ΠΑΡΑΡΤΗΜΑ

- **Περιγραφή: Τα παρακάτω αρχεία δομούν το γραφικό περιβάλλον της ιντερνετικής εφαρμογής στα επίπεδα εισαγωγής/διαγραφής δεδομένων, αρχικοποίησης βάσης και για την εκτέλεση των SQL ερωτημάτων.**

Αρχείο 1 : index.php

```
<!DOCTYPE html>
<html lang="en">
  <?php
  // include ('connection_local.php');
  // $db = connecting();
  include ('connection.php');
  $db_conn = connection();
```

```
 if(isset($_GET['status']) == 'exists'){
   echo'
  <script>alert("Entry Already exists. Try another one.");</script>';
   header('Location: index.php');
 }
  ?>
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Μετρήσεις Κινητών Συσκευών Χρηστών</title>
    <link rel="stylesheet" type="text/css" href="static/style.css">
    <link rel="icon" type="image/x-icon" href="media/sensor.png">
</head>
<body>
```

```
<!-- The sidebar -->
<div class="sidebar">
  <a class="active menu-text" href="index.php">DATABASE INITIALIZATION</a>
  <a class="menu-text" href="users.php">USERS</a>
```

```
  <div class="dropdown">
  <a class="menu-text" href="sensors.php">SENSORS</a>
  <!-- The dropdown content -->
  <div class="dropdown-content">
```

```
        <a class="menu-text active"
href="magnetometer_sensor.php">Magnetometer</a>
        <a class="menu-text" href="accelerometer_sensor.php">Accelerometer</a>
        <a class="menu-text" href="geolocation_sensor.php">Geolocation</a>
        <a class="menu-text" href="gyroscope_sensor.php">Gyroscope</a>
        <a class="menu-text" href="step-counter_sensor.php">Step-counter</a>
        <a class="menu-text" href="barometer_sensor.php">Barometer</a>
        <a class="menu-text" href="proximity_sensor.php">Proximity</a>
  </div>
  </div>
  <a class="menu-text" href="queries.php">QUERIES</a>
</div>
<!-- Page content -->
<div class="content">
  <form action="database_init.php" method="post">
      <button type="submit" name="submit">DATABASE INITIALIASATION</button><!--
DATABASE INITIALIASATION -->
  </form>
</div>
</body>
</html>
```

Αρχείο 2 : accelerometer_sensor.php

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Μετρήσεις Κινητών Συσκευών Χρηστών</title>
    <link rel="stylesheet" type="text/css" href="static/style.css">
    <link rel="icon" type="image/x-icon" href="media/sensor.png">
</head>
<body>

<!-- The sidebar -->
<div class="sidebar">
  <a class="menu-text" href="index.php">DATABASE INITIALIZATION</a>
  <a class="menu-text" href="users.php">USERS</a>
  <div class="dropdown">
  <a class="menu-text active" href="sensors.php">SENSORS</a>
  <!-- The dropdown content -->
```

```
  <div class="dropdown-content">

        <a class="menu-text" href="magnetometer_sensor.php">Magnetometer</a>
        <a class="menu-text active"
href="accelerometer_sensor.php">Accelerometer</a>
        <a class="menu-text" href="geolocation_sensor.php">Geolocation</a>
        <a class="menu-text" href="gyroscope_sensor.php">Gyroscope</a>
        <a class="menu-text" href="step-counter_sensor.php">Step-counter</a>
        <a class="menu-text" href="barometer_sensor.php">Barometer</a>
        <a class="menu-text" href="proximity_sensor.php">Proximity</a>
  </div>
  </div>
  <a class="menu-text" href="queries.php">QUERIES</a>
</div>
<!-- Page content -->
<div class="content">
<form action="accelerometer/insert.php" method="post">
    <table>
        <tr>
            <th>Field</th>
            <th>Value</th>
        </tr>
        <tr>
            <td>UserId:</td>
            <td><input type="text" name="userid" required></td>
        </tr>

        <tr>
            <td>Accelerometer_x:</td>
            <td><input type="text" name="accelerometer_x"
step="0.00000000000000001" required></td>
        </tr>
        <tr>
            <td>Accelerometer_y:</td>
            <td><input type="text" name="accelerometer_y"
step="0.00000000000000001" required></td>
        </tr>
        <tr>
            <td>Accelerometer_z:</td>
            <td><input type="text" name="accelerometer_z"
step="0.00000000000000001" required></td>
        </tr>
        <tr>
            <td>Time Stamp(Unix Epoch):</td>
            <td><input type="text" name="time stamp" required></td>
```

```
        </tr>
    </table>
    <input type="submit" value="Insert Record">
</form>
<form action="accelerometer/delete.php" method="post">
    <table>
    <tr>
            <th>Field</th>
            <th>Value</th>
        </tr>
        <tr>
            <td>UserId:</td>
            <td><input type="text" name="userid" required></td>
        </tr>
        <tr>
            <td>Accelerometer_x:</td>
            <td><input type="text" name="accelerometer_x"
step="0.00000000000000001" required></td>
        </tr>
        <tr>
            <td>Accelerometer_y:</td>
            <td><input type="text" name="accelerometer_y"
step="0.00000000000000001" required></td>
        </tr>
        <tr>
            <td>Accelerometer_z:</td>
            <td><input type="text" name="accelerometer_z"
step="0.00000000000000001" required></td>
        </tr>
        <tr>
            <td>Time Stamp(Unix Epoch):</td>
            <td><input type="text" name="time stamp" required></td>
        </tr>
    </table>
    <input type="submit" value="Delete Record">
</form>
</div>

