
  
.input-group label {  
 display: block;  
 margin-bottom: 6px;  
 font-weight: 500;  
 color: var(--text-main);  
}  
  
/\* Register-specific button styles \*/  
.submit-btn,  
.verify-btn {  
 background: linear-gradient(to right, var(--primary), var(--secondary));  
 color: white;  
 border: none;  
 cursor: pointer;  
 font-weight: bold;  
 transition: 0.3s;  
 margin-top: 10px;  
}  
  
.submit-btn:hover,  
.verify-btn:hover {  
 transform: scale(1.03);  
 box-shadow: 0 0 10px rgba(30, 144, 255, 0.3);  
}  
  
.back-btn {  
 background: transparent;  
 border: 2px solid var(--primary);  
 color: var(--primary);  
}  
  
.make-profile-btn {  
 font-family: 'Outfit', sans-serif;  
 text-decoration: none;  
 border: 2px solid var(--primary);  
 padding: 12px 25px;  
 border-radius: 25px;  
 color: var(--primary);  
 margin-top: 20px;  
 transition: 0.3s;  
 background: transparent;  
}  
  
.make-profile-btn:hover {  
 background-color: var(--primary);  
 color: white;  
}  
  
/\* ===================== REGISTER-SPECIFIC COMPONENTS ===================== \*/  
  
/\* File upload styling \*/  
input[type="file"] {  
 background-color: #f0f9ff;  
 color: var(--text-muted);  
 cursor: pointer;  
}  
  
input[type="file"]::-webkit-file-upload-button {  
 background-color: var(--primary);  
 color: white;  
 border: none;  
 padding: 8px 14px;  
 border-radius: 5px;  
 cursor: pointer;  
}  
  
input[type="file"]::-webkit-file-upload-button:hover {  
 background-color: var(--secondary);  
}  
  
/\* Sports selection styling \*/  
.favorite-sports {  
 display: flex;  
 flex-wrap: wrap;  
 gap: 10px;  
 margin-top: 5px;  
}  
  
.sport-option input[type="checkbox"] {  
 display: none;  
}  
  
.sport-label {  
 display: inline-block;  
 padding: 8px 15px;  
 background: transparent;  
 color: var(--primary);  
 border: 2px solid var(--primary);  
 border-radius: 20px;  
 cursor: pointer;  
 transition: all 0.3s ease;  
}  
  
.sport-label:hover {  
 transform: scale(1.05);  
}  
  
.sport-option input[type="checkbox"]:checked + .sport-label {  
 background: linear-gradient(to right, var(--primary), var(--secondary));  
 color: white;  
 border: none;  
 box-shadow: 0 0 10px rgba(30, 144, 255, 0.3);  
}  
  
/\* ===================== REGISTER-SPECIFIC MESSAGES ===================== \*/  
  
.verification-message {  
 margin-top: 10px;  
 font-weight: bold;  
}  
  
.success {  
 color: var(--secondary);  
}  
  
.error {  
 color: #ef4444;  
}  
  
/\* ===================== REGISTER-SPECIFIC ANIMATIONS ===================== \*/  
  
@keyframes pulse {  
 0% {  
 transform: scale(1);  
 box-shadow: 0 0 0 0 rgba(30, 144, 255, 0.7);  
 }  
 70% {  
 transform: scale(1.05);  
 box-shadow: 0 0 0 10px rgba(30, 144, 255, 0);  
 }  
 100% {  
 transform: scale(1);  
 box-shadow: 0 0 0 0 rgba(30, 144, 255, 0);  
 }  
}  
  
.pulsing {  
 animation: pulse 1.5s infinite;  
}  
  
/\* ===================== PROFILE FORM CONTAINER ===================== \*/  
  
#profileFormContainer {  
 display: none;  
 position: fixed;  
 top: 0;  
 left: 0;  
 width: 100vw;  
 height: 100vh;  
 justify-content: center;  
 align-items: center;  
 animation: fadeInUp 0.6s ease;  
 background: linear-gradient(135deg, var(--bg-light), var(--bg-light2));  
 z-index: 9999;  
}  
  
.profile-form {  
 background-color: var(--card-bg);  
 border: 1px solid var(--card-border);  
 border-radius: var(--radius);  
 box-shadow: var(--shadow);  
 width: 90%;  
 max-width: 600px;  
 max-height: 90vh;  
 padding: 30px 40px;  
 text-align: center;  
 overflow-y: auto;  
}  
  
.error-message {  
 color: red;  
 font-size: 0.85em;  
 margin-top: 5px;  
 display: block;  
}  
  
/\* ===================== REGISTER-SPECIFIC RESPONSIVE ===================== \*/  
  
/\* Register-specific responsive adjustments \*/  
@media screen and (max-width: 768px) {  
 .register-container {  
 flex-direction: column;  
 height: auto;  
 margin: 20px 0;  
 }  
  
 .left-section,  
 .right-section {  
 padding: 25px;  
 }  
  
 .profile-form {  
 padding: 20px;  
 width: 95%;  
 }  
}  
  
@media screen and (max-width: 480px) {  
 .left-section,  
 .right-section {  
 padding: 20px 15px;  
 }  
  
 .profile-form {  
 padding: 15px;  
 }  
}

————————————————————————————————————————

## Register\_Page\Register.html

<!-- ===================== REGISTER PAGE HTML ===================== -->  
<!DOCTYPE html>  
<html lang="en">  
<head>  
 <meta charset="UTF-8" />  
 <meta name="viewport" content="width=device-width, initial-scale=1.0"/>  
 <title>Register | Book&Play</title>  
 <!-- Shared CSS files -->  
 <link rel="stylesheet" href="../../../assets/css/global.css" />  
 <link rel="stylesheet" href="../../../assets/css/components.css" />  
 <link rel="stylesheet" href="../../../assets/css/layouts.css" />  
 <link rel="stylesheet" href="../../../assets/css/responsive.css" />  
 <!-- Register-specific CSS -->  
 <link rel="stylesheet" href="Register.css" />  
 <!-- Shared JavaScript -->  
 <script src="../../../assets/js/shared.js"></script>  
 <!-- Register-specific JavaScript -->  
 <script defer src="Register.js"></script>  
</head>  
<body>  
 <!-- Main container holding both sides -->  
 <div class="register-container">  
  
 <!-- ========== LEFT SECTION ========== -->  
 <div class="left-section" id="leftSection">  
 <h2>Create Your Account</h2>  
 <form id="registerForm">  
 <div class="input-group">  
 <label for="email">Email Address <span style="color:red">\*</span></label>  
 <input type="email" id="email" name="email" required />  
 </div>  
 <div class="input-group">  
 <label for="password">Password <span style="color:red">\*</span></label>  
 <input type="password" id="password" name="password" required />  
 </div>  
 <div class="input-group">  
 <label for="confirmPassword">Confirm Password <span style="color:red">\*</span></label>  
 <input type="password" id="confirmPassword" name="confirmPassword" required />  
 </div>  
 <button type="submit" class="submit-btn">Send Me Code</button>  
 <button type="button" id="backBtn" class="back-btn" onclick="window.location.href='../Login\_Page/login.php'">Back</button>  
 </form>  
  
 <!-- Code input after email confirmation -->  
 <div class="code-verification" id="codeVerification" style="display: none;">  
 <h3>Enter the Code You Received</h3>  
 <input type="text" id="codeInput" placeholder="Enter verification code" />  
 <button class="verify-btn" onclick="verifyCode()">Verify</button>  
 <button type="button" class="back-btn" onclick="window.goBack()">Back</button>  
 <div id="verificationMessage" class="verification-message"></div>  
 </div>  
 </div>  
  
 <!-- ========== RIGHT SECTION ========== -->  
 <div class="right-section" id="rightSection">  
 <h2>Start Your Journey</h2>  
 <p>Customize your profile and get discovered by other players and venue owners!</p>  
 <img src="../../../Images/GroupPlayers\_icon.png">  
 <a href="#" class="make-profile-btn">Make Your Profile</a>  
 </div>  
 </div>  
  
 <!-- ========== PROFILE FORM ========== -->  
 <div id="profileFormContainer" style="display: none;">  
 <form id="profileForm" class="profile-form" method="POST" enctype="multipart/form-data">  
 <h3>Create Your User Profile</h3>  
  
 <!-- Hidden Email Field -->  
 <div class="input-group" style="display: none;">  
 <input type="email" id="profileEmail" name="email" />  
 </div>  
 <!-- Hidden Password Field -->  
 <div class="input-group" style="display: none;">  
 <input type="password" id="profilePassword" name="password" />  
 </div>  
  
 <!-- First Name -->  
 <div class="input-group">  
 <label for="firstName">First Name <span style="color:red">\*</span></label>  
 <input type="text" id="firstName" name="firstName" maxlength="25" pattern="[A-Za-z]+" title="Only letters allowed" required oninput="this.value = this.value.replace(/[^a-zA-Z]/g, '')" />  
 </div>  
  
 <!-- Last Name -->  
 <div class="input-group">  
 <label for="lastName">Last Name <span style="color:red">\*</span></label>  
 <input type="text" id="lastName" name="lastName" maxlength="25" pattern="[A-Za-z]+" title="Only letters allowed" required oninput="this.value = this.value.replace(/[^a-zA-Z]/g, '')" />  
 </div>  
  
 <!-- Username -->  
 <div class="input-group">  
 <label for="username">Username <span style="color:red">\*</span></label>  
 <input type="text" id="username" name="username" maxlength="25" required />  
 <div class="error-message" id="usernameError"></div>  
 </div>  
  
 <!-- Age -->  
 <div class="input-group">  
 <label for="age">Age <span style="color:red">\*</span></label>  
 <input type="number" id="age" name="age" min="10" max="99" required oninput="this.value = this.value.slice(0,2)" />  
 </div>  
  
 <!-- Gender -->  
 <div class="input-group">  
 <label for="gender">Gender <span style="color:red">\*</span></label>  
 <select id="gender" name="gender" required>  
 <option value="" disabled selected>Select Gender</option>  
 <option value="male">Male</option>  
 <option value="female">Female</option>  
 </select>  
 </div>  
  
 <!-- Phone -->  
 <div class="input-group">  
 <label for="phone">Phone Number</label>  
 <input type="tel" id="phone" name="phone" pattern="^\d{10}$" maxlength="10" placeholder="1234567890" title="Enter exactly 10 digits" />  
 </div>  
  
 <!-- Favorite Sports (to be loaded dynamically) -->  
 <div class="input-group">  
 <label for="favoriteSport">Favorite Sports</label>  
 <div class="favorite-sports"></div>  
 </div>  
  
 <!-- Description -->  
 <div class="input-group">  
 <label for="description">Personal Description</label>  
 <textarea id="description" name="description" rows="4" placeholder="Tell us about yourself..."></textarea>  
 </div>  
  
 <!-- Profile Image -->  
 <div class="input-group">  
 <label for="profileImage">Profile Picture</label>  
 <input type="file" id="profileImage" name="user\_image" accept="image/\*" onchange="previewImage()" />  
 <div id="imagePreviewContainer" style="margin-top: 10px;">  
 <img id="profileImagePreview" src="#" alt="Image Preview" style="display: none; width: 120px; height: 120px; border-radius: 50%; object-fit: cover; box-shadow: 0 2px 10px rgba(0,0,0,0.4);" />  
 <br>  
 <button type="button" id="removeImageBtn" style="margin-top: 10px; display: none;" onclick="removeImage()">❌ Remove Image</button>  
 </div>  
 </div>  
  
 <button type="submit" class="submit-btn save-profile-btn">Save Profile</button>  
 <button type="button" class="back-btn" onclick="window.goBack()">Back</button>  
 </form>  
 </div>  
  
 <!-- RTL button examples (Hebrew translated) -->  
 <div class="rtl-buttons-container" style="display: none;" dir="rtl">  
 <button class="rtl-option-btn rtl-primary-btn">Choose File</button>   
 <button class="rtl-option-btn rtl-secondary-btn">No File Selected</button>  
 </div>  
</body>  
</html>

————————————————————————————————————————

## Register\_Page\Register.js

// ✅ DOM elements used throughout the multi-step registration form  
var registerForm = document.getElementById("registerForm"); // Get the registration form element  
var codeSection = document.getElementById("codeVerification"); // Get the code verification section  
var profileForm = document.getElementById("profileForm"); // Get the profile creation form  
var message = document.getElementById("verificationMessage"); // Get the message element for status updates  
var makeProfileBtn = document.querySelector(".make-profile-btn"); // Select the "Make Profile" button  
var rightSection = document.getElementById("rightSection"); // Section for displaying the profile form  
var favoriteSportsContainer = document.querySelector(".favorite-sports"); // Container where sports will be injected  
var registerContainer = document.getElementsByClassName("register-container")[0]; // Get the registration container  
var profileFormContainer = document.getElementById("profileFormContainer"); // Get the container for profile form  
  
// ✅ Load sports dynamically from the server (only if the container exists)  
if (favoriteSportsContainer) { // Check if sports container exists  
 fetch("verify.php?get\_sports=1") // Send GET request to fetch sports  
 .then(response => response.text()) // Convert response to text (HTML expected)  
 .then(data => {  
 favoriteSportsContainer.innerHTML = data; // Insert the returned HTML into the container  
 })  
 .catch(error => { // Handle errors  
 console.error("Error fetching sports:", error); // Log error in console  
 favoriteSportsContainer.innerHTML = "<p>Error loading sports options</p>"; // Display fallback message  
 });  
}  
  
// ✅ Hide the profile section initially  
if (rightSection) {  
 rightSection.classList.remove("active"); // Remove "active" class to hide the section initially  
}  
  
// ✅ Switch to profile creation form when user clicks "Make Profile"  
if (makeProfileBtn) { // Check if the button exists  
 makeProfileBtn.onclick = function(e) { // Add click event  
 e.preventDefault(); // Prevent default form submission  
 if (profileFormContainer) profileFormContainer.style.display = "flex"; // Show profile form  
 if (registerContainer) registerContainer.style.display = "none"; // Hide registration form  
 };  
}  
  
// ✅ Go back to the previous step or login page depending on where the user is  
window.goBack = function(e) {  
 if (e) e.preventDefault();  
   
