

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0,
viewport-fit=cover">
  <title>Sea Catch Scorer</title>
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css"
rel="stylesheet">
  <link href="https://cdn.jsdelivr.net/npm/bootstrap-icons/font/bootstrap-icons.css"
rel="stylesheet">
  <style>
    .splash-screen {
      display: flex;
      flex-direction: column;
      align-items: center;
      justify-content: center;
      height: 100vh;
      background-color: #F5F5F7;
    }
    .splash-screen img {
      max-width: 100%;
      height: auto;
    }
    .splash-screen .btn {
      margin: 10px;
    }
    body {
      overscroll-behavior-y: contain;
    }
    #savedScorecardsList .card {
      border: 1px solid #ddd;
      border-radius: 8px;
      margin-bottom: 10px;
    }
    #savedScorecardsList .card-body {
      padding: 15px;
    }
    #savedScorecardsList .card-title {
      font-size: 1.2rem;
      margin-bottom: 10px;
    }
    #savedScorecardsList .card-text {
      font-size: 1rem;
      margin-bottom: 5px;
    }
    #customSpeciesInput {
      display: none; /* Hidden by default */

```

```

    }
  </style>
</head>
<body>
  <div class="container">
    <!-- Splash Screen -->
    <div id="splashScreen" class="splash-screen">
      
      <button class="btn btn-primary" onclick="startFishing()">Continue to App</button>
      <button class="btn btn-info" data-bs-toggle="modal"
data-bs-target="#savedScorecardsModal">View Saved Scorecards</button>
    </div>
    <!-- Saved Scorecards Modal -->
    <div class="modal fade" id="savedScorecardsModal" tabindex="-1"
aria-labelledby="savedScorecardsModalLabel" aria-hidden="true">
      <div class="modal-dialog modal-lg">
        <div class="modal-content">
          <div class="modal-header">
            <h5 class="modal-title" id="savedScorecardsModalLabel">Saved
Scorecards</h5>
            <button type="button" class="btn-close" data-bs-dismiss="modal"
aria-label="Close"></button>
          </div>
          <div class="modal-body">
            <div id="savedScorecardsList"></div>
          </div>
          <div class="modal-footer">
            <button type="button" class="btn btn-secondary"
data-bs-dismiss="modal">Close</button>
          </div>
        </div>
      </div>
    </div>
    <!-- Angler Registration -->
    <div id="anglerRegistration" class="card-section" style="display: none;">
      <div class="mb-3">
        <label for="anglerName" class="form-label">Angler Name</label>
        <input type="text" id="anglerName" class="form-control" placeholder="Enter your
name">
      </div>
      <div class="mb-3">
        <label for="competitionDate" class="form-label">Date</label>
        <input type="date" id="competitionDate" class="form-control" placeholder="Enter
Date">
      </div>
      <div class="mb-3">
        <label for="venueInput" class="form-label">Venue</label>

```

```

        <input type="text" id="venueInput" class="form-control" placeholder="Enter venue
(e.g., Lake Smith)">
    </div>
    <div class="mb-3 form-check">
        <input type="checkbox" class="form-check-input" id="isCompetition"
onchange="toggleCompetitionFields()">
        <label class="form-check-label" for="isCompetition">Select if a
competition</label>
    </div>
    <div class="mb-3">
        <label for="zoneInput" class="form-label">Zone</label>
        <select id="zoneInput" class="form-select" disabled>
            <option value="" selected>Select a zone</option>
            <option value="A">A</option>
            <option value="B">B</option>
            <option value="C">C</option>
            <option value="D">D</option>
        </select>
    </div>
    <div class="mb-3">
        <label for="pegInput" class="form-label">Peg</label>
        <select id="pegInput" class="form-select" disabled>
            <option value="" selected>Select a peg</option>
            <option value="1">1</option>
            <option value="2">2</option>
            <option value="3">3</option>
            <option value="4">4</option>
            <option value="5">5</option>
            <option value="6">6</option>
            <option value="7">7</option>
            <option value="8">8</option>
        </select>
    </div>
    <div class="d-flex justify-content-between">
        <button class="btn btn-primary" id="startFishingBtn">Record Catch <i class="bi
bi-arrow-right"></i></button>
    </div>
</div>
<!-- Angler Fishing Page -->
<div id="anglerFishingPage" style="display: none;" class="card-section">
    <h2>Angler: <span id="currentAnglerName"></span></h2>
    <p>Venue: <span id="currentVenue"></span>, Zone: <span
id="currentZone"></span>, Peg: <span id="currentPeg"></span>, Date: <span
id="currentDate"></span></p>
    <div class="mb-3">
        <label for="speciesInput" class="form-label">Species</label>
        <select id="speciesInput" class="form-select"
onchange="toggleCustomSpeciesInput()">

```