</body>
</html>
```

Αρχείο 3 : barometer_sensor.php

```
<!DOCTYPE html>
```

```
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Μετρήσεις Κινητών Συσκευών Χρηστών</title>
    <link rel="stylesheet" type="text/css" href="static/style.css">
    <link rel="icon" type="image/x-icon" href="media/sensor.png">
</head>
<body>

<!-- The sidebar -->
<div class="sidebar">
  <a class="menu-text" href="index.php">DATABASE INITIALIZATION</a>
  <a class="menu-text" href="users.php">USERS</a>
  <div class="dropdown">
  <a class="menu-text active" href="sensors.php">SENSORS</a>
  <!-- The dropdown content -->
  <div class="dropdown-content">
        <a class="menu-text" href="magnetometer_sensor.php">Magnetometer</a>
        <a class="menu-text" href="accelerometer_sensor.php">Accelerometer</a>
        <a class="menu-text" href="geolocation_sensor.php">Geolocation</a>
        <a class="menu-text" href="gyroscope_sensor.php">Gyroscope</a>
        <a class="menu-text" href="step-counter_sensor.php">Step-counter</a>
        <a class="menu-text active" href="barometer_sensor.php">Barometer</a>
        <a class="menu-text" href="proximity_sensor.php">Proximity</a>
  </div>
  </div>
  <a class="menu-text" href="queries.php">QUERIES</a>
</div>
<!-- Page content -->
<div class="content">
<form action="barometer/insert.php" method="post">
    <table>
        <tr>
            <th>Field</th>
            <th>Value</th>
        </tr>
        <tr>
            <td>Userid:</td>
            <td><input type="text" name="userid" required></td>
        </tr>
        <tr>
            <td>Pressure:</td>
            <td><input type="number" name="pressure" required></td>
        </tr>
```

```
        <tr>
            <td>Time_stamp(Unix Epoch):</td>
            <td><input type="number" name="timestamp" required></td>
        </tr>
    </table>
    <input type="submit" value="Insert Record">
</form>
<form action="barometer/delete.php" method="post">
    <table>
        <tr>
            <th>Field</th>
            <th>Value</th>
        </tr>
        <tr>
        <tr>
            <td>Userid:</td>
            <td><input type="text" name="userid" required></td>
        </tr>
        <tr>
            <td>Pressure:</td>
            <td><input type="number" name="pressure" required></td>
        </tr>
        <tr>
            <td>Time_stamp(Unix Epoch):</td>
            <td><input type="number" name="timestamp" required></td>
        </tr>
    </table>
    <input type="submit" value="Delete Record">
</form>
</div>

</body>
</html>
```

Αρχείο 4 : geolocation_sensor.php

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Μετρήσεις Κινητών Συσκευών Χρηστών</title>
    <link rel="stylesheet" type="text/css" href="static/style.css">
    <link rel="icon" type="image/x-icon" href="media/sensor.png">
```

```
</head>
<body>

<!-- The sidebar -->
<div class="sidebar">
  <a class="menu-text" href="index.php">DATABASE INITIALIZATION</a>
  <a class="menu-text" href="users.php">USERS</a>
  <div class="dropdown">
  <a class="menu-text active" href="sensors.php">SENSORS</a>
  <!-- The dropdown content -->
  <div class="dropdown-content">

        <a class="menu-text" href="magnetometer_sensor.php">Magnetometer</a>
        <a class="menu-text" href="accelerometer_sensor.php">Accelerometer</a>
        <a class="menu-text active" href="geolocation_sensor.php">Geolocation</a>
        <a class="menu-text" href="gyroscope_sensor.php">Gyroscope</a>
        <a class="menu-text" href="step-counter_sensor.php">Step-counter</a>
        <a class="menu-text" href="barometer_sensor.php">Barometer</a>
        <a class="menu-text" href="proximity_sensor.php">Proximity</a>
  </div>
  </div>
  <a class="menu-text" href="queries.php">QUERIES</a>
</div>
<!-- Page content -->
<div class="content">
<form action="geolocation/insert.php" method="post">
    <table>
        <tr>
            <th>Field</th>
            <th>Value</th>
        </tr>
        <tr>
            <td>UserId:</td>
            <td><input type="text" name="userid" required></td>
        </tr>
        <tr>
            <td>Longtitude:</td>
            <td><input type="number" name="longtitude" step="0.00000000000000001"
required></td>
        </tr>
        <tr>
            <td>Latitude:</td>
            <td><input type="number" name="latitude" step="0.00000000000000001"
required></td>
        </tr>
```

```
        <tr>
            <td>Altitude:</td>
            <td><input type="number" name="altitude" step="0.00000000000000001"
required></td>
        </tr>
        <tr>
            <td>Time_stamp(Unix Epoch):</td>
            <td><input type="text" name="time_stamp" required></td>
        </tr>
    </table>
    <input type="submit" value="Insert Record">
</form>
<form action="geolocation/delete.php" method="post">
    <table>
        <tr>
            <th>Field</th>
            <th>Value</th>
        </tr>
        <tr>
            <td>UserId:</td>
            <td><input type="text" name="userid" required></td>
        </tr>
        <tr>
            <td>Longtitude:</td>
            <td><input type="number" name="longtitude" step="0.00000000000000001"
required></td>
        </tr>
        <tr>
            <td>Latitude:</td>
            <td><input type="number" name="latitude" step="0.00000000000000001"
required></td>
        </tr>
        <tr>
            <td>Altitude:</td>
            <td><input type="number" name="altitude" step="0.00000000000000001"
required></td>
        </tr>
        <tr>
            <td>Time_stamp(Unix Epoch):</td>
            <td><input type="text" name="time_stamp" required></td>
        </tr>
    </table>
    <input type="submit" value="Delete Record">
</form>
</div>
```

```
</body>
</html>
```