 // Determine which stage we're at and go back accordingly  
 if (profileFormContainer && profileFormContainer.style.display === "flex") {  
 profileFormContainer.style.display = "none"; // Hide profile form  
 registerContainer.style.display = "flex"; // Show registration form  
 } else if (codeSection && codeSection.style.display === "block") {  
 codeSection.style.display = "none"; // Hide code verification  
 registerForm.style.display = "block"; // Show registration form  
 } else {  
 window.location.href = '../Login\_Page/Login.php'; // Redirect to login page  
 }  
};  
  
// ✅ Attach "go back" logic to all back buttons  
document.addEventListener("DOMContentLoaded", function() { // Run after page loads  
 var backButtons = document.querySelectorAll('.back-btn'); // Get all back buttons  
 backButtons.forEach(function(btn) { // Loop through buttons  
 if (!btn.id || btn.id !== "backBtn") { // Exclude the main back button if needed  
 btn.onclick = window.goBack; // Assign goBack function  
 }  
 });  
});  
  
// ✅ Handle registration form submission (step 1)  
if (registerForm) { // Check if form exists  
 registerForm.onsubmit = function(e) {  
 e.preventDefault(); // Prevent page reload  
  
 var email = document.getElementById("email").value.trim(); // Get and trim email input  
 var password = document.getElementById("password").value; // Get password  
 var confirmPassword = document.getElementById("confirmPassword").value; // Get confirmation password  
  
 if (password !== confirmPassword) { // Check if passwords match  
 alert("❌ Passwords do not match."); // Show error  
 return false; // Stop form submission  
 }  
  
 // Send data to backend for registration  
 fetch("register.php", {  
 method: "POST", // Use POST method  
 headers: {  
 "Content-Type": "application/x-www-form-urlencoded", // Send data as form-encoded  
 },  
 body: new URLSearchParams({ email, password }), // Convert data to URL-encoded string  
 })  
 .then((res) => res.json()) // Parse response as JSON  
 .then((data) => {  
 if (data.status === "error") {  
 alert(data.message); // Show error from backend  
 } else {  
 if (registerForm) registerForm.style.display = "none"; // Hide registration form  
 if (codeSection) codeSection.style.display = "block"; // Show code verification form  
 }  
 })  
 .catch((err) => {  
 alert("Something went wrong."); // General error  
 console.error(err); // Log error to console  
 });  
  
 return false; // Prevent default behavior  
 };  
}  
  
// ✅ Show selected profile image before upload  
window.previewImage = function() {  
 var input = document.getElementById("profileImage"); // File input element  
 var preview = document.getElementById("profileImagePreview"); // Image element to show preview  
 var removeBtn = document.getElementById("removeImageBtn"); // Button to remove selected image  
  
 if (input && input.files && input.files[0]) { // Check if file is selected  
 var reader = new FileReader(); // Create a new file reader  
 reader.onload = function(e) { // When file is loaded  
 if (preview) {  
 preview.src = e.target.result; // Set preview image source  
 preview.style.display = "block"; // Show the preview image  
 }  
 if (removeBtn) removeBtn.style.display = "inline-block"; // Show the remove button  
 }  
 reader.readAsDataURL(input.files[0]); // Convert file to base64 string  
 }  
};  
  
// ✅ Remove profile image preview and reset file input  
window.removeImage = function() {  
 var input = document.getElementById("profileImage"); // File input  
 var preview = document.getElementById("profileImagePreview"); // Image preview  
 var removeBtn = document.getElementById("removeImageBtn"); // Remove button  
  
 if (input) input.value = ""; // Clear file input  
 if (preview) {  
 preview.src = "#"; // Reset image source  
 preview.style.display = "none"; // Hide preview  
 }  
 if (removeBtn) removeBtn.style.display = "none"; // Hide remove button  
};  
  
// ✅ Send verification code to backend and show result (step 2)  
window.verifyCode = function() {  
 var code = document.getElementById("codeInput"); // Input field for the verification code  
 var codeValue = code ? code.value : ""; // Get code value  
  
 fetch("verify.php", {  
 method: "POST", // POST method  
 headers: {  
 "Content-Type": "application/x-www-form-urlencoded", // URL encoded  
 },  
 body: new URLSearchParams({ code: codeValue }), // Attach code  
 })  
 .then((res) => res.json()) // Convert response to JSON  
 .then((data) => {  
 if (data.status === "success") {  
 if (rightSection) {  
 rightSection.classList.add("active"); // Make profile section visible  
 }  
  
 if (message) {  
 message.textContent = "✅ Code Verified! Now you can create your profile."; // Show success message  
 message.className = "verification-message success"; // Style success  
 }  
  
 if (makeProfileBtn) {  
 makeProfileBtn.classList.add("pulsing"); // Add animation  
 setTimeout(function() {  
 makeProfileBtn.classList.remove("pulsing"); // Remove animation after delay  
 }, 3000);  
 }  
  
 if (window.sessionUser && window.sessionUser.email && window.sessionUser.password) {  
 var profileEmail = document.getElementById("profileEmail"); // Email field  
 var profilePassword = document.getElementById("profilePassword"); // Password field  
 if (profileEmail) profileEmail.value = window.sessionUser.email; // Autofill email  
 if (profilePassword) profilePassword.value = window.sessionUser.password; // Autofill password  
 }  
 } else if (message) {  
 message.textContent = data.message; // Show backend error message  
 message.className = "verification-message error"; // Style error  
 }  
 })  
 .catch((err) => {  
 if (message) {  
 message.textContent = "❌ Something went wrong."; // General error message  
 message.className = "verification-message error"; // Style error  
 }  
 console.error(err); // Log error  
 });  
};  
  
// ✅ Final profile submission after verification (step 3)  
if (profileForm) { // If profile form exists  
 profileForm.onsubmit = function(e) {  
 e.preventDefault(); // Prevent form reload  
  
 var formData = new FormData(profileForm); // Collect all form data  
 var code = document.getElementById("codeInput"); // Get verification code input  
 var codeValue = code ? code.value : ""; // Get code value  
  
 formData.append("code", codeValue); // Add verification code to form data  
  
 fetch("verify.php", {  
 method: "POST", // Send form data via POST  
 body: formData,  
 })  
 .then((res) => res.json()) // Convert response to JSON  
 .then((data) => {  
 if (data.status === "success") {  
 if (message) {  
 message.textContent = data.message; // Show success message  
 message.className = "verification-message success"; // Style success  
 }  
 setTimeout(function() {  
 window.location.href = data.redirect; // Redirect to final destination  
 }, 1500);  
 } else if (message) {  
 message.textContent = data.message; // Show error message  
 message.className = "verification-message error"; // Style error  
 }  
 })  
 .catch((err) => {  
 if (message) {  
 message.textContent = "❌ Something went wrong."; // General error  
 message.className = "verification-message error"; // Style error  
 }  
 console.error(err); // Log error  
 });  
  
 return false; // Prevent default form behavior  
 };  
}

————————————————————————————————————————

## Register\_Page\Register.php

<?php  
/\*\*  
 \* User Registration System  
 \*   
 \* This file handles the user registration process including:  
 \* - Email validation and duplicate checking  
 \* - Password hashing and security  
 \* - Email verification code generation and sending  
 \* - Session management for registration flow  
 \*   
 \* Registration Flow:  
 \* 1. User submits registration form  
 \* 2. System validates email isn't already registered  
 \* 3. Generates 6-digit verification code  
 \* 4. Sends verification email  
 \* 5. Stores temporary data in session  
 \* 6. User completes verification in verify.php  
 \*   
 \* @author BOOK-PLAY Development Team  
 \* @version 1.0  
 \* @since 2025  
 \*/  
  
// Start session and include required files  
session\_start(); // Start the session to track user data between requests  
require\_once '../../../db.php'; // Include the database connection file  
require\_once '../../../mail/MailLink.php'; // Include the mail utility to send verification emails  
  
/\*\*  
 \* Handle POST request for user registration  
 \* Processes registration form submission and sends verification email  
 \*/  
if ($\_SERVER['REQUEST\_METHOD'] === 'POST') {  
 $email = $\_POST['email']; // Get the email from the submitted form  
 $password = password\_hash($\_POST['password'], PASSWORD\_DEFAULT); // Hash the password securely for storage  
  
 // Check if email is already registered  
 $checkEmail = $conn->prepare("SELECT email FROM users WHERE email = ?");  
 $checkEmail->bind\_param("s", $email); // Bind the email to the SQL statement  
 $checkEmail->execute(); // Execute the query  
 $checkEmail->store\_result(); // Store the result to check the row count  
  
 // Return error if email already exists  
 if ($checkEmail->num\_rows > 0) {  
 echo json\_encode(["status" => "error", "message" => "❌ This email is already registered."]);  
 exit();  
 }  
  
 // Generate 6-digit verification code  
 $code = rand(100000, 999999);  
  
 // Store verification data in session for later use  
 $\_SESSION['verification\_code'] = $code; // Save the verification code in session  
 $\_SESSION['temp\_email'] = $email; // Temporarily save the email  
 $\_SESSION['temp\_password'] = $password; // Temporarily save the hashed password  
  
 // Send verification email to user  
 if (!sendVerificationCode($email, $code)) {  
 echo json\_encode(["status" => "error", "message" => "❌ Failed to send email."]);  
 exit();  
 }  
  
 // Return success response  
 echo json\_encode(["status" => "success", "message" => "✅ Verification code sent."]);  
 exit();  
}  
  
/\*\*  
 \* Display registration form for GET requests  
 \* Includes the HTML template and injects session data if available  
 \*/  
include "Register.html"; // Load the HTML form for registration  
  
// Inject session data into JavaScript if temporary user data exists  
if (isset($\_SESSION['temp\_email']) && isset($\_SESSION['temp\_password'])) {  
 echo "<script>  
 window.sessionUser = {  
 email: '" . $\_SESSION['temp\_email'] . "', // Assign session email to JS  
 password: '" . $\_SESSION['temp\_password'] . "' // Assign session password to JS  
 };  
 </script>";  
}  
  
exit(); // Final exit to prevent further execution  
?>

————————————————————————————————————————

## Register\_Page\verify.php

<?php  
/\*\*  
 \* User Registration Verification Handler  
 \*   
 \* This file handles the multi-step user registration verification process:  
 \* 1. Email verification code validation  
 \* 2. User profile creation with sports preferences  
 \* 3. Image upload handling  
 \* 4. Database insertion and session cleanup  
 \*   
 \* Features:  
 \* - Email verification code checking  
 \* - Sports list generation for user preferences  
 \* - Profile image upload and storage  
 \* - User data validation and database insertion  
 \* - Favorite sports association  
 \*   
 \* @author BOOK-PLAY Development Team  
 \* @version 1.0  
 \* @since 2025  
 \*/  
  
// Start session and include database connection  
session\_start(); // Start PHP session for user tracking across requests  
require\_once '../../../db.php'; // Include database connection file  
  
/\*\*  
 \* Generate sports list HTML from database  
 \* Creates checkbox options for all accepted sports  
 \*   
 \* @return string HTML string containing sport options  
 \*/  
function getSportsHTML() {  
 global $conn; // Use the global $conn variable for DB connection  
  
 $sportsHTML = ''; // Initialize the variable that will store the HTML  
  
 // Query to get accepted sports from database  
 $result = $conn->query("SELECT \* FROM `sports` WHERE is\_Accepted = 1");  
  
 if ($result->num\_rows > 0) { // If there are sports returned from DB  
 while ($row = $result->fetch\_assoc()) { // Loop through each sport  
 $sportId = $row['sport\_id']; // Get the sport's ID  
 $sport = htmlspecialchars($row['sport\_name']); // Escape special characters for safe HTML output  
  
 $sportsHTML .= "<div class='sport-option'>"; // Start sport option container  
 $sportsHTML .= "<input type='checkbox' id='sport-$sportId' name='favorite\_sports[]' value='$sportId'>"; // Add checkbox input  
 $sportsHTML .= "<label for='sport-$sportId' class='sport-label'>$sport</label>"; // Label for the checkbox  
 $sportsHTML .= "</div>"; // Close sport option container  
 }  
 } else {  
 $sportsHTML = "<p>No sports available</p>"; // Show message if no sports available  
 }  
  
 return $sportsHTML; // Return the full HTML string  
}  
  
/\*\*  
 \* Handle GET request for sports list  
 \* Returns HTML options for sports selection in registration form  
 \*/  
if (isset($\_GET['get\_sports']) && $\_GET['get\_sports'] == 1) {  
 echo getSportsHTML(); // Output the generated sports HTML  
 exit(); // Terminate script to prevent further execution  
}  
  
// Set header for JSON response for all POST requests  
header('Content-Type: application/json'); // Response type is JSON  
  
/\*\*  
 \* Verify email verification code (Step 2 of registration)  
 \* Validates the 6-digit code sent to user's email  
 \*/  
if ($\_SERVER['REQUEST\_METHOD'] === 'POST' && isset($\_POST['code']) && !isset($\_POST['username'])) {  
 $code = $\_POST['code']; // Get the code from the request  
  
 // Compare submitted code with stored session code  
 if (!isset($\_SESSION['verification\_code']) || $code != $\_SESSION['verification\_code']) {  
 echo json\_encode([  
 "status" => "error",  
 "message" => "❌ Incorrect verification code."  
 ]);  
 exit(); // Stop script if invalid code  
 }  
  
 // Code is correct - proceed to profile creation  
 echo json\_encode(["status" => "success"]);  
 exit(); // Terminate script  
}  
  
/\*\*  
 \* Handle profile form submission (Step 3 of registration)  
 \* Creates user profile with personal information and sports preferences  
 \*/  
if ($\_SERVER['REQUEST\_METHOD'] === 'POST' && isset($\_POST['username'])) {  
 // Extract form data with fallback values  
 $username = $\_POST['username'] ?? ''; // Get username  
 $firstName = $\_POST['firstName'] ?? ''; // Get first name  
 $lastName = $\_POST['lastName'] ?? ''; // Get last name  
 $age = intval($\_POST['age'] ?? 0); // Get age as integer  
 $gender = $\_POST['gender'] ?? ''; // Get gender  
 $phone = $\_POST['phone'] ?? ''; // Get phone number  
 $description = $\_POST['description'] ?? ''; // Get description  
 $userImage = $\_FILES['user\_image'] ?? null; // Get uploaded image if exists  
 $favoriteSports = isset($\_POST['favorite\_sports']) ? $\_POST['favorite\_sports'] : []; // Get favorite sports  
  