```

        <option value="" selected>Select a species</option>
        <option value="Other">Other (specify)</option>
    </select>
    <input type="text" id="customSpeciesInput" class="form-control mt-2"
placeholder="Enter custom species">
</div>
<div class="mb-3">
    <label for="lengthInput" class="form-label">Length (cm)</label>
    <input type="number" id="lengthInput" class="form-control" placeholder="Enter
fish length">
</div>
<button class="btn btn-primary mb-3" id="addFishBtn">Add Fish <i class="bi
bi-plus"></i></button>
<button class="btn btn-secondary mb-3" id="backToRegistrationBtn">Back to
Registration</button>
<h2>Fish Log</h2>
<table class="table">
    <thead>
        <tr>
            <th>Species</th>
            <th>Length (cm)</th>
            <th>Action</th>
        </tr>
    </thead>
    <tbody id="fishLog"></tbody>
</table>
<h2>Summary</h2>
<table class="table">
    <thead>
        <tr>
            <th>Species</th>
            <th>Total Caught</th>
            <th>Total Length (cm)</th>
        </tr>
    </thead>
    <tbody id="summaryTable"></tbody>
    <tfoot>
        <tr class="total-row">
            <td><strong>Overall Total Length</strong></td>
            <td><strong id="totalFishCaught">0</strong></td>
            <td><strong id="totalCombinedLength">0.00 cm</strong></td>
        </tr>
        <tr class="total-row">
            <td><strong>Number of Species Caught</strong></td>
            <td><strong id="totalSpeciesCaught">0</strong></td>
            <td></td>
        </tr>
    </tfoot>

```

```

        </table>
        <button id="endSessionButton" class="btn btn-primary" data-bs-toggle="modal"
data-bs-target="#endSessionModal">End Session <i class="bi bi-arrow-right"></i></button>
    </div>
    <!-- End Session Confirmation Modal -->
    <div class="modal fade" id="endSessionModal" tabindex="-1"
aria-labelledby="endSessionModalLabel" aria-hidden="true">
        <div class="modal-dialog">
            <div class="modal-content">
                <div class="modal-header">
                    <h5 class="modal-title" id="endSessionModalLabel">Confirm End
Session</h5>
                    <button type="button" class="btn-close" data-bs-dismiss="modal"
aria-label="Close"></button>
                </div>
                <div class="modal-body">
                    Are you sure your session is ended?
                </div>
                <div class="modal-footer">
                    <button type="button" class="btn btn-primary"
data-bs-dismiss="modal">CONTINUE FISHING</button>
                    <button type="button" class="btn btn-danger"
id="confirmEndSessionBtn">END MY SESSION</button>
                </div>
            </div>
        </div>
    </div>
    <!-- Tally Verification Page -->
    <div id="tallyVerificationPage" style="display: none;" class="card-section">
        <h2>Catch Summary & Verification</h2>
        <table class="table">
            <thead>
                <tr>
                    <th>Species</th>
                    <th>Total Caught</th>
                    <th>Total Length (cm)</th>
                </tr>
            </thead>
            <tbody id="tallyTable"></tbody>
            <tfoot>
                <tr class="total-row">
                    <td><strong>Overall Total Length</strong></td>
                    <td><strong id="tallyTotalFishCaught">0</strong></td>
                    <td><strong id="tallyTotalCombinedLength">0.00 cm</strong></td>
                </tr>
                <tr class="total-row">
                    <td><strong>Number of Species Caught</strong></td>
                    <td><strong id="tallyTotalSpeciesCaught">0</strong></td>

```