Αρχείο 5 : gyroscope_sensor.php

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Μετρήσεις Κινητών Συσκευών Χρηστών</title>
    <link rel="stylesheet" type="text/css" href="static/style.css">
    <link rel="icon" type="image/x-icon" href="media/sensor.png">
</head>
<body>

<!-- The sidebar -->
<div class="sidebar">
  <a class="menu-text" href="index.php">DATABASE INITIALIZATION</a>
  <a class="menu-text" href="users.php">USERS</a>
  <div class="dropdown">
  <a class="menu-text active" href="sensors.php">SENSORS</a>
  <!-- The dropdown content -->
  <div class="dropdown-content">

        <a class="menu-text" href="magnetometer_sensor.php">Magnetometer</a>
        <a class="menu-text" href="accelerometer_sensor.php">Accelerometer</a>
        <a class="menu-text" href="geolocation_sensor.php">Geolocation</a>
        <a class="menu-text active" href="gyroscope_sensor.php">Gyroscope</a>
        <a class="menu-text" href="step-counter_sensor.php">Step-counter</a>
        <a class="menu-text" href="barometer_sensor.php">Barometer</a>
        <a class="menu-text" href="proximity_sensor.php">Proximity</a>
  </div>
  </div>
  <a class="menu-text" href="queries.php">QUERIES</a>
</div>

<!-- Page content -->
<div class="content">
<form action="gyroscope/insert.php" method="post">
    <table>
        <tr>
```

```
            <th>Field</th>
            <th>Value</th>
        </tr>
        <tr>
            <td>UserId:</td>
            <td><input type="text" name="userid" required></td>
        </tr>
        <tr>
            <td>gyroscope_x:</td>
            <td><input type="number" name="gyroscope_x"
step="0.00000000000000001" required></td>
        </tr>
        <tr>
            <td>gyroscope_y:</td>
            <td><input type="number" name="gyroscope_y"
step="0.00000000000000001" required></td>
        </tr>
        <tr>
            <td>gyroscope_z:</td>
            <td><input type="number" name="gyroscope_z"
step="0.00000000000000001" required></td>
        </tr>
        <tr>
            <td>Time_Stamp(Unix Epoch):</td>
            <td><input type="number" name="time_stamp" required></td>
        </tr>
    </table>
    <input type="submit" value="Insert Record">
</form>
```

```
<form action="gyroscope/delete.php" method="post">
    <table>
        <tr>
            <th>Field</th>
            <th>Value</th>
        </tr>
        <tr>
            <td>UserId:</td>
            <td><input type="text" name="userid" required></td>
        </tr>
        <tr>
            <td>gyroscope_x:</td>
            <td><input type="number" name="gyroscope_x"
step="0.00000000000000001" required></td>
        </tr>
```

```
        <tr>
            <td>gyroscope_y:</td>
            <td><input type="number" name="gyroscope_y"
step="0.00000000000000001" required></td>
        </tr>
        <tr>
            <td>gyroscope_z:</td>
            <td><input type="number" name="gyroscope_z"
step="0.00000000000000001" required></td>
        </tr>
        <tr>
            <td>Time_Stamp(Unix Epoch):</td>
            <td><input type="number" name="time_stamp" required></td>
        </tr>
    </table>
    <input type="submit" value="Delete Record">
</form>
</div>