 // Validate session data exists (email and password from step 1)  
 if (!isset($\_SESSION['temp\_email']) || !isset($\_SESSION['temp\_password'])) {  
 echo json\_encode(["status" => "error", "message" => "⏳ Session expired. Please try again."]);  
 exit(); // Exit if session data is missing  
 }  
  
 // Retrieve stored email and password from session  
 $email = $\_SESSION['temp\_email']; // Retrieve stored email  
 $password = $\_SESSION['temp\_password']; // Retrieve stored password  
  
 // Check if username already exists in database  
 $checkUser = $conn->prepare("SELECT username FROM users WHERE username = ?");  
 $checkUser->bind\_param("s", $username); // Bind username to query  
 $checkUser->execute(); // Execute the query  
 $checkUser->store\_result(); // Store the result to check row count  
  
 if ($checkUser->num\_rows > 0) { // If username is already taken  
 echo json\_encode(["status" => "error", "message" => "❌ Username already exists."]);  
 exit(); // Exit script  
 }  
 $checkUser->close(); // Close statement  
  
 /\*\*  
 \* Handle profile image upload  
 \* Processes uploaded image file and stores it in uploads directory  
 \*/  
 $userImagePath = null; // Default to null  
 if ($userImage && $userImage['tmp\_name']) { // If file uploaded  
 $targetDir = "../../../uploads/users/"; // Directory to store uploads  
 if (!file\_exists($targetDir)) {  
 mkdir($targetDir, 0755, true); // Create directory if not exists  
 }  
 $filename = uniqid() . "\_" . basename($userImage["name"]); // Generate unique file name  
 $targetFile = $targetDir . $filename; // Full path to save image  
  
 if (move\_uploaded\_file($userImage["tmp\_name"], $targetFile)) { // Move uploaded file  
 $userImagePath = $filename; // Save file name for DB  
 }  
 }  
  
 // Check if username already exists  
$stmt = $conn->prepare("SELECT username FROM users WHERE username = ?");  
$stmt->bind\_param("s", $\_POST['username']);  
$stmt->execute();  
$result = $stmt->get\_result();  
  
if ($result->num\_rows > 0) {  
 echo json\_encode([  
 "status" => "error",  
 "message" => "❌ Username already exists. Please choose another one."  
 ]);  
 exit;  
}  
  
  
 // 💾 Insert user data into 'users' table  
 $stmt = $conn->prepare("INSERT INTO users (username, email, first\_name, last\_name, password, description, age, gender, phone\_number, user\_image)  
 VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?)");  
 $stmt->bind\_param("ssssssisss", $username, $email, $firstName, $lastName, $password, $description, $age, $gender, $phone, $userImagePath); // Bind values  
  
 if ($stmt->execute()) { // If insert successful  
 $userId = $conn->insert\_id; // Get inserted user ID  
  
 // ➕ Insert favorite sports into 'user\_favorite\_sports'  
 if (!empty($favoriteSports)) {  
 $sportInsertStmt = $conn->prepare("INSERT INTO user\_favorite\_sports (username, sport\_id) VALUES (?, ?)");  
 foreach ($favoriteSports as $sportId) {  
 $sportInsertStmt->bind\_param("si", $username, $sportId); // Bind sport  
 $sportInsertStmt->execute(); // Execute insert  
 }  
 $sportInsertStmt->close(); // Close statement  
 }  
  
 // 🧹 Clean session after successful registration  
 unset($\_SESSION['temp\_email'], $\_SESSION['temp\_password'], $\_SESSION['verification\_code']);  
  
 echo json\_encode([  
 "status" => "success",  
 "message" => "✅ Profile created successfully.",  
 "redirect" => "../../../index.php" // Redirect user to homepage  
 ]);  
 } else {  
 echo json\_encode(["status" => "error", "message" => "❌ Failed to save user: " . $conn->error]); // Error inserting  
 }  
  
 $stmt->close(); // Close insert statement  
 $conn->close(); // Close DB connection  
 exit(); // End script  
}  
  
// ❌ If none of the above conditions are met, return method error  
echo json\_encode(["status" => "error", "message" => "⛔ Invalid request method."]); // Return generic error  
exit(); // Exit script  
?>

————————————————————————————————————————

## Reset\_Password\Reset\_Password.css

/\* ===================== RESET PASSWORD PAGE SPECIFIC STYLES ===================== \*/  
  
/\* Override body styles for reset password page (no navbar padding needed) \*/  
body {  
 padding-top: 0; /\* Remove navbar padding for reset password page \*/  
 display: flex; /\* Center content \*/  
 min-height: 100vh; /\* Full height \*/  
 justify-content: center; /\* Center horizontally \*/  
 align-items: center; /\* Center vertically \*/  
 overflow-x: hidden; /\* Prevent horizontal scroll \*/  
 padding: 15px; /\* Page padding \*/  
}  
  
/\* ===================== RESET PASSWORD-SPECIFIC LAYOUT ===================== \*/  
  
/\* Reset password container - unique to this page \*/  
.container {  
 background: var(--card-bg);  
 backdrop-filter: blur(10px);  
 border: 1px solid rgba(0, 0, 0, 0.05);  
 padding: 35px 30px;  
 border-radius: var(--radius);  
 box-shadow: var(--shadow);  
 width: 100%;  
 max-width: 420px;  
 text-align: center;  
 animation: fadeIn 0.4s ease;  
}  
  
/\* ===================== RESET PASSWORD-SPECIFIC COMPONENTS ===================== \*/  
  
/\* Step transitions - unique to reset password flow \*/  
.step {  
 display: none; /\* Hide by default \*/  
}  
  
.step.active {  
 display: block; /\* Show active step \*/  
 animation: fadeIn 0.4s ease-in-out; /\* Fade animation \*/  
}  
  
/\* Reset password-specific messages \*/  
.error {  
 color: #ef4444; /\* Red error text \*/  
 font-size: 14px; /\* Font size \*/  
 margin-top: -4px; /\* Slightly above \*/  
}  
  
.message {  
 margin: 8px 0; /\* Margin \*/  
 font-size: 14px; /\* Text size \*/  
}  
  
/\* ===================== RESET PASSWORD-SPECIFIC ANIMATIONS ===================== \*/  
  
@keyframes fadeIn {  
 from {  
 opacity: 0; /\* Hidden \*/  
 transform: translateY(-10px); /\* Move up \*/  
 }  
 to {  
 opacity: 1; /\* Visible \*/  
 transform: translateY(0); /\* Original position \*/  
 }  
}  
  
/\* ===================== RESET PASSWORD-SPECIFIC RESPONSIVE ===================== \*/  
  
/\* Reset password-specific responsive adjustments \*/  
@media screen and (max-width: 768px) {  
 .container {  
 padding: 30px 25px; /\* Smaller padding \*/  
 max-width: 95%; /\* Adjust width \*/  
 }  
}  
  
@media screen and (max-width: 480px) {  
 .container {  
 padding: 25px 20px; /\* Smaller padding \*/  
 max-width: 95%; /\* Adjust width \*/  
 margin: 0 10px; /\* Side margin \*/  
 }  
}  
  
@media screen and (max-width: 350px) {  
 .container {  
 padding: 20px 15px; /\* Tighter padding \*/  
 }  
}

————————————————————————————————————————

## Reset\_Password\Reset\_Password.html

<!-- ===================== RESET PASSWORD PAGE ===================== -->  
<!DOCTYPE html>  
<html lang="en">  
<head>  
 <meta charset="UTF-8">  
 <meta name="viewport" content="width=device-width, initial-scale=1.0">  
 <title>Reset Password</title>  
 <!-- Shared CSS files -->  
 <link rel="stylesheet" href="../../../assets/css/global.css" />  
 <link rel="stylesheet" href="../../../assets/css/components.css" />  
 <link rel="stylesheet" href="../../../assets/css/layouts.css" />  
 <link rel="stylesheet" href="../../../assets/css/responsive.css" />  
 <!-- Reset Password-specific CSS -->  
 <link rel="stylesheet" href="Reset\_Password.css">  
 <!-- Shared JavaScript -->  
 <script src="../../../assets/js/shared.js"></script>  
 <!-- Reset Password-specific JavaScript -->  
 <script src="Reset\_Password.js"></script>  
</head>  
<body>  
 <!-- ===================== MAIN CONTAINER ===================== -->  
 <div class="container">  
 <form id="resetForm">  
 <!-- ========== STEP 1: Enter Email ========== -->  
 <div class="step active" id="step1">  
 <h2>Enter Your Email</h2>  
 <input type="email" id="email" placeholder="Enter your email" required>  
 <div id="message-step1" class="message"></div>  
 <button type="button" onclick="sendCode()">Send Code</button>  
 <button type="button" class="back-btn" onclick="window.location.href='../Login\_Page/login.php'">Back</button>  
 </div>  
  
 <!-- ========== STEP 2: Enter Verification Code ========== -->  
 <div class="step" id="step2">  
 <h2>Enter Verification Code</h2>  
 <input type="text" id="code" placeholder="Enter code" required>  
 <div id="message-step2" class="message"></div>  
 <button type="button" id="verifyCodeBtn">Verify</button>  
 <button type="button" class="back-btn" onclick="showStep(1)">Back</button>  
 </div>  
  
 <!-- ========== STEP 3: Reset Password ========== -->  
 <div class="step" id="step3">  
 <h2>Reset Password</h2>  
 <input type="password" id="password" placeholder="New Password" required>  
 <input type="password" id="confirmPassword" placeholder="Confirm Password" required>  
 <div id="message-step3" class="message"></div>  
 <button type="button" id="resetbtn">Reset Password</button>  
 <button type="button" class="back-btn" onclick="showStep(2)">Back</button>  
 </div>  
 </form>  
 </div>  
 <!-- ===================== END MAIN CONTAINER ===================== -->  
</body>  
</html>  
<!-- ===================== END RESET PASSWORD PAGE ===================== -->

————————————————————————————————————————

## Reset\_Password\Reset\_Password.js

/\*\*  
\* Password Reset Flow - Client-side implementation  
\*   
\* This script handles the 3-step password reset process:  
\* 1. Email submission and verification code sending  
\* 2. Verification code validation  
\* 3. New password creation and submission  
\*/  
  
document.addEventListener("DOMContentLoaded", function () {  
 // Get DOM elements for input fields  
 const emailInput = document.getElementById("email");  
 const codeInput = document.getElementById("code");  
 const newPasswordInput = document.getElementById("password");  
 const confirmPasswordInput = document.getElementById("confirmPassword");  
 const message = document.getElementById("message");  
 const messageStep1 = document.getElementById("message-step1");  
const messageStep2 = document.getElementById("message-step2");  
const messageStep3 = document.getElementById("message-step3");  
  
 // Get DOM elements for the different steps of the reset flow  
 const step1 = document.getElementById("step1");  
 const step2 = document.getElementById("step2");  
 const step3 = document.getElementById("step3");  
  
 /\*\*  
 \* Controls which step of the password reset flow is visible  
 \* @param {number} stepNumber - The step to display (1, 2, or 3)  
 \*/  
window.showStep = function(stepNumber){  
 step1.style.display = stepNumber === 1 ? "block" : "none";  
 step2.style.display = stepNumber === 2 ? "block" : "none";  
 step3.style.display = stepNumber === 3 ? "block" : "none";  
  
 // Clear all messages  
 messageStep1.textContent = "";  
 messageStep2.textContent = "";  
 messageStep3.textContent = "";  
};  
  
 // Initialize the flow at step 1  
 showStep(1);  
  
 /\*\*  
 \* Sends a verification code to the provided email  
 \* This function is called when the "Send Code" button is clicked  
 \*/  
window.sendCode = function () {  
 const email = emailInput.value.trim();  
 messageStep1.textContent = "";  
  
 if (!email) {  
 messageStep1.textContent = "📭 Please enter your email.";  
 return;  
 }  
  
 // ✅ Show "Verifying..." message  
 messageStep1.textContent = "⏳ Verifying email...";  
  
 fetch("send\_code.php", {  
 method: "POST",  
 headers: { "Content-Type": "application/x-www-form-urlencoded" },  
 body: new URLSearchParams({ action: "send\_code", email })  
 })  
 .then(res => res.json())  
 .then(data => {  
 messageStep1.textContent = data.message;  
 if (data.success) showStep(2);  
 })  
 .catch(() => {  
 messageStep1.textContent = "❌ Could not send verification code.";  
 });  
};  
  
  
 /\*\*  
 \* Verifies the code entered by the user  
 \* Event listener for the "Verify Code" button  
 \*/  
document.getElementById("verifyCodeBtn").addEventListener("click", function () {  
 const code = codeInput.value.trim();  
 messageStep2.textContent = "";  
  
 if (!code) {  
 messageStep2.textContent = "⛔ Please enter the code.";  
 return;  
 }  
  
 // ✅ Show "Verifying..." message  
 messageStep2.textContent = "🔍 Verifying code...";  
  
 fetch("send\_code.php", {  
 method: "POST",  
 headers: { "Content-Type": "application/x-www-form-urlencoded" },  
 body: new URLSearchParams({ action: "verify\_code", code })  
 })  
 .then(res => res.json())  
 .then(data => {  
 messageStep2.textContent = data.message;  
 if (data.success) showStep(3);  
 })  
 .catch(() => {  
 messageStep2.textContent = "❌ Verification failed.";  
 });  
});  
  
