In **AccountManagement.php**, the code dynamically generated HTML table rows to display the logged-in user's profile data — such as name, surname, email, phone number, and role. Depending on whether the user is editing their profile, the table either displayed plain text or rendered input fields for email and phone. While functional, the code was long and repetitive, with the same HTML structure repeated for each field, making it harder to maintain and extend.

|  |
| --- |
| //creating table row with table data to showcase users info: changeable and non changeable if editing or if not  $Rows = '';  foreach ($users as $user) {  if (isset($\_GET['edit']) && isset($user['username']) && $\_SESSION['usernm'] == $user['username']) {  $Rows .= "  <form method='POST' action='AccountManagement.php'>  <tr><th>Name</th></tr>  <tr>  <td><input type='text' name='name' value='" . htmlspecialchars($user['name']) . "' disabled></td>  </tr>  <tr><th>Surname</th></tr>  <tr>  <td><input type='text' name='surname' value='" . htmlspecialchars($user['surname']) . "' disabled></td>  </tr>  <tr><th>Email</th></tr>  <tr>  <td><input type='email' name='email' value='" . htmlspecialchars($user['email']) . "' required></td>  </tr>  <tr><th>Phone Number</th></tr>  <tr>  <td><input type='text' name='phone' value='" . htmlspecialchars($user['phone']) . "' required></td>  </tr>  <tr><th>Role</th></tr>  <tr>  <td><input type='text' name='role' value='" . htmlspecialchars($user['role']) . "' disabled></td>  </tr>  <tr>  <td>  <button type='submit' name='save' class='save-button'>Save</button>  </td>  </form>  </tr>";  } else if ($\_SESSION['usernm'] == $user['username']){  $Rows .= "<tr>  <tr><th>Name</th></tr>  <td>" . htmlspecialchars($user['name']) . "</td>  </tr>  <tr><th>Surname</th></tr>  <tr>  <td>" . htmlspecialchars($user['surname']) . "</td>  </tr>  <tr><th>Email</th></tr>  <tr>  <td>" . htmlspecialchars($user['email']) . "</td>  </tr>  <tr><th>Phone Number</th></tr>  <tr>  <td>" . htmlspecialchars($user['phone']) . "</td>  </tr>  <tr><th>Role</th></tr>  <tr>  <td>".htmlspecialchars($user['role'])."</td>  </tr>  <tr>  <td><a href='AccountManagement.php?edit=1' class='edit-button'>Edit</a>  <a href='../htmlfiles/OldPassword.html' class='change-password'>Change Password</a></td>  </tr>";  }  }  } else {  $Rows = "<tr><td colspan='5'>Error: User data file not found.</td></tr>";  } |

To improve structure and readability, the code was refactored by introducing a helper function named generateInputRow() inside **AccountManagement.php**. This function takes the field label, input name, value, and whether it should be editable, and returns a clean HTML row. This change eliminated duplicated code, made the layout easier to follow, and allows future modifications to input formatting or structure to be done in one place. The refactored code is more modular and maintainable while preserving all original functionality.

|  |
| --- |
| //this function will create a table row with label and tits input field disabled or required.     function generateInputRow($label, $name, $value, $editable = false) {      $disabled = $editable ? '' : 'disabled';      $required = $editable ? 'required' : '';      return "          <tr><th>{$label}</th></tr>          <tr>              <td><input type='text' name='{$name}' value='" . htmlspecialchars($value) . "' {$disabled} {$required}></td>          </tr>      ";  }    //creating  table row with table data to showcase users info: changeable and non changeable if editing or if not      $Rows = '';      foreach ($users as $user) {          if ($\_SESSION['usernm'] === $user['username']) {              $isEditing = isset($\_GET['edit']);              if ($isEditing) {                  $Rows .= "<form method='POST' action='AccountManagement.php'>";                  $Rows .= generateInputRow('Name', 'name', $user['name']);                  $Rows .= generateInputRow('Surname', 'surname', $user['surname']);                  $Rows .= generateInputRow('Email', 'email', $user['email'], true);                  $Rows .= generateInputRow('Phone Number', 'phone', $user['phone'], true);                  $Rows .= generateInputRow('Role', 'role', $user['role']);                  $Rows .= "<tr><td><button type='submit' name='save' class='save-button'>Save</button></td></tr>";                  $Rows .= "</form>";              } else {                  $Rows .= "<tr><th>Name</th></tr><tr><td>" . htmlspecialchars($user['name']) . "</td></tr>";                  $Rows .= "<tr><th>Surname</th></tr><tr><td>" . htmlspecialchars($user['surname']) . "</td></tr>";                  $Rows .= "<tr><th>Email</th></tr><tr><td>" . htmlspecialchars($user['email']) . "</td></tr>";                  $Rows .= "<tr><th>Phone Number</th></tr><tr><td>" . htmlspecialchars($user['phone']) . "</td></tr>";                  $Rows .= "<tr><th>Role</th></tr><tr><td>" . htmlspecialchars($user['role']) . "</td></tr>";                  $Rows .= "<tr><td>                      <a href='AccountManagement.php?edit=1' class='edit-button'>Edit</a>                      <a href='../htmlfiles/OldPassword.html' class='change-password'>Change Password</a>                  </td></tr>";              }          }      }  } else {      $Rows = "<tr><td colspan='5'>Error: User data file not found.</td></tr>";  } |