</body>
</html>
```

Αρχείο 6 : g magnetometer_sensor.php

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Μετρήσεις Κινητών Συσκευών Χρηστών</title>
    <link rel="stylesheet" type="text/css" href="static/style.css">
    <link rel="icon" type="image/x-icon" href="media/sensor.png">
</head>
<body>

<!-- The sidebar -->
<div class="sidebar">
  <a class="menu-text" href="index.php">DATABASE INITIALIZATION</a>
  <a class="menu-text" href="users.php">USERS</a>
  <div class="dropdown">
  <a class="menu-text active" href="sensors.php">SENSORS</a>
  <!-- The dropdown content -->
  <div class="dropdown-content">
```

```
        <a class="menu-text active"
href="magnetometer_sensor.php">Magnetometer</a>
        <a class="menu-text" href="accelerometer_sensor.php">Accelerometer</a>
        <a class="menu-text" href="geolocation_sensor.php">Geolocation</a>
        <a class="menu-text" href="gyroscope_sensor.php">Gyroscope</a>
        <a class="menu-text" href="step-counter_sensor.php">Step-counter</a>
        <a class="menu-text" href="barometer_sensor.php">Barometer</a>
        <a class="menu-text" href="proximity_sensor.php">Proximity</a>
  </div>
  </div>
  <a class="menu-text" href="queries.php">QUERIES</a>
</div>
```

```
<!-- Page content -->
<div class="content">
<form action="magnetometer/insert.php" method="post">
    <table>
        <tr>
            <th>Field</th>
            <th>Value</th>
        </tr>
        <tr>
            <td>UserId:</td>
            <td><input type="text" name="userid" required></td>
        </tr>
        <tr>
            <td>Magnetometer_x:</td>
            <td><input type="number" name="magnetometer_x"
step="0.00000000000000001" required></td>
        </tr>
        <tr>
            <td>Magnetometer_y:</td>
            <td><input type="number" name="magnetometer_y"
step="0.00000000000000001" required></td>
        </tr>
        <tr>
            <td>Magnetometer_z:</td>
            <td><input type="number" name="magnetometer_z"
step="0.00000000000000001" required></td>
        </tr>
        <tr>
            <td>Time_Stamp(Unix Epoch):</td>
            <td><input type="number" name="time_stamp" step="0.00000000000000001"
required></td>
        </tr>
```

```
    </table>
    <input type="submit" value="Insert Record">
</form>
```

```
<form action="magnetometer/delete.php" method="post">
    <table>
        <tr>
            <th>Field</th>
            <th>Value</th>
        </tr>
        <tr>
            <td>UserId:</td>
            <td><input type="text" name="userid" required></td>
        </tr>
        <tr>
            <td>Magnetometer_x:</td>
            <td><input type="number" name="magnetometer_x"
step="0.00000000000000001" required></td>
        </tr>
        <tr>
            <td>Magnetometer_y:</td>
            <td><input type="number" name="magnetometer_y"
step="0.00000000000000001" required></td>
        </tr>
        <tr>
            <td>Magnetometer_z:</td>
            <td><input type="number" name="magnetometer_z"
step="0.00000000000000001" required></td>
        </tr>
        <tr>
            <td>Time_Stamp(Unix Epoch):</td>
            <td><input type="number" name="time_stamp" required></td>
        </tr>
    </table>
    <input type="submit" value="Delete Record">
</form>
</div>

</body>
</html>
```

Αρχείο 7 : proximity_sensor.php

```
<!DOCTYPE html>
```

```
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Μετρήσεις Κινητών Συσκευών Χρηστών</title>
    <link rel="stylesheet" type="text/css" href="static/style.css">
    <link rel="icon" type="image/x-icon" href="media/sensor.png">
</head>
<body>

<!-- The sidebar -->
<div class="sidebar">
  <a class="menu-text" href="index.php">DATABASE INITIALIZATION</a>
  <a class="menu-text" href="users.php">USERS</a>
  <div class="dropdown">
  <a class="menu-text active" href="sensors.php">SENSORS</a>
  <!-- The dropdown content -->
  <div class="dropdown-content">

        <a class="menu-text" href="magnetometer_sensor.php">Magnetometer</a>
        <a class="menu-text" href="accelerometer_sensor.php">Accelerometer</a>
        <a class="menu-text" href="geolocation_sensor.php">Geolocation</a>
        <a class="menu-text" href="gyroscope_sensor.php">Gyroscope</a>
        <a class="menu-text" href="step-counter_sensor.php">Step-counter</a>
        <a class="menu-text" href="barometer_sensor.php">Barometer</a>
        <a class="menu-text active" href="proximity_sensor.php">Proximity</a>
  </div>
  </div>
  <a class="menu-text" href="queries.php">QUERIES</a>
</div>

<!-- Page content -->
<div class="content">
<form action="proximity/insert.php" method="post">
    <table>
        <tr>
            <th>Field</th>
            <th>Value</th>
        </tr>
        <tr>
            <td>Userid:</td>
            <td><input type="text" name="userid" required></td>
        </tr>
        <tr>
            <td>Proximity:</td>
```

```
            <td><input type="text" name="proximity" required></td>
        </tr>
        <tr>
            <td>Time_stamp(Unix Epoch):</td>
            <td><input type="number" name="timestamp" required></td>
        </tr>
    </table>
    <input type="submit" value="Insert Record">
</form>
```

```
<form action="proximity/delete.php" method="post">
    <table>
        <tr>
            <th>Field</th>
            <th>Value</th>
        </tr>
        <tr>
            <td>Userid:</td>
            <td><input type="text" name="userid" required></td>
        </tr>
        <tr>
            <td>Proximity:</td>
            <td><input type="text" name="proximity" required></td>
        </tr>
        <tr>
            <td>Time_stamp(Unix Epoch):</td>
            <td><input type="number" name="timestamp" required></td>
        </tr>
    </table>
    <input type="submit" value="Delete Record">
</form>
</div>

</body>
</html>
```

Αρχείο 8 : step-counter_sensor.php

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Μετρήσεις Κινητών Συσκευών Χρηστών</title>
```