  
 /\*\*  
 \* Submits the new password to the server  
 \* Event listener for the "Reset Password" button  
 \*/  
document.getElementById("resetbtn").addEventListener("click", function () {  
 const newPassword = newPasswordInput.value;  
 const confirmPassword = confirmPasswordInput.value;  
 messageStep3.textContent = "";  
  
 if (!newPassword || !confirmPassword) {  
 messageStep3.textContent = "⚠️ Please fill out both password fields.";  
 return;  
 }  
  
 if (newPassword !== confirmPassword) {  
 messageStep3.textContent = "❗ Passwords do not match.";  
 return;  
 }  
  
 fetch("send\_code.php", {  
 method: "POST",  
 headers: { "Content-Type": "application/x-www-form-urlencoded" },  
 body: new URLSearchParams({  
 action: "reset\_password",  
 password: newPassword,  
 confirmPassword: confirmPassword  
 })  
 })  
 .then(res => res.json())  
 .then(data => {  
 messageStep3.textContent = data.message;  
 if (data.success) {  
 messageStep3.textContent += " Redirecting...";  
 setTimeout(() => {  
 window.location.href = "../Login\_Page/Login.php";  
 }, 2000);  
 }  
 })  
 .catch(() => {  
 messageStep3.textContent = "❌ Failed to reset password.";  
 });  
});   
});

————————————————————————————————————————

## Reset\_Password\Reset\_Password.php

<?php  
session\_start();  
$htmlPath = \_\_DIR\_\_ . "/Reset\_Password.html";  
  
if (file\_exists($htmlPath)) {  
 readfile($htmlPath);  
} else {  
 echo "❌ Reset\_Password.html not found!";  
}  
?>

————————————————————————————————————————

## Reset\_Password\send\_code.php

<?php  
// Start session to maintain user state across requests  
session\_start();  
// Connect to database  
require\_once '../../../db.php';  
// Include email sending functionality  
require\_once '../../../mail/MailLink.php';  
  
// Set response header to JSON  
header('Content-Type: application/json');  
  
// Get the requested action from POST data (with null coalescing operator)  
$action = $\_POST['action'] ?? '';  
  
// SECTION 1: SEND VERIFICATION CODE  
// Handles sending a verification code to the user's email for password reset  
if ($action === 'send\_code') {  
 // Get email from POST data  
 $email = $\_POST['email'] ?? '';  
  
 // Validate email is not empty  
 if (empty($email)) {  
 echo json\_encode(["success" => false, "message" => "📭 Please enter your email."]);  
 exit;  
 }  
  
 // Check if email exists in the database  
 $stmt = $conn->prepare("SELECT username FROM users WHERE email = ?");  
 $stmt->bind\_param("s", $email);  
 $stmt->execute();  
 $result = $stmt->get\_result();  
  
 // If email not found, return error  
 if ($result->num\_rows === 0) {  
 echo json\_encode(["success" => false, "message" => "❌ Email not registered."]);  
 exit;  
 }  
   
 // Get the username associated with the email  
 $row = $result->fetch\_assoc();  
 $\_SESSION['username'] = $row['username']; // Store username in session  
   
 // Generate a random 6-digit verification code  
 $code = rand(100000, 999999);  
   
 // Store code and email in session for later verification  
 $\_SESSION['reset\_code'] = $code;  
 $\_SESSION['reset\_email'] = $email;  
  
 // Send verification code email and return appropriate response  
 if (sendVerificationCode($email, $code)) {  
 echo json\_encode(["success" => true, "message" => "✅ Verification code sent."]);  
 } else {  
 echo json\_encode(["success" => false, "message" => "❌ Failed to send email."]);  
 }  
 exit;  
}  
  
// SECTION 2: VERIFY CODE  
// Validates the verification code entered by the user  
if ($action === 'verify\_code') {  
 // Get code from POST data  
 $code = $\_POST['code'] ?? '';  
  
 // Check if code matches the one stored in session  
 if (!isset($\_SESSION['reset\_code']) || $code != $\_SESSION['reset\_code']) {  
 echo json\_encode(["success" => false, "message" => "❌ Incorrect verification code."]);  
 } else {  
 echo json\_encode(["success" => true, "message" => "✅ Code verified."]);  
 }  
 exit;  
}  
  
// SECTION 3: RESET PASSWORD  
// Handles the actual password reset after verification  
if ($action === 'reset\_password') {  
 // Get password data from POST  
 $newPassword = $\_POST['password'] ?? '';  
 $confirmPassword = $\_POST['confirmPassword'] ?? '';  
  
 // Validate that passwords match  
 if ($newPassword !== $confirmPassword) {  
 echo json\_encode(["success" => false, "message" => "❌ Passwords do not match."]);  
 exit;  
 }  
  
 // Ensure session is still valid  
 if (!isset($\_SESSION['reset\_email'])) {  
 echo json\_encode(["success" => false, "message" => "⏳ Session expired. Try again."]);  
 exit;  
 }  
  
 // Hash the new password for security  
 $hashedPassword = password\_hash($newPassword, PASSWORD\_DEFAULT);  
 $email = $\_SESSION['reset\_email'];  
  
 // Update the password in the database  
 $stmt = $conn->prepare("UPDATE users SET password = ? WHERE email = ?");  
 $stmt->bind\_param("ss", $hashedPassword, $email);  
  
 // Execute query and return appropriate response  
 if ($stmt->execute()) {  
 // Clear session variables for security  
 unset($\_SESSION['reset\_email'], $\_SESSION['reset\_code']);  
 echo json\_encode(["success" => true, "message" => "✅ Password updated."]);  
 } else {  
 echo json\_encode(["success" => false, "message" => "❌ Failed to update password."]);  
 }  
 exit;  
}  
  
// If no valid action was specified, return error  
echo json\_encode(["success" => false, "message" => "⛔ Invalid action."]);  
exit;  
?>

————————————————————————————————————————

## User\_Selection\_Page\is\_admin.php

<?php  
// check-admin-status.php - Returns user's admin status as JSON  
  
// Start session if not already started  
if (session\_status() === PHP\_SESSION\_NONE) {  
 session\_start();  
}  
  
// Include database connection  
include '../../../db.php';  
  
// Default response  
$response = ['is\_admin' => 0];  
  
// Check if user is logged in  
if (isset($\_SESSION['user\_id'])) {  
 $userId = $\_SESSION['user\_id'];  
   
 // Prepare SQL to get is\_admin status  
 $stmt = $conn->prepare("SELECT is\_admin FROM users WHERE username = ?");  
 $stmt->bind\_param("s", $userId);  
 $stmt->execute();  
 $result = $stmt->get\_result();  
   
 if ($row = $result->fetch\_assoc()) {  
 $response['is\_admin'] = (int)$row['is\_admin'];  
 }  
   
 $stmt->close();  
}   
  
// Set header to JSON content type  
header('Content-Type: application/json');  
  
// Return JSON response  
echo json\_encode($response);  
?>

————————————————————————————————————————

## User\_Selection\_Page\user-selection.css

/\* ===================== USER SELECTION PAGE SPECIFIC STYLES ===================== \*/  
:root{  
 --tertiary: #9333ea; /\* Purple for Admin \*/  
  
}  
/\* Override body styles for user selection page (no navbar padding needed) \*/  
body {  
 padding-top: 0; /\* Remove navbar padding for user selection page \*/  
 display: flex; /\* Use Flexbox \*/  
 justify-content: center; /\* Center horizontally \*/  
 align-items: center; /\* Center vertically \*/  
 flex-direction: column; /\* Stack children vertically \*/  
 padding: 20px; /\* Padding around content \*/  
}  
  
/\* ===================== USER SELECTION-SPECIFIC LAYOUT ===================== \*/  
  
/\* User selection container - unique to this page \*/  
.container {  
 text-align: center; /\* Center align text \*/  
 padding: 40px; /\* Inner padding \*/  
 background: var(--card-bg); /\* Light transparent card background \*/  
 backdrop-filter: blur(14px); /\* Blur effect \*/  
 border-radius: var(--radius); /\* Rounded corners \*/  
 box-shadow: 0 0 30px rgba(30, 144, 255, 0.1); /\* Subtle shadow \*/  
 max-width: 1100px; /\* Max container width \*/  
 width: 95%; /\* Responsive width \*/  
 border: 1px solid var(--card-border); /\* Border color \*/  
 margin: 20px 0; /\* Top and bottom margin \*/  
}  
  
.container h1 {  
 font-size: 32px; /\* Main heading size \*/  
 margin-bottom: 10px; /\* Space below heading \*/  
 background: linear-gradient(to right, var(--primary), var(--secondary)); /\* Gradient text \*/  
 -webkit-background-clip: text; /\* Clip text for gradient \*/  
 background-clip: text;  
 -webkit-text-fill-color: transparent; /\* Make text transparent \*/  
}  
  
.container p {  
 font-size: 16px; /\* Paragraph text size \*/  
 color: var(--muted-text); /\* Muted text color \*/  
 margin-bottom: 30px; /\* Space below paragraph \*/  
}  
  
/\* ===================== USER SELECTION-SPECIFIC COMPONENTS ===================== \*/  
  
/\* User options container \*/  
.user-options {  
 display: flex; /\* Arrange cards in a row \*/  
 justify-content: center; /\* Center cards \*/  
 gap: 30px; /\* Space between cards \*/  
 flex-wrap: wrap; /\* Wrap on small screens \*/  
}  
  
/\* User card styling \*/  
.user-card {  
 background: #ffffffcc; /\* Semi-transparent white \*/  
 backdrop-filter: blur(10px); /\* Glass effect \*/  
 padding: 30px; /\* Inner padding \*/  
 border-radius: var(--radius); /\* Rounded corners \*/  
 width: 260px; /\* Fixed width \*/  
 text-align: center; /\* Center content \*/  
 cursor: pointer; /\* Pointer cursor on hover \*/  
 transition: transform 0.3s, box-shadow 0.3s; /\* Smooth hover effect \*/  
 box-shadow: 0 0 14px rgba(0,0,0,0.05); /\* Subtle shadow \*/  
 border: 1px solid var(--card-border); /\* Card border \*/  
 flex: 1 1 260px; /\* Flex behavior \*/  
 min-width: 240px; /\* Minimum width \*/  
 max-width: 320px; /\* Maximum width \*/  
}  
  
.user-card img {  
 width: 170px; /\* Image width \*/  
 height: 170px; /\* Image height \*/  
 margin-bottom: 15px; /\* Space below image \*/  
 filter: drop-shadow(0 0 12px rgba(23, 107, 190, 0.3)); /\* Glow effect \*/  
 object-fit: contain; /\* Keep aspect ratio \*/  
}  
  
.user-card h2 {  
 font-size: 20px; /\* Card title size \*/  
 margin-bottom: 10px; /\* Space below title \*/  
 background: linear-gradient(to right, var(--primary), var(--secondary)); /\* Gradient text \*/  
 -webkit-background-clip: text; /\* Clip background to text \*/  
 background-clip: text;  
 -webkit-text-fill-color: transparent; /\* Transparent fill \*/  
}  
  
.user-card p {  
 font-size: 14px; /\* Card description size \*/  
 color: var(--muted-text); /\* Muted color \*/  
}  
  
/\* ===================== USER SELECTION-SPECIFIC HOVER EFFECTS ===================== \*/  
  
.user-card.player:hover {  
 transform: scale(1.05); /\* Zoom effect \*/  
 box-shadow: 0 0 60px rgba(30, 144, 255, 0.2); /\* Blue shadow \*/  
}  
  
.user-card.owner:hover {  
 transform: scale(1.05); /\* Zoom effect \*/  
 box-shadow: 0 0 60px rgba(34, 197, 94, 0.2); /\* Green shadow \*/  
}  
  
.user-card.admin:hover {  
 transform: scale(1.05); /\* Zoom effect \*/  
 box-shadow: 0 0 60px rgba(147, 51, 234, 0.2); /\* Purple shadow \*/  
}  
  
.user-card.admin h2 {  
 background: linear-gradient(to right, var(--primary), var(--tertiary)); /\* Purple for Admin \*/  
 -webkit-background-clip: text;  
 background-clip: text;  
 -webkit-text-fill-color: transparent;  
}  
  
.user-card.admin img {  
 filter: drop-shadow(0 0 12px rgba(147, 51, 234, 0.3)); /\* Purple glow \*/  
}  
  
/\* ===================== USER SELECTION-SPECIFIC RESPONSIVE ===================== \*/  
  
/\* User selection-specific responsive adjustments \*/  
@media screen and (max-width: 992px) {  
 .container {  
 padding: 30px; /\* Reduce padding \*/  
 }  
 .user-card {  
 padding: 25px; /\* Reduce card padding \*/  
 }  
 .user-card img {  
 width: 150px;  
 height: 150px;  
 }  
}  
  
@media screen and (max-width: 768px) {  
 .container h1 {  
 font-size: 28px; /\* Smaller heading \*/  
 }  
 .user-options {  
 gap: 20px; /\* Reduce gap between cards \*/  
 }  
 .user-card {  
 flex: 1 1 200px;  
 min-width: 200px;  
 }  
}  
  
@media screen and (max-width: 576px) {  
 .container {  
 padding: 25px 20px; /\* Less padding \*/  
 width: 100%; /\* Full width \*/  
 }  
 .container h1 {  
 font-size: 24px; /\* Smaller heading \*/  
 }  
 .container p {  
 font-size: 14px; /\* Smaller paragraph \*/  
 margin-bottom: 20px;  
 }  
 .user-card {  
 padding: 20px; /\* Reduce card padding \*/  
 flex: 1 1 100%;  
 max-width: 260px;  
 }  
 .user-card img {  
 width: 130px;  
 height: 130px;  
 }  
 .user-card h2 {  
 font-size: 18px; /\* Smaller title \*/  
 }  
 .user-card p {  
 font-size: 13px; /\* Smaller text \*/  
 }  
}  
  
@media screen and (max-width: 350px) {  
 .container {  
 padding: 20px 15px; /\* Tighter padding \*/  
 }  
 .user-card {  
 padding: 15px; /\* Smallest card padding \*/  
 }  
 .user-card img {  
 width: 110px;  
 height: 110px; /\* Smallest image \*/  
 }  
}