```

        <td></td>
      </tr>
    </tfoot>
  </table>
  <div class="mb-3">
    <label for="catchVerifiedBy" class="form-label">Catch Verified By (For
Competitions Only)</label>
    <input type="text" id="catchVerifiedBy" class="form-control" placeholder="Enter
name of verifier" disabled>
  </div>
  <button class="btn btn-primary mb-3" id="saveAndEndSessionBtn">Save & End
Session <i class="bi bi-check-circle"></i></button>
</div>
</div>
<script>
  // Fish categories data
  const samfFishCategories = {
    "Flatfish": ["Brill", "Dab", "Flounder", "Halibut", "Megrim", "Plaice", "Sole (Dover)",
"Sole (Lemon)", "Sole (Thickback)", "Turbot"],
    "Round Fish": ["Bass", "Bream (Black)", "Bream (Gilthead)", "Bream (Red)", "Bream
(Sea)", "Coalfish", "Cod", "Conger Eel", "Gurnard (Red)", "Gurnard (Tub)", "Haddock",
"Hake", "John Dory", "Ling", "Mackerel", "Monkfish", "Mullet (Grey)", "Mullet (Red)",
"Pollack", "Poor Cod", "Pouting", "Saithe", "Salmon", "Shark (Blue)", "Shark (Porbeagle)",
"Shark (Thresher)", "Smoothhound", "Sprat", "Tope", "Triggerfish", "Whiting"],
    "Rays and Skates": ["Ray (Blonde)", "Ray (Small-eyed)", "Ray (Spotted)", "Ray
(Starry)", "Ray (Thornback)", "Skate"],
    "Wrasse": ["Wrasse (Ballan)", "Wrasse (Cuckoo)"]
  };
  let fishData = [];
  let currentAngler = null;
  let currentDate = null;
  let currentVenue = null;
  let currentZone = null;
  let currentPeg = null;

  // Populate species dropdown
  function populateSpeciesDropdown() {
    const speciesInput = document.getElementById('speciesInput');
    speciesInput.innerHTML = '<option value="" selected>Select a
species</option><option value="Other">Other (specify)</option>';
    for (const category in samfFishCategories) {
      const optgroup = document.createElement('optgroup');
      optgroup.label = category;
      samfFishCategories[category].forEach(species => {
        const option = document.createElement('option');
        option.value = species;
        option.textContent = species;
        optgroup.appendChild(option);
      });
    }
  }

```

```

    });
    speciesInput.appendChild(optgroup);
  }
}

// Toggle custom species input visibility
function toggleCustomSpeciesInput() {
  const speciesInput = document.getElementById('speciesInput');
  const customSpeciesInput = document.getElementById('customSpeciesInput');
  customSpeciesInput.style.display = speciesInput.value === 'Other' ? 'block' : 'none';
}

// Start fishing
function startFishing() {
  document.getElementById('splashScreen').style.display = 'none';
  document.getElementById('anglerRegistration').style.display = 'block';
}

// Toggle competition fields
function toggleCompetitionFields() {
  const isCompetition = document.getElementById('isCompetition').checked;
  document.getElementById('zoneInput').disabled = !isCompetition;
  document.getElementById('pegInput').disabled = !isCompetition;
  document.getElementById('catchVerifiedBy').disabled = !isCompetition;
}