```
    <link rel="stylesheet" type="text/css" href="static/style.css">
    <link rel="icon" type="image/x-icon" href="media/sensor.png">
</head>
<body>

<!-- The sidebar -->
<div class="sidebar">
  <a class="menu-text" href="index.php">DATABASE INITIALIZATION</a>
  <a class="menu-text" href="users.php">USERS</a>
  <div class="dropdown">
  <a class="menu-text active" href="sensors.php">SENSORS</a>
  <!-- The dropdown content -->
  <div class="dropdown-content">

        <a class="menu-text" href="magnetometer_sensor.php">Magnetometer</a>
        <a class="menu-text" href="accelerometer_sensor.php">Accelerometer</a>
        <a class="menu-text" href="geolocation_sensor.php">Geolocation</a>
        <a class="menu-text" href="gyroscope_sensor.php">Gyroscope</a>
        <a class="menu-text active" href="step-counter_sensor.php">Step-
counter</a>
        <a class="menu-text" href="barometer_sensor.php">Barometer</a>
        <a class="menu-text" href="proximity_sensor.php">Proximity</a>
  </div>
  </div>
  <a class="menu-text" href="queries.php">QUERIES</a>
</div>
```

```
<!-- Page content -->
<div class="content">
<form action="step-counter/insert.php" method="post">
    <table>
        <tr>
            <th>Field</th>
            <th>Value</th>
        </tr>
        <tr>
            <td>Userid:</td>
            <td><input type="text" name="userid" required></td>
        </tr>
        <tr>
            <td>Footsteps:</td>
            <td><input type="number" name="footsteps" required></td>
        </tr>
        <tr>
            <td>Time_Stamp(Unix Epoch):</td>
```

```
            <td><input type="number" name="timestamp" required></td>
        </tr>
    </table>
    <input type="submit" value="Insert Record">
</form>
```

```
<form action="step-counter/delete.php" method="post">
    <table>
        <tr>
            <th>Field</th>
            <th>Value</th>
        </tr>
        <tr>
            <td>Userid:</td>
            <td><input type="text" name="userid" required></td>
        </tr>
        <tr>
            <td>Footsteps:</td>
            <td><input type="number" name="footsteps" required></td>
        </tr>
        <tr>
            <td>Time_Stamp(Unix Epoch):</td>
            <td><input type="number" name="timestamp" required></td>
        </tr>
    </table>
    <input type="submit" value="Delete Record">
</form>
</div>

</body>
</html>
```

Αρχείο 9 : users.php

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Μετρήσεις Κινητών Συσκευών Χρηστών</title>
    <link rel="stylesheet" type="text/css" href="static/style.css">
    <link rel="icon" type="image/x-icon" href="media/sensor.png">
</head>
<body>

<!-- The sidebar -->
```

```
<div class="sidebar">
  <a class="menu-text" href="index.php">DATABASE INITIALIZATION</a>
  <a class="active menu-text" href="users.php">USERS</a>
  <div class="dropdown">
  <a class="menu-text" href="sensors.php">SENSORS</a>
  <!-- The dropdown content -->
  <div class="dropdown-content">
        <a class="menu-text active"
href="magnetometer_sensor.php">Magnetometer</a>
        <a class="menu-text" href="accelerometer_sensor.php">Accelerometer</a>
        <a class="menu-text" href="geolocation_sensor.php">Geolocation</a>
        <a class="menu-text" href="gyroscope_sensor.php">Gyroscope</a>
        <a class="menu-text" href="step-counter_sensor.php">Step-counter</a>
        <a class="menu-text" href="barometer_sensor.php">Barometer</a>
        <a class="menu-text" href="proximity_sensor.php">Proximity</a>
  </div>
  </div>
  <a class="menu-text" href="queries.php">QUERIES</a>
</div>
```

```
<!-- Page content -->
<div class="content">
<form action="users/insert.php" method="post">
    <table>
        <tr>
            <th>Field</th>
            <th>Value</th>
        </tr>
        <tr>
            <td>Username:</td>
            <td><input type="text" name="username" required></td>
        </tr>
        <tr>
            <td>UserId:</td>
            <td><input type="text" name="userid" required></td>
        </tr>
        <tr>
            <td>Password:</td>
            <td><input type="password" name="password" required></td>
        </tr>
        <tr>
            <td>Email:</td>
            <td><input type="email" name="email" required></td>
        </tr>
        <tr>
```

```
            <td>First Name:</td>
            <td><input type="text" name="first_name" required></td>
        </tr>
        <tr>
            <td>Surname:</td>
            <td><input type="text" name="surname" required></td>
        </tr>
    </table>
    <input type="submit" value="Insert Record">
</form>
```

```
<form action="users/delete.php" method="post">
    <table>
        <tr>
            <th>Field</th>
            <th>Value</th>
        </tr>
        <tr>
            <td>Username:</td>
            <td><input type="text" name="username" required></td>
        </tr>
        <tr>
            <td>UserId:</td>
            <td><input type="text" name="userid" required></td>
        </tr>
        <tr>
            <td>Password:</td>
            <td><input type="password" name="password" required></td>
        </tr>
        <tr>
            <td>Email:</td>
            <td><input type="email" name="email" required></td>
        </tr>
        <tr>
            <td>First Name:</td>
            <td><input type="text" name="first_name" required></td>
        </tr>
        <tr>
            <td>Surname:</td>
            <td><input type="text" name="surname" required></td>
        </tr>
    </table>
    <input type="submit" value="Delete Record">
</form>
</div>
```

```
</body>
</html>
```