————————————————————————————————————————

## User\_Selection\_Page\user-selection.html

<!-- ===================== START: User Type Selection Page ===================== -->  
<!DOCTYPE html>  
<html lang="en">  
<head>  
 <meta charset="UTF-8">  
 <meta name="viewport" content="width=device-width, initial-scale=1.0">  
 <title>Select User Type | Book&Play</title>  
 <!-- Shared CSS files -->  
 <link rel="stylesheet" href="../../../assets/css/global.css" />  
 <link rel="stylesheet" href="../../../assets/css/components.css" />  
 <link rel="stylesheet" href="../../../assets/css/layouts.css" />  
 <link rel="stylesheet" href="../../../assets/css/responsive.css" />  
 <!-- User Selection-specific CSS -->  
 <link rel="stylesheet" href="user-selection.css">  
 <!-- Shared JavaScript -->  
 <script src="../../../assets/js/shared.js"></script>  
 <!-- User Selection-specific JavaScript -->  
 <script src="UserSelection.js"></script>  
</head>  
<body>  
 <!-- ===================== START: Main Container ===================== -->  
 <div class="container">  
 <h1>Select Your User Type</h1>  
 <p>Choose whether you are a Player, Facility Owner, or Admin</p>  
  
 <!-- ===================== START: User Options ===================== -->  
 <div class="user-options">  
 <!-- Player Card -->  
 <div class="user-card player" id="player" onclick="window.location.href='../../player/HomePage/HomePage.php'">  
 <img src="../../../Images/player\_icon.png" alt="Player">  
 <h2>Player</h2>  
 <p>Find games, join groups, and connect with other players!</p>  
 </div>  
  
 <!-- Facility Owner Card -->  
 <div class="user-card owner" id="facilityOwner" onclick="window.location.href='../../Facility\_Owner/HomePage/HomePage.php'">  
 <img src="../../../Images/staduim\_icon.png" alt="Facility Owner">  
 <h2>Facility Owner</h2>  
 <p>Manage your sports facility, book reservations, and connect with players.</p>  
 </div>  
  
 <!-- Admin Card (Initially Hidden) -->  
 <div class="user-card admin" id="admin" style="display: none;" onclick="window.location.href='../../Admin/HomePage/HomePage.php'">  
 <img src="../../../Images/Admin\_icon.png" alt="Admin">  
 <h2>Admin</h2>  
 <p>Manage the platform, monitor activities, and ensure smooth operations.</p>  
 </div>  
 </div>  
 <!-- ===================== END: User Options ===================== -->  
 </div>  
 <!-- ===================== END: Main Container ===================== -->  
</body>  
</html>  
<!-- ===================== END: User Type Selection Page ===================== -->

————————————————————————————————————————

## User\_Selection\_Page\user-selection.php

<?php  
session\_start();  
if (!isset($\_SESSION['user\_id'])) {  
 // Unset all session variables  
 session\_unset();  
 // Destroy the session completely  
 session\_destroy();  
 header('Location: ../Login\_Page/Login.php');  
 exit();  
}  
include 'user-selection.html';  
exit();  
?>

————————————————————————————————————————

## User\_Selection\_Page\UserSelection.js

// ================== START: User Selection Script ==================  
  
// Wait for the DOM to be fully loaded  
// DOMContentLoaded ensures the script runs after HTML is ready  
document.addEventListener('DOMContentLoaded', function() {  
 checkAdminStatus(); // Call function to check if the user is an admin  
});  
  
// Function to check if the user is an admin  
function checkAdminStatus() {  
 const xhr = new XMLHttpRequest(); // Create a new AJAX request  
 xhr.open('GET', 'is\_admin.php', true); // Send GET request to backend  
  
 // When the response is received  
 xhr.onload = function() {  
 if (this.status === 200) { // Check if request was successful  
 const response = JSON.parse(this.responseText); // Parse JSON response  
 const adminCard = document.getElementById('admin'); // Get the admin card div  
  
 if (response.is\_admin === 1) {  
 adminCard.style.display = 'block'; // Show admin card if user is admin  
 } else {  
 adminCard.style.display = 'none'; // Hide it otherwise  
 }  
 }  
 };  
  
 // Handle request error  
 xhr.onerror = function() {  
 console.error('Request error'); // Log error  
 };  
  
 xhr.send(); // Send the AJAX request  
}  
  
// ================== END: User Selection Script ==================

————————————————————————————————————————

## Owner.css

:root {  
 --primary: #1e90ff;  
 --secondary: #22c55e;  
 --background: linear-gradient(135deg, #e0f7fa, #d0f8ce);  
 --card-bg: rgba(255, 255, 255, 0.85);  
 --text: #0f172a;  
 --text-muted: #64748b;  
 --shadow: 0 6px 20px rgba(30, 144, 255, 0.1);  
 --radius: 16px;  
 --transition: all 0.3s ease;  
}  
  
body {  
 font-family: 'Outfit', sans-serif;  
 margin: 0;  
 background: var(--background);  
 color: var(--text);  
}  
  
.navbar {  
 background: white;  
 padding: 15px 30px;  
 box-shadow: var(--shadow);  
 font-size: 24px;  
 font-weight: bold;  
}  
  
.logo span {  
 color: var(--primary);  
}  
  
.section-title {  
 text-align: center;  
 margin: 30px 0;  
 font-size: 28px;  
 color: var(--text);  
}  
  
/\* Table Container \*/  
.table-container {  
 margin: 0 auto;  
 width: 90%;  
 background: var(--card-bg);  
 border-radius: var(--radius);  
 padding: 24px;  
 box-shadow: var(--shadow);  
 backdrop-filter: blur(10px);  
}  
  
/\* Table \*/  
table {  
 width: 100%;  
 border-collapse: collapse;  
}  
  
th, td {  
 padding: 14px;  
 text-align: center;  
 font-size: 15px;  
}  
  
th {  
 background: linear-gradient(to right, #4fc3f7, #81c784);  
 color: white;  
 font-weight: 600;  
 border-top-left-radius: var(--radius);  
 border-top-right-radius: var(--radius);  
}  
  
td {  
 background: rgba(255, 255, 255, 0.95);  
 border-bottom: 1px solid #e2e8f0;  
}  
  
/\* Buttons \*/  
.accept,  
.reject,  
.confirm-btn,  
.add-admin-btn {  
 border: none;  
 padding: 10px 16px;  
 border-radius: var(--radius);  
 color: white;  
 font-weight: 500;  
 cursor: pointer;  
 box-shadow: var(--shadow);  
 transition: var(--transition);  
}  
  
.accept {  
 background: linear-gradient(to right, #34d399, #22c55e);  
}  
  
.reject {  
 background: linear-gradient(to right, #f87171, #ef4444);  
 margin-left: 10px;  
}  
  
.add-admin-btn {  
 display: block;  
 margin: 30px auto;  
 font-size: 16px;  
 background: linear-gradient(to right, #4fc3f7, #1e90ff);  
}  
  
.add-admin-btn:hover,  
.confirm-btn:hover,  
.accept:hover,  
.reject:hover {  
 opacity: 0.9;  
 transform: scale(1.02);  
}  
  
/\* Modal \*/  
.modal {  
 display: none;  
 position: fixed;  
 top: 0; left: 0;  
 width: 100%; height: 100%;  
 background: rgba(0, 0, 0, 0.4);  
 backdrop-filter: blur(3px);  
 z-index: 1000;  
}  
  
.modal-content {  
 background: var(--card-bg);  
 border-radius: var(--radius);  
 max-width: 420px;  
 padding: 30px;  
 margin: 100px auto;  
 box-shadow: var(--shadow);  
 text-align: center;  
 backdrop-filter: blur(10px);  
}  
  
.modal-content input {  
 width: 100%;  
 padding: 12px;  
 margin: 16px 0;  
 border-radius: var(--radius);  
 border: 1px solid #cbd5e1;  
 box-shadow: var(--shadow);  
 font-size: 1rem;  
 outline: none;  
}  
  
.confirm-btn {  
 background: linear-gradient(to right, #4fc3f7, #1e90ff);  
}  
  
.close {  
 float: right;  
 font-size: 22px;  
 cursor: pointer;  
 color: #64748b;  
 transition: var(--transition);  
}  
  
.close:hover {  
 color: #1e90ff;  
}

————————————————————————————————————————

## Owner.html

<!DOCTYPE html>  
<html lang="en">  
<head>  
 <meta charset="UTF-8">  
 <title>Owner Panel | Book&Play</title>  
 <link rel="stylesheet" href="owner.css">  
 <link href="https://fonts.googleapis.com/css2?family=Outfit:wght@400;600&display=swap" rel="stylesheet">  
</head>  
<body>  
 <nav class="navbar">  
 <div class="logo">Book<span>&</span>Play</div>  
 </nav>  
  
 <h2 class="section-title">Manage Admin Requests</h2>  
  
 <div class="table-container">  
 <table>  
 <thead>  
 <tr>  
 <th>Username</th>  
 <th>Email</th>  
 <th>Action</th>  
 </tr>  
 </thead>  
 <tbody>  
 <tr>  
 <td>ali\_admin</td>  
 <td>ali@example.com</td>  
 <td>  
 <button class="accept">✔ Accept</button>  
 <button class="reject">✖ Reject</button>  
 </td>  
 </tr>  
 <tr>  
 <td>sara\_request</td>  
 <td>sara@bookplay.com</td>  
 <td>  
 <button class="accept">✔ Accept</button>  
 <button class="reject">✖ Reject</button>  
 </td>  
 </tr>  
 </tbody>  
 </table>  
 </div>  
  
 <button class="add-admin-btn" onclick="openModal()">+ Add Admin</button>  
  
 <!-- Modal -->  
 <div class="modal" id="adminModal">  
 <div class="modal-content">  
 <span class="close" onclick="closeModal()">&times;</span>  
 <h3>Add New Admin</h3>  
 <input type="text" placeholder="Enter username..." id="newAdminInput">  
 <button class="confirm-btn">Add Admin</button>  
 </div>  
 </div>  
  
 <script src="owner.js"></script>  
</body>  
</html>

————————————————————————————————————————

## Owner.js

function openModal() {  
 document.getElementById("adminModal").style.display = "block";  
}  
  
function closeModal() {  
 document.getElementById("adminModal").style.display = "none";  
}

————————————————————————————————————————

## Owner.php

<?php  
include 'Owner.html';  
?>

————————————————————————————————————————

## components.css

/\* components.css \*/  
  
/\* NAVBAR \*/  
.fresh-navbar {  
 position: fixed;  
 top: 0;  
 left: 0;  
 right: 0;  
 height: var(--navbar-height);  
 z-index: 1000;  
 display: flex;  
 justify-content: space-between;  
 align-items: center;  
 padding: 12px 24px;  
 background: linear-gradient(135deg, rgba(224, 247, 250, 0.95), rgba(208, 248, 206, 0.95));  
 backdrop-filter: blur(12px);  
 border-bottom: 1px solid var(--border-subtle);  
 box-shadow: 0 1px 3px var(--shadow-light), 0 1px 2px var(--shadow-medium);  
}  
  
.logo {  
 font-weight: bold;  
 font-size: 20px;  
 background: linear-gradient(135deg, var(--primary), var(--secondary));  
 -webkit-background-clip: text;  
 -webkit-text-fill-color: transparent;  
}  
  
.nav-wrapper {  
 display: flex;  
 align-items: center;  
}  
  
.nav-links {  
 list-style: none;  
 display: flex;  
 gap: 8px;  
 align-items: center;  
}  
  
.nav-links li a {  
 padding: 10px 20px;  
 border-radius: 24px;  
 font-size: 14px;  
 font-weight: 500;  
 text-decoration: none;  
 color: var(--text-main);  
 background: var(--card-bg);  
 border: 1px solid var(--border-subtle);  
 transition: var(--transition);  
 box-shadow: 0 1px 3px var(--shadow-light);  
}  
  
.nav-links li a:hover {  
 transform: translateY(-1px);  
 box-shadow: 0 4px 12px var(--shadow-light);  
 background: linear-gradient(to right, #4fc3f7, #81c784);  
 color: white;  
 border-color: transparent;  
}  
  
.nav-links li a.active {  
 color: white;  
 background: linear-gradient(to right, #4fc3f7, #81c784);  
 font-weight: 600;  
 border-color: transparent;  
 box-shadow: 0 2px 8px rgba(30, 144, 255, 0.25);  
}  
  
.nav-links li a.logout {  
 background: rgba(239, 68, 68, 0.1);  
 color: #dc2626;  
 font-weight: 600;  
 border-color: rgba(239, 68, 68, 0.2);  
}  
  
.nav-links li a.logout:hover {  
 background: #dc2626;  
 color: white;  
 border-color: #dc2626;  
}  
  
/\* FACILITY OWNER NAVBAR \*/  
.facility-owner-navbar {  
 position: fixed;  
 top: 0;  
 left: 0;  
 right: 0;  
 height: var(--navbar-height);  
 z-index: 1000;  
 display: flex;  
 justify-content: space-between;  
 align-items: center;  
 padding: 12px 24px;  
 background: linear-gradient(135deg, rgba(224, 247, 250, 0.95), rgba(208, 248, 206, 0.95));  
 backdrop-filter: blur(12px);  
 border-bottom: 1px solid var(--border-subtle);  
 box-shadow: 0 1px 3px var(--shadow-light), 0 1px 2px var(--shadow-medium);  
}  
  
.facility-owner-navbar .nav-links {  
 list-style: none;  
 display: flex;  
 gap: 8px;  
 align-items: center;  
}  
  
.facility-owner-navbar .nav-links li a {  
 padding: 10px 20px;  
 border-radius: 24px;  
 font-size: 14px;  
 font-weight: 500;  
 text-decoration: none;  
 color: var(--text-main);  
 background: var(--card-bg);  
 border: 1px solid var(--border-subtle);  
 transition: var(--transition);  
 box-shadow: 0 1px 3px var(--shadow-light);  
}  
  
.facility-owner-navbar .nav-links li a:hover {  
 transform: translateY(-1px);  
 box-shadow: 0 4px 12px var(--shadow-light);  
 background: linear-gradient(to right, #4fc3f7, #81c784);  
 color: white;  
 border-color: transparent;  
}  
  
.facility-owner-navbar .nav-links li a.active {  
 color: white;  
 background: linear-gradient(to right, #4fc3f7, #81c784);  
 font-weight: 600;  
 border-color: transparent;  
 box-shadow: 0 2px 8px rgba(30, 144, 255, 0.25);  
}  
  