// Register angler
function registerAngler() {
  const anglerName = document.getElementById('anglerName').value.trim();
  const competitionDate = document.getElementById('competitionDate').value;
  const venueInput = document.getElementById('venueInput').value.trim();
  const isCompetition = document.getElementById('isCompetition').checked;
  const zoneInput = document.getElementById('zoneInput').value;
  const pegInput = document.getElementById('pegInput').value;
  if (!anglerName || !competitionDate || !venueInput) {
    alert("Please fill in all required fields!");
    return;
  }
  if (isCompetition && (!zoneInput || !pegInput)) {
    alert("Please select a zone and peg for the competition!");
    return;
  }
  currentAngler = anglerName;
  currentDate = competitionDate;
  currentVenue = venueInput;
  currentZone = zoneInput;
  currentPeg = pegInput;
  document.getElementById('currentAnglerName').textContent = currentAngler;

```

```

document.getElementById('currentDate').textContent = currentDate;
document.getElementById('currentVenue').textContent = currentVenue;
document.getElementById('currentZone').textContent = currentZone;
document.getElementById('currentPeg').textContent = currentPeg;
populateSpeciesDropdown();
document.getElementById('anglerRegistration').style.display = 'none';
document.getElementById('anglerFishingPage').style.display = 'block';
}

// Add fish
function addFish() {
  const speciesInput = document.getElementById('speciesInput');
  const customSpeciesInput = document.getElementById('customSpeciesInput');
  const lengthInput = document.getElementById('lengthInput');
  let species = speciesInput.value;
  const length = parseFloat(lengthInput.value);
  if (species === 'Other') {
    species = customSpeciesInput.value.trim();
    if (!species) {
      alert("Please enter a custom species name!");
      return;
    }
  } else if (species === "") {
    alert("Please select a species!");
    return;
  }
  if (isNaN(length) || length <= 0 || length > 500) {
    alert("Please enter a valid length (0-500 cm)!");
    return;
  }
  const newFish = { species, length };
  fishData.push(newFish);
  localStorage.setItem(`session_${currentAngler}`, JSON.stringify({ fishData,
currentAngler, currentDate, currentVenue, currentZone, currentPeg }));
  updateFishLog();
  updateSummary();
  speciesInput.value = "";
  customSpeciesInput.value = "";
  lengthInput.value = "";
  toggleCustomSpeciesInput();
}

// Update fish log
function updateFishLog() {
  const fishLog = document.getElementById('fishLog');
  fishLog.innerHTML = "";
  fishData.forEach((fish, index) => {
    const row = document.createElement('tr');

```



```

        row.innerHTML = `<td>${fish.species}</td><td>${fish.length} cm</td><td><button
class="btn btn-danger btn-sm" onclick="deleteFish(${index})">Delete</button></td>`;
        fishLog.appendChild(row);
    });
}

// Update summary
function updateSummary() {
    const summaryTable = document.getElementById('summaryTable');
    const totalFishCaught = document.getElementById('totalFishCaught');
    const totalCombinedLength = document.getElementById('totalCombinedLength');
    const totalSpeciesCaught = document.getElementById('totalSpeciesCaught');
    summaryTable.innerHTML = "";
    const summary = {};
    let overallTotalFish = 0;
    let overallTotalLength = 0;
    let uniqueSpecies = new Set();
    fishData.forEach(fish => {
        summary[fish.species] = summary[fish.species] || { count: 0, totalLength: 0 };
        summary[fish.species].count++;
        summary[fish.species].totalLength += fish.length;
        overallTotalFish++;
        overallTotalLength += fish.length;
        uniqueSpecies.add(fish.species);
    });
    for (const species in summary) {
        const row = document.createElement('tr');
        row.innerHTML =
`<td>${species}</td><td>${summary[species].count}</td><td>${summary[species].totalLength.toFixed(2)} cm</td>`;
        summaryTable.appendChild(row);
    }
    totalFishCaught.textContent = overallTotalFish;
    totalCombinedLength.textContent = overallTotalLength.toFixed(2) + ' cm';
    totalSpeciesCaught.textContent = uniqueSpecies.size;
}