Αρχείο 10 : sensors.php

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Μετρήσεις Κινητών Συσκευών Χρηστών</title>
    <link rel="stylesheet" type="text/css" href="static/style.css">
    <link rel="icon" type="image/x-icon" href="media/sensor.png">
</head>
<body>

<!-- The sidebar -->
<div class="sidebar">
  <a class="menu-text" href="index.php">DATABASE INITIALIZATION</a>
  <a class="menu-text" href="users.php">USERS</a>
  <div class="dropdown">
  <a class="menu-text active" href="sensors.php">SENSORS</a>
  <!-- The dropdown content -->
  <div class="dropdown-content">
```

```
        <a class="menu-text" href="magnetometer_sensor.php">Magnetometer</a>
        <a class="menu-text" href="accelerometer_sensor.php">Accelerometer</a>
        <a class="menu-text" href="geolocation_sensor.php">Geolocation</a>
        <a class="menu-text" href="gyroscope_sensor.php">Gyroscope</a>
        <a class="menu-text" href="step-counter_sensor.php">Step-counter</a>
        <a class="menu-text" href="barometer_sensor.php">Barometer</a>
        <a class="menu-text" href="proximity_sensor.php">Proximity</a>
  </div>
  </div>
  <a class="menu-text" href="queries.php">QUERIES</a>
</div>
```

```
<!-- Page content -->
<div class="content">
    <h1 class="menu-text">Select the Desired Sensor<br></h1>
</div>

</body>
</html>
```

- **Περιγραφή: Το παρακάτω αρχείο είναι υπεύθυνο για την αρχικοποίηση της Βάσης Δεδομένων.**

Αρχείο : database_init.php

```php
<?php
include ('copy.php');
include ('connection.php');
function FinalRelations_create($db_conn){
    // **CREATE FINAL TABLES **
        // Σχέση Users
        $sql1 =
        "CREATE TABLE IF NOT EXISTS Users
        (id SERIAL PRIMARY KEY NOT NULL,
         userid VARCHAR(40),
         username VARCHAR(40),
         password VARCHAR(40),
         email VARCHAR(40),
         first_name VARCHAR(40),
         surname VARCHAR(40))";
        // Σχέση Magnetometer
        $sql2 =
        "CREATE TABLE IF NOT EXISTS Magnetometer
        (id SERIAL PRIMARY KEY NOT NULL,
         userid VARCHAR(40),
         magnetometer_x DOUBLE PRECISION,
         magnetometer_y DOUBLE PRECISION,
         magnetometer_z DOUBLE PRECISION,
         time_stamp timestamp)";

        // Σχέση Accelerometer
        $sql3 =
        "CREATE TABLE IF NOT EXISTS Accelerometer
        (id SERIAL PRIMARY KEY NOT NULL,
         userid VARCHAR(40),
         accelerometer_x DOUBLE PRECISION,
         accelerometer_y DOUBLE PRECISION,
```

```
 accelerometer_z DOUBLE PRECISION,
 time_stamp timestamp)";

// Σχέση Geolocation
$sql4 =
"CREATE TABLE IF NOT EXISTS Geolocation
(id SERIAL PRIMARY KEY NOT NULL,
 userid VARCHAR(40),
 longtitude DOUBLE PRECISION,
 latitude   DOUBLE PRECISION,
 altitude   DOUBLE PRECISION,
 time_stamp timestamp)";

// Σχέση Gyroscope
$sql5 =
"CREATE TABLE IF NOT EXISTS Gyroscope
(id SERIAL PRIMARY KEY NOT NULL,
 userid VARCHAR(40),
 gyroscope_x DOUBLE PRECISION,
 gyroscope_y DOUBLE PRECISION,
 gyroscope_z DOUBLE PRECISION,
 time_stamp timestamp)";

// Σχέση Step_counter
$sql6 =
"CREATE TABLE IF NOT EXISTS Step_counter
(id SERIAL PRIMARY KEY NOT NULL,
 userid VARCHAR(40),
 footsteps INT,
 time_stamp timestamp)";

// Σχέση Proximity
$sql7 =
"CREATE TABLE IF NOT EXISTS Proximity
(id SERIAL PRIMARY KEY NOT NULL,
 userid VARCHAR(40),
 proximity_boolean BOOLEAN,
 time_stamp timestamp)";

// Σχέση Barometer
$sql8 =
"CREATE TABLE IF NOT EXISTS Barometer
(id SERIAL PRIMARY KEY NOT NULL,
 userid VARCHAR(40),
 atm DOUBLE PRECISION,
```