.facility-owner-navbar .nav-links li a.logout {  
 background: rgba(239, 68, 68, 0.1);  
 color: #dc2626;  
 font-weight: 600;  
 border-color: rgba(239, 68, 68, 0.2);  
}  
  
.facility-owner-navbar .nav-links li a.logout:hover {  
 background: #dc2626;  
 color: white;  
 border-color: #dc2626;  
}  
  
/\* SPORTS SCROLL \*/  
.sports-scroll {  
 display: flex;  
 gap: 12px;  
 overflow-x: auto;  
 padding: 16px 24px;  
 background: linear-gradient(135deg, rgba(224, 247, 250, 0.4), rgba(208, 248, 206, 0.4));  
 scrollbar-width: thin;  
 scrollbar-color: var(--primary) transparent;  
}  
  
.sports-scroll::-webkit-scrollbar {  
 height: 6px;  
}  
  
.sports-scroll::-webkit-scrollbar-thumb {  
 background: var(--primary);  
 border-radius: 3px;  
}  
  
/\* Make each sport-item clickable label \*/  
.sport-item {  
 flex-shrink: 0;  
 position: relative;  
 display: inline-block;  
 cursor: pointer;  
}  
  
/\* Hide the checkbox input \*/  
.sport-checkbox {  
 display: none;  
}  
  
/\* Base style for sport circle \*/  
.sport-circle {  
 width: 70px;  
 height: 70px;  
 border-radius: 50%;  
 background: var(--card-bg);  
 font-size: 11px;  
 font-weight: 500;  
 text-align: center;  
 display: flex;  
 flex-direction: column;  
 align-items: center;  
 justify-content: center;  
 gap: 3px;  
 box-shadow: 0 2px 8px rgba(0,0,0,0.1);  
 transition: var(--transition);  
 border: 2px solid transparent;  
}  
  
/\* Hover effect \*/  
.sport-circle:hover {  
 transform: translateY(-2px);  
 background: rgba(30, 144, 255, 0.05);  
 border-color: rgba(30, 144, 255, 0.2);  
}  
  
/\* Selected effect when checkbox is checked \*/  
.sport-checkbox:checked + .sport-circle {  
 background: linear-gradient(135deg, var(--primary), var(--secondary));  
 color: white;  
 box-shadow: 0 4px 12px rgba(30, 144, 255, 0.3);  
 border-color: rgba(30, 144, 255, 0.4);  
 transform: translateY(-2px);  
 position: relative;  
}  
  
/\* Add check mark icon \*/  
.sport-checkbox:checked + .sport-circle::after {  
 content: "✓";  
 position: absolute;  
 top: 1px;  
 right: 25px;  
 font-size: 16px;  
 font-weight: bold;  
 color: white;  
}

————————————————————————————————————————

## global.css

/\*\*  
 \* Global CSS Styles  
 \*   
 \* This file contains global CSS variables, reset styles, and base styling  
 \* for the BOOK-PLAY application. It defines the design system including  
 \* colors, spacing, typography, and common utilities.  
 \*   
 \* @author BOOK-PLAY Development Team  
 \* @version 1.0  
 \* @since 2025  
 \*/  
  
/\* ===== CSS VARIABLES (Design System) ===== \*/  
:root {  
 /\* Primary brand colors \*/  
 --primary: #1e90ff; /\* Main blue color for primary actions \*/  
 --secondary: #22c55e; /\* Green color for secondary actions \*/  
   
 /\* Background colors \*/  
 --bg-light: #e0f7fa; /\* Light blue background \*/  
 --bg-secondary: #d0f8ce; /\* Light green background \*/  
   
 /\* Card and component colors \*/  
 --card-bg: rgba(255, 255, 255, 0.95); /\* Semi-transparent white for cards \*/  
 --card-border: rgba(209, 213, 219, 0.6); /\* Subtle border color \*/  
   
 /\* Text colors \*/  
 --text-main: #0f172a; /\* Primary text color \*/  
 --text-muted: #64748b; /\* Secondary/muted text \*/  
 --text-light: #94a3b8; /\* Light text for captions \*/  
   
 /\* Shadow effects \*/  
 --shadow-light: rgba(30, 144, 255, 0.08); /\* Light blue shadow \*/  
 --shadow-medium: rgba(0, 0, 0, 0.08); /\* Medium shadow \*/  
 --shadow-strong: rgba(0, 0, 0, 0.15); /\* Strong shadow \*/  
   
 /\* Border radius values \*/  
 --radius: 12px; /\* Default border radius \*/  
 --radius-medium: 10px; /\* Medium border radius \*/  
 --radius-small: 8px; /\* Small border radius \*/  
   
 /\* Animation and transitions \*/  
 --transition: all 0.3s cubic-bezier(0.4, 0, 0.2, 1); /\* Smooth transition \*/  
 --border-subtle: rgba(0, 0, 0, 0.06); /\* Subtle border \*/  
   
 /\* Layout measurements \*/  
 --navbar-height: 80px; /\* Fixed navbar height \*/  
}  
  
/\* ===== CSS RESET ===== \*/  
\* {  
 margin: 0; /\* Remove default margins \*/  
 padding: 0; /\* Remove default padding \*/  
 box-sizing: border-box; /\* Include padding and border in element width \*/  
}  
  
/\* ===== BASE BODY STYLES ===== \*/  
body {  
 font-family: 'Inter', sans-serif; /\* Modern sans-serif font \*/  
 background: linear-gradient(135deg, var(--bg-light), var(--bg-secondary)); /\* Gradient background \*/  
 color: var(--text-main); /\* Primary text color \*/  
 min-height: 100vh; /\* Full viewport height \*/  
 line-height: 1.6; /\* Comfortable line spacing \*/  
 -webkit-font-smoothing: antialiased; /\* Smooth font rendering \*/  
 padding-top: var(--navbar-height); /\* Account for fixed navbar \*/  
}

————————————————————————————————————————

## layouts.css

/\* layouts.css \*/  
  
/\* ===================== COMMON LAYOUT PATTERNS ===================== \*/  
  
/\* Header Layout \*/  
.header-bar {  
 width: 100%;  
 padding: 20px 5%;  
 display: flex;  
 justify-content: space-between;  
 align-items: center;  
 flex-wrap: wrap;  
 gap: 15px;  
 box-sizing: border-box;  
}  
  
/\* Logo Layout \*/  
.logo-title {  
 font-size: 32px;  
 font-weight: bold;  
 background: linear-gradient(to right, #1e90ff, #22c55e);  
 -webkit-background-clip: text;  
 background-clip: text;  
 -webkit-text-fill-color: transparent;  
 white-space: nowrap;  
}  
  
.logo-title span {  
 color: #f472b6;  
}  
  
/\* Profile Container Layout \*/  
.profile-container {  
 position: relative;  
 cursor: pointer;  
 z-index: 100;  
}  
  
/\* Regular Navbar Layout \*/  
.fresh-navbar {  
 display: flex;  
 align-items: center;  
}  
  
.fresh-navbar .nav-wrapper {  
 display: flex;  
 align-items: center;  
}  
  
.fresh-navbar .nav-links {  
 display: flex;  
 list-style: none;  
 margin: 0;  
 padding: 0;  
 gap: 20px;  
 align-items: center;  
}  
  
.fresh-navbar .nav-links li {  
 margin: 0;  
}  
  
.fresh-navbar .nav-links a {  
 text-decoration: none;  
 color: var(--text-main);  
 font-size: 14px;  
 font-weight: 500;  
 padding: 8px 16px;  
 border-radius: 6px;  
 transition: all 0.3s ease;  
 white-space: nowrap;  
}  
  
.fresh-navbar .nav-links a:hover {  
 background-color: rgba(30, 144, 255, 0.1);  
 color: var(--primary);  
}  
  
.fresh-navbar .nav-links a.active {  
 background-color: var(--primary);  
 color: white;  
}  
  
.fresh-navbar .nav-links a.logout {  
 background-color: #f43f5e;  
 color: white;  
}  
  
.fresh-navbar .nav-links a.logout:hover {  
 background-color: #e11d48;  
}  
  
/\* Admin Navbar Layout \*/  
.admin-navbar {  
 display: flex;  
 align-items: center;  
}  
  
.admin-navbar .nav-links {  
 display: flex;  
 list-style: none;  
 margin: 0;  
 padding: 0;  
 gap: 20px;  
 align-items: center;  
}  
  
.admin-navbar .nav-links li {  
 margin: 0;  
}  
  
.admin-navbar .nav-links a {  
 text-decoration: none;  
 color: var(--text-main);  
 font-size: 14px;  
 font-weight: 500;  
 padding: 8px 16px;  
 border-radius: 6px;  
 transition: all 0.3s ease;  
 white-space: nowrap;  
}  
  
.admin-navbar .nav-links a:hover {  
 background-color: rgba(30, 144, 255, 0.1);  
 color: var(--primary);  
}  
  
.admin-navbar .nav-links a.active {  
 background-color: var(--primary);  
 color: white;  
}  
  
.admin-navbar .nav-links a.logout {  
 background-color: #f43f5e;  
 color: white;  
}  
  
.admin-navbar .nav-links a.logout:hover {  
 background-color: #e11d48;  
}  
  
.profile-icon {  
 width: 60px;  
 height: 60px;  
 border-radius: 50%;  
 object-fit: cover;  
 box-shadow: 0 0 6px var(--accent-glow);  
}  
  
.profile-menu {  
 position: fixed;  
 top: 70px;  
 right: 30px;  
 background-color: white;  
 border: 1px solid var(--card-border);  
 border-radius: var(--radius);  
 box-shadow: 0 4px 16px rgba(0, 0, 0, 0.06);  
 overflow: hidden;  
 display: none;  
 flex-direction: column;  
 min-width: 180px;  
 z-index: 101;  
 transition: opacity 0.3s ease, transform 0.3s ease;  
}  
  
.profile-menu a {  
 padding: 12px 18px;  
 text-decoration: none;  
 color: var(--text-main);  
 font-size: 14px;  
}  
  
.profile-menu a:hover {  
 background-color: #f3f4f6;  
}  
  
.profile-menu.active {  
 display: flex;  
}  
  
/\* Button Container Layouts \*/  
.main-buttons {  
 display: flex;  
 gap: 30px;  
 margin-top: 60px;  
 flex-wrap: wrap;  
 justify-content: center;  
 width: 90%;  
 max-width: 800px;  
}  
  
.secondary-buttons {  
 display: grid;  
 grid-template-columns: repeat(auto-fit, minmax(140px, 1fr));  
 gap: 30px;  
 margin: 60px auto;  
 width: 90%;  
 max-width: 800px;  
}  
  
.admin-buttons {  
 display: flex;  
 gap: 30px;  
 margin-top: 60px;  
 flex-wrap: wrap;  
 justify-content: center;  
 width: 90%;  
 max-width: 800px;  
}  
  
/\* Card Layout \*/  
.glow-card {  
 background: var(--card-bg);  
 border: 1px solid var(--card-border);  
 border-radius: var(--radius);  
 padding: 20px;  
 text-align: center;  
 color: var(--text-main);  
 box-shadow: 0 4px 12px rgba(30, 144, 255, 0.05);  
 transition: all 0.3s ease;  
 cursor: pointer;  
 width: 160px;  
 height: 160px;  
 display: flex;  
 flex-direction: column;  
 align-items: center;  
 justify-content: center;  
 gap: 10px;  
}  
  
.main-buttons .glow-card {  
 width: 200px;  
 height: 200px;  
}  
  
.glow-card img.icon {  
 width: 80px;  
 height: auto;  
 margin-bottom: 10px;  
}  
  
.glow-card span {  
 font-size: 16px;  
 font-weight: 500;  
 text-align: center;  
}  
  
.glow-card:hover {  
 transform: translateY(-5px);  
 box-shadow: 0 8px 25px rgba(30, 144, 255, 0.15);  
}  
  
/\* Footer Layout \*/  
footer {  
 width: 100%;  
 background: rgba(255, 255, 255, 0.8);  
 backdrop-filter: blur(10px);  
 border-top: 1px solid var(--card-border);  
 padding: 20px;  
 text-align: center;  
 margin-top: auto;  
}  
  
.footer-content {  
 color: var(--text-muted);  
 font-size: 14px;  
}  
  
/\* Container Layout \*/  
.container {  
 z-index: 2;  
 background: rgba(255, 255, 255, 0.8);  
 padding: 30px 25px;  
 border-radius: 20px;  
 box-shadow: 0 0 30px rgba(30, 144, 255, 0.15);  
 width: 90%;  
 max-width: 420px;  
 backdrop-filter: blur(15px);  
 -webkit-backdrop-filter: blur(15px);  
 border: 1px solid #d1d5db;  
 text-align: center;  
 color: #1e293b;  
 margin: 20px auto;  
}  
  
/\* Form Layout \*/  
form {  
 width: 100%;  
}  
  
form input {  
 width: 100%;  
 padding: 14px;  
 margin: 12px 0;  
 border: 1px solid #cbd5e1;  
 border-radius: 10px;  
 background: #f8fafc;  
 color: #1e293b;  
 font-size: 15px;  
 transition: border 0.3s ease;  
 box-sizing: border-box;  
}  
  
form input:focus {  
 border-color: #1e90ff;  
 outline: none;  
 box-shadow: 0 0 5px rgba(30, 144, 255, 0.2);  
}  
  
/\* Button Layout \*/  
button {  
 width: 100%;  
 padding: 14px;  
 margin-top: 15px;  
 border: none;  
 border-radius: 10px;  
 background: linear-gradient(to right, #1e90ff, #22c55e);  
 color: white;  
 font-weight: bold;  
 font-size: 16px;  
 cursor: pointer;  
 transition: transform 0.2s ease, box-shadow 0.3s ease;  
 box-shadow: 0 0 10px rgba(30, 144, 255, 0.2);  
}  
  
button:hover {  
 transform: scale(1.03);  
 box-shadow: 0 0 15px rgba(30, 144, 255, 0.3);  
}  
  
button:active {  
 transform: scale(0.98);  
}  
  
/\* Logo Layout (for forms) \*/  
.logo {  
 font-size: 28px;  
 font-weight: bold;  
 background: linear-gradient(to right, #1e90ff, #22c55e);  
 -webkit-background-clip: text;  
 background-clip: text;  
 -webkit-text-fill-color: transparent;  
 margin-bottom: 16px;  
 line-height: 1.2;  
}  
  
