Provide here the image for the structure

When the manager opens the staff account management page, the system reads the user data from users.json and displays each staff member (excluding customers). This is done using a loop that dynamically builds a visual list of users with their basic information. This matches the diagram’s **“Views Staff Accounts”** action.

|  |
| --- |
| // Build rows for display      $Rows = '';      foreach ($users as $user) {          // Skip customers entirely          if ($user['role'] === 'customer') {              continue;          }          $Rows .= "<div class='user-block' id='" . htmlspecialchars($user['username']) . "'>";          $Rows .= "<form method='POST' action='StaffManagement.php'>";          $Rows .= "<table>";  …  else {            // Read-only display            $Rows .= "<tr><th>Name</th></tr>";            $Rows .= "<tr><td>" . htmlspecialchars($user['name']) . "</td></tr>";            $Rows .= "<tr><th>Surname</th></tr>";            $Rows .= "<tr><td>" . htmlspecialchars($user['surname']) . "</td></tr>";            $Rows .= "<tr><th>Email</th></tr>";            $Rows .= "<tr><td>" . htmlspecialchars($user['email']) . "</td></tr>";            $Rows .= "<tr><th>Phone Number</th></tr>";            $Rows .= "<tr><td>" . htmlspecialchars($user['phone']) . "</td></tr>";            $Rows .= "<tr><th>Role</th></tr>";            $Rows .= "<tr><td>" . htmlspecialchars($user['role']) . "</td></tr>";            $Rows .= "<tr>";            $Rows .= "<td>";            $Rows .= "<a href='StaffManagement.php?edit=" . urlencode($user['username']) . "' class='edit-button'>Edit</a>";            $Rows .= "<a href='../htmlfiles/OldPassword.html' class='change-password'>Change Password</a>";            $Rows .= "</td>";            $Rows .= "</tr>";          } |

The manager can click the "Add" button, which uses JavaScript to generate a blank form for entering a new staff member's details. When submitted, a new user entry is added to the existing users array and saved to the JSON file. This reflects the **“Create New Staff Account”** branch in the diagram.

|  |
| --- |
| if ($\_SERVER['REQUEST\_METHOD'] === 'POST' && isset($\_POST['add\_new\_user'])) {      // Create a new user entry      $newUser = [          'username' => $\_POST['username'],          'name' => $\_POST['name'],          'surname' => $\_POST['surname'],          'email' => $\_POST['email'],          'phone' => $\_POST['phone'],          'role' => $\_POST['role']      ];      // add to the user list      $users[] = $newUser;      // Save back to JSON      $json\_en = json\_encode($users, JSON\_PRETTY\_PRINT);      file\_put\_contents($json, $json\_en);      // Redirect back to the new user's section      header("Location: StaffManagement.php#" . urlencode($newUser['username']));      exit();  } |

The new user form includes fields for name, surname, email, phone number, role, and a generated username. These input fields represent the **“Inputs Details”** step in the diagram, where the manager provides the necessary data for a new account.

|  |
| --- |
| // Build rows for display      $Rows = '';      foreach ($users as $user) {          // Skip customers entirely          if ($user['role'] === 'customer') {              continue;          }          $Rows .= "<div class='user-block' id='" . htmlspecialchars($user['username']) . "'>";          $Rows .= "<form method='POST' action='StaffManagement.php'>";          $Rows .= "<table>";          if (isset($\_GET['edit']) && $\_GET['edit'] === $user['username']) {              // Editable fields            $Rows .= "<tr><th>Name</th></tr>";            $Rows .= "<tr><td><input type='text' name='name' value='" . htmlspecialchars($user['name']) . "' disabled></td></tr>";            $Rows .= "<tr><th>Surname</th></tr>";            $Rows .= "<tr><td><input type='text' name='surname' value='" . htmlspecialchars($user['surname']) . "' disabled></td></tr>";            $Rows .= "<tr><th>Email</th></tr>";            $Rows .= "<tr><td><input type='email' name='email' value='" . htmlspecialchars($user['email']) . "' required></td></tr>";            $Rows .= "<tr><th>Phone Number</th></tr>";            $Rows .= "<tr><td><input type='text' name='phone' value='" . htmlspecialchars($user['phone']) . "' required></td></tr>";            // Role dropdown            $Rows .= "<tr><th>Role</th></tr>";            $Rows .= "<tr><td><select name='role' required>";            foreach ($roles as $r) {                $sel = ($user['role'] === $r) ? ' selected' : '';                $Rows .= "<option value=\"{$r}\"{$sel}>" . ucfirst($r) . "</option>";            }            $Rows .= "</select></td></tr>";            $Rows .= "<tr>";            $Rows .= "<td>";            $Rows .= "<input type='hidden' name='target\_username' value='" . htmlspecialchars($user['username']) . "'>";            $Rows .= "<button type='submit' name='save' class='save-button'>Save</button>";            $Rows .= "</td>";            $Rows .= "</tr>"; |