```
 time_stamp timestamp)";
 // Συσχέτιση Magnetometer - Users
 $sql9 =
 "CREATE TABLE IF NOT EXISTS Magnetometer_measurement
 ( id_Users INTEGER,
   id_Magnetometer INTEGER,
   FOREIGN KEY (id_Users) REFERENCES Users(id),
   FOREIGN KEY (id_Magnetometer) REFERENCES Magnetometer(id))";
// Συσχέτιση Accelerometer - Users
 $sql10 =
 "CREATE TABLE IF NOT EXISTS Accelerometer_measurement
 ( id_Users INTEGER,
   id_Accelerometer INTEGER,
   FOREIGN KEY (id_Users) REFERENCES Users(id),
   FOREIGN KEY (id_Accelerometer) REFERENCES Accelerometer(id))";
 // Συσχέτιση Geolocation - Users
 $sql11 =
 "CREATE TABLE IF NOT EXISTS Geolocation_measurement
 ( id_Users INTEGER,
   id_Geolocation INTEGER,
   FOREIGN KEY (id_Users) REFERENCES Users(id),
   FOREIGN KEY (id_Geolocation) REFERENCES Geolocation(id))";
 // Συσχέτιση Gyroscope - Users
 $sql12 =
 "CREATE TABLE IF NOT EXISTS Gyroscope_measurement
 ( id_Users INTEGER,
   id_Gyroscope INTEGER,
   FOREIGN KEY (id_Users) REFERENCES Users(id),
   FOREIGN KEY (id_Gyroscope) REFERENCES Gyroscope(id))";
 // Συσχέτιση Step-counter - Users
 $sql13 =
 "CREATE TABLE IF NOT EXISTS Step_counter_measurement
 ( id_Users INTEGER,
   id_Step_counter INTEGER,
   FOREIGN KEY (id_Users) REFERENCES Users(id),
   FOREIGN KEY (id_Step_counter) REFERENCES Step_counter(id))";

 // Συσχέτιση Proximity - Users
 $sql14 =
 "CREATE TABLE IF NOT EXISTS Proximity_measurement
 ( id_Users INTEGER,
   id_Proximity INTEGER,
   FOREIGN KEY (id_Users) REFERENCES Users(id),
   FOREIGN KEY (id_Proximity) REFERENCES Proximity(id))";
 // Συσχέτιση Barometer - Users
```

```
         $sql15 =
         "CREATE TABLE IF NOT EXISTS Barometer_measurement
         (id_Users INTEGER,
          id_Barometer INTEGER,
          FOREIGN KEY (id_Users) REFERENCES Users(id),
          FOREIGN KEY (id_Barometer) REFERENCES Barometer(id))";
          //**Execution** of SQL queries
          $relations =
array($sql1,$sql2,$sql3,$sql4,$sql5,$sql6,$sql7,$sql8,$sql9,$sql11,$sql10,$sql12,
$sql13,$sql14,$sql15);
          for($i=0;$i<count($relations);$i++){
            $ret = pg_query($db_conn, $relations[$i]);
            echo "".$relations[$i];
            if(!$ret) {
            echo pg_last_error($db_conn);
            } else {
            // echo "Table created successfully\n";
            }
          }
    }

    function TmpRelations_create($db_conn)
    {
        // **CREATE TEMPORARY TABLES**
        // Σχέση Users_tmp
        $sqltmp1 =
        "CREATE TABLE IF NOT EXISTS Users_tmp
        ( userid VARCHAR(40),
          email VARCHAR(40),
          username VARCHAR(30),
          password VARCHAR(40),
          surname VARCHAR(40),
          first_name VARCHAR(40))";

        // Σχέση Geolocation_tmp
        $sqltmp2 =
        "CREATE TABLE IF NOT EXISTS Geolocation_tmp
        ( userid VARCHAR(40),
         time_stamp INT,
         latitude   DOUBLE PRECISION,
         longtitude DOUBLE PRECISION,
         altitude   DOUBLE PRECISION )";
        // Σχέση sensors_tmp
        $sqltmp3 =
        "CREATE TABLE IF NOT EXISTS sensors_tmp
```

```
        (userid VARCHAR(40),
         time_stamp INT,
         pressure       DOUBLE PRECISION,
         acceleration_x DOUBLE PRECISION,
         acceleration_y DOUBLE PRECISION,
         acceleration_z DOUBLE PRECISION,
         gyroscope_x    DOUBLE PRECISION,
         gyroscope_y    DOUBLE PRECISION,
         gyroscope_z    DOUBLE PRECISION,
         magnetometer_x DOUBLE PRECISION,
         magnetometer_y DOUBLE PRECISION,
         magnetometer_z DOUBLE PRECISION,
         proximity      BOOLEAN,
         steps          INT)";
        //**Execution** of SQL queries
        $TmpRelations = array($sqltmp1,$sqltmp2,$sqltmp3);
        for($i=0;$i<count($TmpRelations);$i++){
          $ret = pg_query($db_conn, $TmpRelations[$i]);
          if(!$ret) {
          echo pg_last_error($db_conn);
          } else {
        // echo "Table created successfully\n";
         }
       }
    copyRecordstoTmpTables($db_conn);
}
function Relations_init($db_conn){
 // **INSERT QUERIES FOR RELATIONS**
      $sqlinsert1 = "INSERT INTO Users (userid, username, password, email,
first_name, surname)
                    SELECT userid,username, password, email, first_name, surname
                    FROM Users_tmp";
      $sqlinsert2 = "INSERT INTO Geolocation (userid,
longtitude,latitude,altitude,time_stamp)
                    SELECT userid, longtitude, latitude, altitude,
TO_TIMESTAMP(time_stamp)::TIMESTAMP
                    FROM Geolocation_tmp";
      $sqlinsert3 = "INSERT INTO Magnetometer (userid,
magnetometer_x,magnetometer_y,magnetometer_z,time_stamp)
                    SELECT userid,
magnetometer_x,magnetometer_y,magnetometer_z,TO_TIMESTAMP(time_stamp)::TIMESTAMP
                    FROM sensors_tmp";
      $sqlinsert4 = "INSERT INTO Accelerometer (userid,
accelerometer_x,accelerometer_y,accelerometer_z,time_stamp)
```