/\* Slogan Layout \*/  
.slogan {  
 margin-top: 20px;  
 font-size: 13px;  
 color: #64748b;  
}  
  
.slogan a {  
 color: #1e90ff;  
 text-decoration: none;  
}  
  
.slogan a:hover {  
 text-decoration: underline;  
}  
  
/\* Background Icons Layout \*/  
.bg-icons {  
 position: absolute;  
 top: 0;  
 left: 0;  
 width: 100%;  
 height: 100%;  
 overflow: hidden;  
 z-index: 1;  
 pointer-events: none;  
}  
  
.bg-icons img {  
 position: absolute;  
 width: 15%;  
 max-width: 260px;  
 opacity: 0.15;  
 pointer-events: none;  
 animation: float 12s ease-in-out infinite;  
}  
  
/\* Icon Positions \*/  
.icon1 { top: 5%; left: 2%; filter: drop-shadow(0 0 20px #38bdf8); }  
.icon2 { top: 35%; left: 2%; filter: drop-shadow(0 0 20px #22c55e); }  
.icon3 { bottom: 5%; left: 2%; filter: drop-shadow(0 0 20px #94a3b8); }  
.icon4 { top: 5%; right: 2%; filter: drop-shadow(0 0 20px #60a5fa); }  
.icon5 { top: 35%; right: 2%; filter: drop-shadow(0 0 20px #34d399); }  
.icon6 { bottom: 5%; right: 2%; filter: drop-shadow(0 0 20px #94a3b8); }  
  
/\* Floating Animation \*/  
@keyframes float {  
 0%, 100% { transform: translateY(0); }  
 50% { transform: translateY(-8px); }  
}  
  
/\* Floating Admin Button \*/  
.floating-admin-btn {  
 position: fixed;  
 bottom: 20px;  
 right: 20px;  
 background: linear-gradient(135deg, #facc15, #fef08a);  
 color: #1e293b;  
 border: none;  
 border-radius: 12px;  
 padding: 12px 20px;  
 font-size: 14px;  
 font-weight: 600;  
 box-shadow: 0 4px 12px rgba(250, 204, 21, 0.4);  
 cursor: pointer;  
 transition: all 0.3s ease;  
 z-index: 999;  
}  
  
.floating-admin-btn:hover {  
 background: linear-gradient(135deg, #eab308, #fde047);  
 box-shadow: 0 6px 16px rgba(250, 204, 21, 0.6);  
}  
  
/\* Username Display \*/  
.username-display {  
 margin-right: 10px;  
 font-weight: 500;  
 color: var(--text-main);  
}  
  
/\* Fade In Animation \*/  
@keyframes fadeInUp {  
 from {  
 opacity: 0;  
 transform: translateY(30px);  
 }  
 to {  
 opacity: 1;  
 transform: translateY(0);  
 }  
}  
  
.main-buttons,  
.secondary-buttons,  
.admin-buttons {  
 animation: fadeInUp 0.6s ease-out;  
}

————————————————————————————————————————

## responsive.css

/\* responsive.css \*/  
  
/\* ===================== COMMON RESPONSIVE BREAKPOINTS ===================== \*/  
  
/\* Large devices (laptops/desktops, up to 1200px) \*/  
@media screen and (max-width: 1200px) {  
 /\* Common grid adjustments \*/  
 .secondary-buttons,  
 .admin-buttons {  
 grid-template-columns: repeat(auto-fit, minmax(130px, 1fr));  
 gap: 25px;  
 }  
   
 /\* Button container adjustments \*/  
 .main-buttons {  
 gap: 25px;  
 }  
}  
  
/\* Medium devices (tablets, up to 992px) \*/  
@media screen and (max-width: 992px) {  
 /\* Logo adjustments \*/  
 .logo-title {  
 font-size: 28px;  
 }  
   
 /\* Button adjustments \*/  
 .main-buttons {  
 gap: 20px;  
 }  
   
 .main-buttons .glow-card,  
 .glow-card {  
 width: 180px;  
 height: 180px;  
 }  
   
 .glow-card img.icon {  
 width: 70px;  
 }  
   
 .secondary-buttons,  
 .admin-buttons {  
 gap: 20px;  
 }  
}  
  
/\* Small devices (landscape phones, up to 768px) \*/  
@media screen and (max-width: 768px) {  
 /\* Header adjustments \*/  
 .header-bar {  
 padding: 15px 20px;  
 }  
   
 .logo-title {  
 font-size: 26px;  
 }  
   
 /\* Navbar adjustments \*/  
 .fresh-navbar .nav-links,  
 .admin-navbar .nav-links {  
 gap: 10px;  
 }  
   
 .fresh-navbar .nav-links a,  
 .admin-navbar .nav-links a {  
 font-size: 12px;  
 padding: 6px 12px;  
 }  
   
 /\* Button container adjustments \*/  
 .main-buttons {  
 margin-top: 40px;  
 gap: 15px;  
 }  
   
 .main-buttons .glow-card,  
 .glow-card {  
 width: 160px;  
 height: 160px;  
 }  
   
 .secondary-buttons,  
 .admin-buttons {  
 grid-template-columns: repeat(auto-fit, minmax(120px, 1fr));  
 margin: 40px auto;  
 gap: 15px;  
 }  
   
 .glow-card {  
 width: 140px;  
 height: 140px;  
 padding: 15px;  
 }  
   
 .glow-card img.icon {  
 width: 60px;  
 margin-bottom: 8px;  
 }  
   
 .glow-card span {  
 font-size: 14px;  
 }  
   
 /\* Profile menu adjustments \*/  
 .profile-menu {  
 min-width: 160px;  
 right: -10px;  
 }  
}  
  
/\* Extra small devices (phones, up to 576px) \*/  
@media screen and (max-width: 576px) {  
 /\* Navbar adjustments \*/  
 .fresh-navbar .nav-links,  
 .admin-navbar .nav-links {  
 gap: 8px;  
 }  
   
 .fresh-navbar .nav-links a,  
 .admin-navbar .nav-links a {  
 font-size: 11px;  
 padding: 5px 8px;  
 }  
   
 /\* Hide some nav items on very small screens \*/  
 .fresh-navbar .nav-links li:nth-child(3),  
 .fresh-navbar .nav-links li:nth-child(4),  
 .fresh-navbar .nav-links li:nth-child(5),  
 .admin-navbar .nav-links li:nth-child(3),  
 .admin-navbar .nav-links li:nth-child(4) {  
 display: none;  
 }  
}  
@media screen and (max-width: 576px) {  
 /\* Header adjustments \*/  
 .header-bar {  
 padding: 15px;  
 justify-content: center;  
 }  
   
 .logo-title {  
 font-size: 24px;  
 text-align: center;  
 }  
   
 .profile-container {  
 position: absolute;  
 top: 15px;  
 right: 15px;  
 }  
   
 /\* Button container adjustments \*/  
 .main-buttons {  
 margin-top: 30px;  
 flex-direction: column;  
 align-items: center;  
 gap: 15px;  
 }  
   
 .main-buttons .glow-card {  
 width: 180px;  
 height: 180px;  
 }  
   
 .secondary-buttons,  
 .admin-buttons {  
 grid-template-columns: repeat(2, 1fr);  
 gap: 15px;  
 margin: 30px auto;  
 }  
   
 .glow-card {  
 width: 100%;  
 height: 130px;  
 padding: 10px;  
 }  
   
 .glow-card img.icon {  
 width: 50px;  
 margin-bottom: 5px;  
 }  
   
 .glow-card span {  
 font-size: 13px;  
 }  
   
 /\* Footer adjustments \*/  
 footer {  
 padding: 15px;  
 font-size: 12px;  
 }  
}  
  
/\* Very small devices (small phones, up to 380px) \*/  
@media screen and (max-width: 380px) {  
 .logo-title {  
 font-size: 22px;  
 }  
   
 .main-buttons .glow-card {  
 width: 160px;  
 height: 160px;  
 }  
   
 .secondary-buttons,  
 .admin-buttons {  
 grid-template-columns: 1fr;  
 gap: 10px;  
 }  
}  
  
/\* Extra very small devices (tiny phones, up to 350px) \*/  
@media screen and (max-width: 350px) {  
 .logo-title {  
 font-size: 20px;  
 }  
   
 .main-buttons .glow-card {  
 width: 140px;  
 height: 140px;  
 }  
   
 .glow-card {  
 width: 100%;  
 height: 120px;  
 padding: 8px;  
 }  
   
 .glow-card img.icon {  
 width: 40px;  
 margin-bottom: 4px;  
 }  
   
 .glow-card span {  
 font-size: 12px;  
 }  
   
 .header-bar {  
 padding: 10px;  
 }  
   
 .profile-container {  
 top: 10px;  
 right: 10px;  
 }  
   
 .profile-icon {  
 width: 50px;  
 height: 50px;  
 }  
}  
  
/\* ===================== HEIGHT-BASED MEDIA QUERIES ===================== \*/  
  
/\* Short screens (height-based) \*/  
@media screen and (max-height: 600px) {  
 body {  
 align-items: flex-start;  
 padding: 20px 10px;  
 }  
   
 .container {  
 margin-top: 10px;  
 }  
}  
  
@media screen and (max-height: 500px) and (orientation: landscape) {  
 body {  
 padding: 10px;  
 }  
   
 .container {  
 padding: 15px;  
 margin: 10px auto;  
 max-width: 460px;  
 }  
}  
  
/\* ===================== FORM-SPECIFIC RESPONSIVE ===================== \*/  
  
/\* Form responsive adjustments \*/  
@media screen and (max-width: 768px) {  
 .container {  
 padding: 25px 20px;  
 }  
   
 .logo {  
 font-size: 24px;  
 }  
   
 form input,  
 button {  
 padding: 12px;  
 }  
}  
  
@media screen and (max-width: 480px) {  
 .container {  
 padding: 20px 15px;  
 width: 95%;  
 margin: 15px auto;  
 }  
   
 .logo {  
 font-size: 22px;  
 }  
   
 form input,  
 button {  
 padding: 10px;  
 font-size: 14px;  
 }  
   
 .slogan {  
 font-size: 12px;  
 }  
}  
  
@media screen and (max-width: 350px) {  
 .container {  
 padding: 15px 10px;  
 width: 98%;  
 margin: 10px auto;  
 }  
   
 .logo {  
 font-size: 20px;  
 margin-bottom: 10px;  
 }  
   
 form input,  
 button {  
 padding: 8px;  
 font-size: 13px;  
 }  
   
 .slogan {  
 font-size: 11px;  
 }  
}  
  
@media screen and (max-width: 320px) {  
 .container {  
 padding: 15px 10px;  
 border-radius: 15px;  
 margin: 10px auto;  
 }  
   
 .logo {  
 font-size: 20px;  
 margin-bottom: 12px;  
 }  
   
 form input {  
 margin: 8px 0;  
 }  
}  
  
/\* ===================== BACKGROUND ICONS RESPONSIVE ===================== \*/  
  
@media screen and (max-width: 768px) {  
 .bg-icons img {  
 width: 20%;  
 }  
}  
  
@media screen and (max-width: 480px) {  
 .bg-icons img {  
 width: 25%;  
 }  
   
 .icon2,  
 .icon5 {  
 display: none; /\* Hide some icons on small screen \*/  
 }  
}  
  
@media screen and (max-width: 350px) {  
 .bg-icons img {  
 width: 30%;  
 }  
   
 .icon1,  
 .icon3,  
 .icon4,  
 .icon6 {  
 opacity: 0.08; /\* Further reduce visibility \*/  
 }  
}  
  
@media screen and (max-width: 320px) {  
 .icon1,  
 .icon3,  
 .icon4,  
 .icon6 {  
 opacity: 0.1; /\* Reduce visibility \*/  
 }  
}  
  
@media screen and (max-height: 500px) and (orientation: landscape) {  
 .bg-icons img {  
 opacity: 0.08; /\* Lower visibility \*/  
 }  
   
 .icon2,  
 .icon5 {  
 display: none;  
 }  
}

————————————————————————————————————————

## shared.js

/\*\*  
 \* Shared JavaScript Functions  
 \*   
 \* This file contains common JavaScript functions used across the BOOK-PLAY application.  
 \* It includes utility functions for UI interactions, user profile management,  
 \* and admin request functionality.  
 \*   
 \* @author BOOK-PLAY Development Team  
 \* @version 1.0  
 \* @since 2025  
 \*/  
  
/\*\*  
 \* Toggle profile menu visibility  
 \* Shows/hides the user profile dropdown menu  
 \*/  
function toggleProfileMenu() {  
 const menu = document.getElementById('profileMenu');  
 menu.classList.toggle('active');  
}  
  
/\*\*  
 \* Close profile menu when clicking outside  
 \* Automatically closes the profile dropdown when user clicks elsewhere  
 \*/  
document.addEventListener('click', function (e) {  
 const profileContainer = document.querySelector('.profile-container');  
 const menu = document.getElementById('profileMenu');  
 if (!profileContainer.contains(e.target)) {  
 menu.classList.remove('active');  
 }  
});  
  
/\*\*  
 \* Show temporary message notification  
 \* Creates and displays a temporary toast message on screen  
 \*   
 \* @param {string} message - The message text to display  
 \* @param {number} duration - Duration in milliseconds to show the message  
 \*/  
function showTempMessage(message, duration) {  
 const msg = document.createElement("div");  
 msg.textContent = message;  
 msg.style.position = "fixed";  
 msg.style.bottom = "80px";  
 msg.style.right = "20px";  
 msg.style.background = "#1e90ff";  
 msg.style.color = "white";  
 msg.style.padding = "12px 20px";  
 msg.style.borderRadius = "10px";  
 msg.style.boxShadow = "0 4px 12px rgba(0,0,0,0.2)";  
 msg.style.zIndex = 1000;  
 msg.style.fontSize = "14px";  
 document.body.appendChild(msg);  
 setTimeout(() => msg.remove(), duration);  
}  
  