// Delete fish
function deleteFish(index) {
    fishData.splice(index, 1);
    localStorage.setItem('session_${currentAngler}', JSON.stringify({ fishData,
currentAngler, currentDate, currentVenue, currentZone, currentPeg }));
    updateFishLog();
    updateSummary();
}

// Back to registration
function goBackToRegistration() {

```

```

    document.getElementById('anglerFishingPage').style.display = 'none';
    document.getElementById('anglerRegistration').style.display = 'block';
}

// End session
function endSession() {
    document.getElementById('anglerFishingPage').style.display = 'none';
    document.getElementById('tallyVerificationPage').style.display = 'block';
    populateTallyTable();
}

// Populate tally table
function populateTallyTable() {
    const tallyTable = document.getElementById('tallyTable');
    const tallyTotalFishCaught = document.getElementById('tallyTotalFishCaught');
    const tallyTotalCombinedLength =
document.getElementById('tallyTotalCombinedLength');
    const tallyTotalSpeciesCaught =
document.getElementById('tallyTotalSpeciesCaught');
    tallyTable.innerHTML = "";
    const summary = {};
    let overallTotalFish = 0;
    let overallTotalLength = 0;
    let uniqueSpecies = new Set();
    fishData.forEach(fish => {
        summary[fish.species] = summary[fish.species] || { count: 0, totalLength: 0 };
        summary[fish.species].count++;
        summary[fish.species].totalLength += fish.length;
        overallTotalFish++;
        overallTotalLength += fish.length;
        uniqueSpecies.add(fish.species);
    });
    for (const species in summary) {
        const row = document.createElement('tr');
        row.innerHTML =
`<td>${species}</td><td>${summary[species].count}</td><td>${summary[species].totalLength.toFixed(2)} cm</td>`;
        tallyTable.appendChild(row);
    }
    tallyTotalFishCaught.textContent = overallTotalFish;
    tallyTotalCombinedLength.textContent = overallTotalLength.toFixed(2) + ' cm';
    tallyTotalSpeciesCaught.textContent = uniqueSpecies.size;
}

// Load saved scorecards
function loadSavedScorecards() {
    const savedScorecardsList = document.getElementById('savedScorecardsList');
    savedScorecardsList.innerHTML = "";

```

```

for (let i = 0; i < localStorage.length; i++) {
  const key = localStorage.key(i);
  if (key.startsWith('scorecard_')) {
    const scorecardData = JSON.parse(localStorage.getItem(key));
    const scorecardContainer = document.createElement('div');
    scorecardContainer.classList.add('card', 'mb-3');
    const cardBody = document.createElement('div');
    cardBody.classList.add('card-body');
    cardBody.innerHTML = `
      <h5 class="card-title">Scorecard - ${scorecardData.date} -
    ${scorecardData.venue}</h5>
      <p class="card-text">Angler: ${scorecardData.angler}, Zone:
    ${scorecardData.zone || 'N/A'}, Peg: ${scorecardData.peg || 'N/A'}</p>
      <table class="table">
        <thead><tr><th>Species</th><th>Length (cm)</th></tr></thead>
        <tbody>${scorecardData.fishData.map(f =>
    `<tr><td>${f.species}</td><td>${f.length}</td></tr>`).join("")} || `<tr><td colspan="2">No fish
    caught.</td></tr>`</tbody>
      </table>
      <p class="card-text"><strong>Total Fish:</strong>
    ${scorecardData.fishData.length}, <strong>Total Length:</strong>
    ${scorecardData.fishData.reduce((sum, f) => sum + f.length, 0).toFixed(2)} cm</p>
      <button class="btn btn-danger btn-sm"
    onclick="localStorage.removeItem('${key}'); loadSavedScorecards();">Delete
    Scorecard</button>
    `;
    scorecardContainer.appendChild(cardBody);
    savedScorecardsList.appendChild(scorecardContainer);
  }
}

```

// Save & End Session Button Logic