Once the form is filled, clicking the "Save" button triggers a POST request that appends the new user to the JSON array. The updated array is then saved using file\_put\_contents. This matches the **“Saves Details”** action handled by the system.

|  |
| --- |
| //saves users detail changes    if ($\_SERVER['REQUEST\_METHOD'] === 'POST' && isset($\_POST['save'])) {      $targetUsername = $\_POST['target\_username'];      $updatedEmail = $\_POST['email'];      $updatedPhone = $\_POST['phone'];      $updatedRole = $\_POST['role'];      foreach ($users as &$user) {          if ($user['username'] === $targetUsername) {              $user['email'] = $updatedEmail;              $user['phone'] = $updatedPhone;              $user['role'] = $updatedRole;              break;          }      }      $json\_en = json\_encode($users, JSON\_PRETTY\_PRINT);      file\_put\_contents($json, $json\_en);      header("Location: StaffManagement.php#" . urlencode($targetUsername));      exit();  } |

If the manager chooses to modify an existing account, they click an "Edit" link beside a staff member. This triggers edit mode using a GET parameter. Once changes are submitted, the backend locates the user and updates the email, phone number, or role. This follows the **“Modify Existing Staff Account”** path in the diagram.

|  |
| --- |
| if (isset($\_GET['edit']) && $\_GET['edit'] === $user['username']) {              // Editable fields            $Rows .= "<tr><th>Name</th></tr>";            $Rows .= "<tr><td><input type='text' name='name' value='" . htmlspecialchars($user['name']) . "' disabled></td></tr>";            $Rows .= "<tr><th>Surname</th></tr>";            $Rows .= "<tr><td><input type='text' name='surname' value='" . htmlspecialchars($user['surname']) . "' disabled></td></tr>";            $Rows .= "<tr><th>Email</th></tr>";            $Rows .= "<tr><td><input type='email' name='email' value='" . htmlspecialchars($user['email']) . "' required></td></tr>";            $Rows .= "<tr><th>Phone Number</th></tr>";            $Rows .= "<tr><td><input type='text' name='phone' value='" . htmlspecialchars($user['phone']) . "' required></td></tr>";            // Role dropdown            $Rows .= "<tr><th>Role</th></tr>";            $Rows .= "<tr><td><select name='role' required>";            foreach ($roles as $r) {                $sel = ($user['role'] === $r) ? ' selected' : '';                $Rows .= "<option value=\"{$r}\"{$sel}>" . ucfirst($r) . "</option>";            }            $Rows .= "</select></td></tr>";            $Rows .= "<tr>";            $Rows .= "<td>";            $Rows .= "<input type='hidden' name='target\_username' value='" . htmlspecialchars($user['username']) . "'>";            $Rows .= "<button type='submit' name='save' class='save-button'>Save</button>";            $Rows .= "</td>";            $Rows .= "</tr>"; |

Any changes made — whether adding a new account or editing an existing one — are saved to the users.json file using json\_encode() and file\_put\_contents(). This represents the **“Updates Database”** action shown on the system side of the diagram. Saving always with:

|  |
| --- |
| //read json  $json = '../Data/users.json';  …….  if (file\_exists($json)) {    $jsonData = file\_get\_contents($json);    $users = json\_decode($jsonData, true);  ……..   // Save back to JSON      $json\_en = json\_encode($users, JSON\_PRETTY\_PRINT);      file\_put\_contents($json, $json\_en); |

If the manager is done editing, saving, deleting or adding staff accounts they can return to the dashboard or any other site with the help of the sidebar.

|  |
| --- |
| <!-- Sidebar Navigation -->    <div class="sidebar" id="sidebar">        <a href="../phpfiles/dash.php"><span>Dashboard</span></a>        <a href="../phpFiles/AccountManagement.php"><span>Account Management</span></a>        <a href="#"><span>Analytics</span></a>        <a href="../phpFiles/Schedule.php"><span>Schedule</span></a>        <a href="../phpFiles/inventory.php"><span>Inventory</span></a>        <a href="../phpFiles/manage\_reservations.php"><span>Reservations</span></a>        <a href="../phpFiles/orders.php"><span>Orders</span></a>        <a href="../phpFiles/StaffManagement.php"><span>Staff Management</span></a>        <a href="../phpFiles/scheduleManager.php"><span>Schedule Management</span></a>        <a href="../phpFiles/PreviousOrders.php"><span>Previous Orders</span></a>      </div> |