/\*\*  
 \* Load user profile image and username  
 \* Fetches and displays the current user's profile image and username  
 \* from the server and updates the UI elements  
 \*/  
function loadUserProfile() {  
 fetch("HomePage.php?action=get\_user\_image")  
 .then((res) => res.json())  
 .then((data) => {  
 const imgElement = document.getElementById("userProfileImage");  
 const usernameSpan = document.getElementById("usernameDisplay");  
   
 // Update profile image if available  
 if (imgElement && data.image) {  
 imgElement.src = `../../../uploads/users/${data.image}`;  
 }  
   
 // Update username display if available  
 if (usernameSpan && data.username) {  
 usernameSpan.textContent = data.username;  
 }  
 })  
 .catch((err) => {  
 console.error("Failed to load user image or username", err);  
 });  
}  
  
/\*\*  
 \* Initialize admin request button functionality  
 \* Sets up the admin request button with proper event handling and status checking  
 \*/  
function initAdminRequestButton() {  
 const adminBtn = document.getElementById("adminRequestBtn");  
 if (!adminBtn) return; // Exit if the button doesn't exist  
  
 // Check admin status and if a request has already been submitted  
 fetch("../../admin\_actions.php?action=check")  
 .then((res) => res.json())  
 .then((data) => {  
 // Hide the button if the user is already an admin  
 if (data.is\_admin) {  
 adminBtn.style.display = "none";  
 return;  
 }  
  
 // Handle button click event  
 adminBtn.addEventListener("click", () => {  
 if (data.already\_requested) {  
 // Show waiting message if request already submitted  
 showTempMessage(  
 "🕐 Your admin request is under review. Thank you for your patience.",  
 4000  
 );  
 } else {  
 // Send new admin request  
 fetch("../../admin\_actions.php", {  
 method: "POST",  
 headers: {  
 "Content-Type": "application/x-www-form-urlencoded",  
 },  
 body: "action=submit",  
 })  
 .then((res) => res.json())  
 .then((response) => {  
 if (response.success) {  
 showTempMessage(  
 "✅ Your admin request has been submitted. We'll review it as soon as possible.",  
 4000  
 );  
 data.already\_requested = true; // Prevent re-submission  
 } else {  
 showTempMessage(  
 "❌ " + (response.message || "Failed to submit request."),  
 4000  
 );  
 }  
 })  
 .catch((error) => {  
 console.error("Error submitting request:", error);  
 showTempMessage(  
 "❌ Could not connect to server. Try again later.",  
 4000  
 );  
 });  
 }  
 });  
 })  
 .catch((error) => {  
 console.error("Error checking admin status:", error);  
 showTempMessage(  
 "❌ Could not check admin status. Try again later.",  
 4000  
 );  
 });  
}  
  
/\*\*  
 \* Initialize shared functionality when DOM is loaded  
 \* Sets up common features like user profile loading and admin request button  
 \*/  
document.addEventListener("DOMContentLoaded", () => {  
 loadUserProfile(); // Load and display user profile information  
 initAdminRequestButton(); // Initialize admin request functionality  
});

————————————————————————————————————————

## footer.php

<?php  
/\*\*  
 \* Footer Component  
 \*   
 \* This component provides a consistent footer across all pages with:  
 \* - Copyright information  
 \* - Application branding  
 \*   
 \* Usage: Simply include this file in any page  
 \*   
 \* @author BOOK-PLAY Development Team  
 \* @version 1.0  
 \* @since 2025  
 \*/  
?>  
<!-- Application footer with copyright information -->  
<footer>  
 <div class="footer-content">  
 <p>&copy; 2025 Book&Play. All rights reserved.</p>  
 </div>  
</footer>

————————————————————————————————————————

## header-bar.php

<?php  
/\*\*  
 \* Header Bar Component  
 \*   
 \* This component provides a consistent header bar across all pages with:  
 \* - Logo/title display  
 \* - Navigation menu (if navbar.php is included)  
 \* - User profile section with dropdown menu  
 \*   
 \* Required variables (optional):  
 \* - $profileImagePath: Custom path for user profile image  
 \* - $editProfilePath: Custom path for edit profile link  
 \* - $logoutPath: Custom path for logout link  
 \* - $isAdmin: Set to true for admin pages to show admin navbar  
 \*   
 \* @author BOOK-PLAY Development Team  
 \* @version 1.0  
 \* @since 2025  
 \*/  
?>  
<!-- Main header container with logo and navigation -->  
<div class="header-bar">  
 <!-- Application logo/title -->  
 <div class="logo-title">BOOK&PLAY</div>  
   
 <!-- Include navigation menu component -->  
 <?php include 'navbar.php'; ?>  
   
 <!-- User profile section with dropdown menu -->  
 <div class="profile-container" onclick="toggleProfileMenu()">  
 <!-- Username display (populated by JavaScript) -->  
 <span id="usernameDisplay" class="username-display"></span>  
   
 <!-- User profile image with fallback to default -->  
 <img  
 id="userProfileImage"  
 class="profile-icon"  
 src="<?php echo isset($profileImagePath) ? $profileImagePath : '../../../uploads/users/default.jpg'; ?>"  
 alt="Profile Image"  
 />  
   
 <!-- Profile dropdown menu -->  
 <div class="profile-menu" id="profileMenu">  
 <a href="<?php echo isset($editProfilePath) ? $editProfilePath : '../../../auth/EditProfile/EditProfile.php'; ?>">Edit Profile</a>  
 <a href="<?php echo isset($logoutPath) ? $logoutPath : '../../../logout.php'; ?>">Logout</a>  
 </div>  
 </div>  
</div>

————————————————————————————————————————

## sports-scroll.php

<?php  
/\*\*  
 \* Sports Scroll Component  
 \*   
 \* This component creates a horizontal scrollable list of sports with checkboxes  
 \* for filtering venues. It dynamically fetches sports from the database and  
 \* creates interactive filter controls.  
 \*   
 \* Features:  
 \* - Dynamic sports loading from database  
 \* - Checkbox-based filtering  
 \* - Horizontal scrollable layout  
 \* - Real-time venue filtering  
 \*   
 \* @author BOOK-PLAY Development Team  
 \* @version 1.0  
 \* @since 2025  
 \*/  
?>  
  
<!-- Sports filter container -->  
<section class="sports-scroll" id="sportsContainer"></section>  
  
<script>  
/\*\*  
 \* Initialize sports filter component  
 \* Fetches available sports and creates interactive filter controls  
 \*/  
document.addEventListener("DOMContentLoaded", () => {  
 // Fetch sports data from server  
 fetch("../BookVenue/fetch\_sports.php")  
 .then(res => res.json())  
 .then(data => {  
 if (data.success) {  
 const container = document.getElementById("sportsContainer");  
 container.innerHTML = ""; // Clear existing content  
  
 // Create filter controls for each sport  
 data.sports.forEach(sport => {  
 // Create label container for each sport item  
 const label = document.createElement("label");  
 label.className = "sport-item";  
  
 // Create checkbox input for sport selection  
 const input = document.createElement("input");  
 input.type = "checkbox";  
 input.className = "sport-checkbox";  
  
 // Create visual circle element for sport name  
 const circle = document.createElement("div");  
 circle.className = "sport-circle";  
 circle.textContent = sport;  
 circle.dataset.sport = sport; // Store sport name as data attribute  
  
 // Add change event listener for real-time filtering  
 input.addEventListener("change", () => {  
 // Get all checked sports  
 const checked = document.querySelectorAll(".sport-checkbox:checked");  
 const selectedSports = Array.from(checked).map(cb => cb.nextElementSibling.dataset.sport);  
   
 // Apply filter to venues  
 filterVenuesBySports(selectedSports);  
 });  
  
 // Assemble sport filter item  
 label.appendChild(input);  
 label.appendChild(circle);  
 container.appendChild(label);  
 });  
 }  
 })  
 .catch(err => console.error("Failed to fetch sports", err));  
});  
</script>

————————————————————————————————————————

## MailLink.php

<?php  
require 'vendor/autoload.php';  
require\_once \_\_DIR\_\_ . '/../db.php';  
  
use PHPMailer\PHPMailer\PHPMailer;  
use PHPMailer\PHPMailer\Exception;  
  
/\*\*  
 \* Email Verification System  
 \*   
 \* This function sends a verification code to a specified email address.  
 \* Used for account verification, password resets, and security confirmations.  
 \*  
 \* @param string $to Recipient's email address  
 \* @param string $code Verification code to be sent  
 \* @return bool True if email sent successfully, false otherwise  
 \*/  
function sendVerificationCode($to, $code) {  
 // Initialize PHPMailer with exception handling enabled  
 $mail = new PHPMailer(true);  
   
 try {  
 // Configure SMTP settings  
 $mail->isSMTP(); // Use SMTP protocol  
 $mail->Host = 'smtp.gmail.com'; // Gmail SMTP server  
 $mail->SMTPAuth = true; // Enable SMTP authentication  
 $mail->Username = 'bookandplay.team@gmail.com'; // SMTP username  
 $mail->Password = 'zfdi dqpm tyug tfvh'; // App Password (not regular password)  
 $mail->SMTPSecure = PHPMailer::ENCRYPTION\_STARTTLS; // Enable TLS encryption  
 $mail->Port = 587; // TCP port for TLS connection  
  
 // Set email sender and recipient  
 $mail->setFrom('bookandplay.team@gmail.com', 'Book And Play');  
 $mail->addAddress($to); // Add recipient  
  
 // Set email content  
 $mail->isHTML(true); // Set email format to HTML  
 $mail->Subject = 'Your Verification Code'; // Email subject  
 $mail->Body = "<h3>Your verification code is: <b>$code</b></h3>"; // HTML message  
  
 // Send email and return success  
 $mail->send();  
 return true;  
  
 } catch (Exception $e) {  
 // Log error and return failure  
 error\_log("Error sending verification email: {$mail->ErrorInfo}");  
 return false;  
 }  
}  
?>

————————————————————————————————————————

## admin\_actions.php

<?php  
/\*\*  
 \* Admin Access Request Handler  
 \*   
 \* This file handles admin access requests from users. It provides two main actions:  
 \* 1. Check current admin status and existing requests  
 \* 2. Submit new admin access requests  
 \*   
 \* @author BOOK-PLAY Development Team  
 \* @version 1.0  
 \* @since 2025  
 \*/  
  
// Start session and include database connection  
session\_start();  
require\_once '../../../db.php';  
  
// Get current user from session  
$username = $\_SESSION['user\_id'] ?? '';  
  
// Check if user is logged in  
if (!$username) {  
 echo json\_encode(["success" => false, "message" => "Not logged in"]);  
 exit;  
}  
  
// Get action from POST or GET request  
$action = $\_POST['action'] ?? $\_GET['action'] ?? '';  
  
/\*\*  
 \* Check admin status and existing requests  
 \* Returns current admin status and whether user has already requested admin access  
 \*/  
if ($action === 'check') {  
 // Check if user is already an admin  
 $stmt = $conn->prepare("SELECT is\_admin FROM users WHERE username = ?");  
 $stmt->bind\_param("s", $username);  
 $stmt->execute();  
 $stmt->bind\_result($isAdmin);  
 $stmt->fetch();  
 $stmt->close();  
  
 // Check if user has already submitted an admin request  
 $stmt = $conn->prepare("SELECT COUNT(\*) FROM reports WHERE username = ? AND type = 'admin\_request'");  
 $stmt->bind\_param("s", $username);  
 $stmt->execute();  
 $stmt->bind\_result($count);  
 $stmt->fetch();  
 $stmt->close();  
  
 // Return admin status and request status  
 echo json\_encode([  
 "is\_admin" => $isAdmin == 1,  
 "already\_requested" => $count > 0  
 ]);  
 exit;  
}  
  
/\*\*  
 \* Submit new admin access request  
 \* Creates a new admin request if user hasn't already submitted one  
 \*/  
if ($action === 'submit') {  
 // Check if user has already submitted a request  
 $stmt = $conn->prepare("SELECT COUNT(\*) FROM reports WHERE username = ? AND type = 'admin\_request'");  
 $stmt->bind\_param("s", $username);  
 $stmt->execute();  
 $stmt->bind\_result($count);  
 $stmt->fetch();  
 $stmt->close();  
  
 // Prevent duplicate requests  
 if ($count > 0) {  
 echo json\_encode(["success" => false, "message" => "Request already exists"]);  
 exit;  
 }  
  
 // Insert new admin request  
 $stmt = $conn->prepare("INSERT INTO reports (username, type, message) VALUES (?, 'admin\_request', 'Requesting admin access')");  
 $stmt->bind\_param("s", $username);  
 $success = $stmt->execute();  
 $stmt->close();  
  
 echo json\_encode(["success" => $success]);  
 exit;  
}  
  
// Return error for invalid actions  
echo json\_encode(["success" => false, "message" => "Invalid action"]);  
?>

————————————————————————————————————————

## index.php

<?php  
/\*\*  
 \* BOOK-PLAY Application Entry Point  
 \*   
 \* This file serves as the main entry point for the BOOK-PLAY application.  
 \* It automatically redirects users to the login page when they access the root URL.  
 \*   
 \* @author BOOK-PLAY Development Team  
 \* @version 1.0  
 \* @since 2025  
 \*/  
  
// Redirect to login page - all users must authenticate first  
header("Location: pages/auth/Login\_Page/Login.php");  
exit();  
?>

————————————————————————————————————————

## logout.php

<?php  
/\*\*  
 \* User Logout Handler  
 \*   
 \* This file handles user logout functionality by destroying the session  
 \* and redirecting users back to the login page.  
 \*   
 \* @author BOOK-PLAY Development Team  
 \* @version 1.0  
 \* @since 2025  
 \*/  
  
// Start session to access session variables  
session\_start();  
  
// (Optional) Log logout time to database or file for audit purposes  
  
// Clear all session variables  
session\_unset();  
  
// Destroy the session completely  
session\_destroy();  
  
// Redirect to login page  
header("Location: pages/auth/Login\_Page/Login.php");  
exit;  
?>

————————————————————————————————————————