```
document.getElementById('saveAndEndSessionBtn').addEventListener('click', () => {
```

```
  // Save the scorecard
```

```
  const sessionData = {
```

```
    angler: currentAngler,
```

```
    date: currentDate,
```

```
    venue: currentVenue,
```

```
    zone: currentZone || 'N/A',
```

```
    peg: currentPeg || 'N/A',
```

```
    fishData,
```

```
    catchVerifiedBy: document.getElementById('catchVerifiedBy').value.trim() || 'N/A'
```

```
  };
```

```
  const scorecardId = `scorecard_${Date.now()}`;
```

```
  localStorage.setItem(scorecardId, JSON.stringify(sessionData));
```

```
  // Prompt user to email the scorecard
```

```

    const shouldEmail = confirm("Scorecard saved successfully! Would you like to email
it to a recipient?");
    if (shouldEmail) {
        const emailInput = prompt("Please enter the recipient's email address:");
        if (emailInput && validateEmail(emailInput)) {
            const emailBody = `
                <h2>Sea Catch Scorer - Scorecard</h2>
                <h3>General Information</h3>
                <p><strong>Angler:</strong> ${sessionData.angler}</p>
                <p><strong>Date:</strong> ${sessionData.date}</p>
                <p><strong>Venue:</strong> ${sessionData.venue}</p>
                <p><strong>Zone:</strong> ${sessionData.zone}</p>
                <p><strong>Peg:</strong> ${sessionData.peg}</p>
                ${sessionData.catchVerifiedBy !== 'N/A' ? `<p><strong>Verified By:</strong>
${sessionData.catchVerifiedBy}</p>` : ""}
                <h3>Fish Data</h3>
                <table border="1">
                    <tr><th>Species</th><th>Length (cm)</th></tr>
                    ${sessionData.fishData.length > 0 ? sessionData.fishData.map(f =>
`<tr><td>${f.species}</td><td>${f.length}</td></tr>`).join("") : `<tr><td colspan="2">No fish
caught.</td></tr>`}
                </table>
                <p><strong>Total Fish:</strong> ${sessionData.fishData.length}</p>
                <p><strong>Total Length:</strong> ${sessionData.fishData.reduce((sum, f)
=> sum + f.length, 0).toFixed(2)} cm</p>
            `;
            const mailtoLink = `mailto:${emailInput}?subject=Sea Catch Scorer -
Scorecard&body=${encodeURIComponent(emailBody)}`;
            window.location.href = mailtoLink;
        } else if (emailInput && !validateEmail(emailInput)) {
            alert("Invalid email address entered. The scorecard will not be emailed.");
        }
    }

    // Reset app and return to splash screen
    resetApp();
});

// Validate email
function validateEmail(email) {
    return /^[^\\s@]+@[^\\s@]+\\.([^\\s@]+)$/.test(email);
}

// Reset app state
function resetApp() {
    fishData = [];
    currentAngler = null;
    currentDate = null;

```

```

    currentVenue = null;
    currentZone = null;
    currentPeg = null;
    localStorage.removeItem(`session_${currentAngler}`);
    document.getElementById('tallyVerificationPage').style.display = 'none';
    document.getElementById('splashScreen').style.display = 'flex';
}

// Event listeners
document.addEventListener('DOMContentLoaded', () => {
    document.getElementById('startFishingBtn').addEventListener('click', registerAngler);
    document.getElementById('addFishBtn').addEventListener('click', addFish);
    document.getElementById('backToRegistrationBtn').addEventListener('click',
goBackToRegistration);
    document.getElementById('confirmEndSessionBtn').addEventListener('click', () => {
        endSession();
        const endSessionModal =
bootstrap.Modal.getInstance(document.getElementById('endSessionModal'));
        endSessionModal.hide();
    });

    document.getElementById('savedScorecardsModal').addEventListener('show.bs.modal',
loadSavedScorecards);
    toggleCompetitionFields();
});
</script>
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"></script>
</body>
</html>

```