```
                    SELECT userid,
acceleration_x,acceleration_y,acceleration_z,TO_TIMESTAMP(time_stamp)::TIMESTAMP
                    FROM sensors_tmp";
      $sqlinsert5 = "INSERT INTO Gyroscope (userid,
gyroscope_x,gyroscope_y,gyroscope_z,time_stamp)
                    SELECT userid,
gyroscope_x,gyroscope_y,gyroscope_z,TO_TIMESTAMP(time_stamp)::TIMESTAMP
                    FROM sensors_tmp";
      $sqlinsert6 = "INSERT INTO Step_counter (userid,footsteps,time_stamp)
                    SELECT userid,steps,TO_TIMESTAMP(time_stamp)::TIMESTAMP
                    FROM sensors_tmp";

      $sqlinsert7 = "INSERT INTO Proximity (userid,proximity_boolean,time_stamp)
                    SELECT userid,proximity,TO_TIMESTAMP(time_stamp)::TIMESTAMP
                    FROM sensors_tmp";
      $sqlinsert8 = "INSERT INTO Barometer (userid,atm,time_stamp)
                    SELECT userid,pressure,TO_TIMESTAMP(time_stamp)::TIMESTAMP
                    FROM sensors_tmp";
      $sqldrop1 = "DROP TABLE sensors_tmp";
      $sqldrop2 = "DROP TABLE Users_tmp";
      $sqldrop3 = "DROP TABLE Geolocation_tmp";
      //**Execution** of SQL queries
      $droptmptables = array($sqldrop1,$sqldrop2,$sqldrop3);
      $relationload =
array($sqlinsert1,$sqlinsert2,$sqlinsert3,$sqlinsert4,$sqlinsert5,$sqlinsert6,$sq
linsert7,$sqlinsert8);
      for($i=0;$i<count($relationload);$i++){
        $ret = pg_query($db_conn, $relationload[$i]);
        if(!$ret) {
        echo pg_last_error($db_conn);
        } else {
        //echo "Table created successfully\n";
        }
      }
     for($i=0;$i<count($droptmptables);$i++){
      $ret = pg_query($db_conn, $droptmptables[$i]);
      if(!$ret) {
      echo pg_last_error($db_conn);
      } else {
      //echo "Table created successfully\n";
      }
     }
    }
    function correlation_init($db_conn){
     // Extracts the corresponding id measurements(sensors) by userid
```

```
      $Idpairs1 = "SELECT Users.id AS id_u, Magnetometer.id AS id_m
                   FROM Users, Magnetometer
                   WHERE Users.userid = Magnetometer.userid";

      $Idpairs2 = "SELECT Users.id AS id_u, Barometer.id AS id_m
                   FROM Users, Barometer
                   WHERE Users.userid = Barometer.userid";
      $Idpairs3 = "SELECT Users.id AS id_u, Accelerometer.id AS id_m
                   FROM Users, Accelerometer
                   WHERE Users.userid = Accelerometer.userid";
      $Idpairs4 = "SELECT Users.id AS id_u, Proximity.id AS id_m
                   FROM Users, Proximity
                   WHERE Users.userid = Proximity.userid";
      $Idpairs5 = "SELECT Users.id AS id_u, Step_counter.id AS id_m
                   FROM Users, Step_counter
                   WHERE Users.userid = Step_counter.userid";
      $Idpairs6 = "SELECT Users.id AS id_u, Geolocation.id AS id_m
                   FROM Users, Geolocation
                   WHERE Users.userid = Geolocation.userid";
      $Idpairs7 = "SELECT Users.id AS id_u, Gyroscope.id AS id_m
                   FROM Users, Gyroscope
                   WHERE Users.userid = Gyroscope.userid";
      $correlationsrecs =
array($Idpairs1,$Idpairs2,$Idpairs3,$Idpairs4,$Idpairs5,$Idpairs6,$Idpairs7);

    for($i=0;$i<count($correlationsrecs);$i++){
    $result = pg_query($db_conn, $correlationsrecs[$i]);
    if(!$result) {
      echo pg_last_error($db_conn);
    } else {
      //echo "Table created successfully\n";
    }
    // Fetch the result row
    while ($row = pg_fetch_assoc($result)){
      $id_Users = $row['id_u'];
      $id_Measurement = $row['id_m'];
      // Check if values are not empty
    if (!empty($id_Users) && !empty($id_Measurement)) {
      $insertMeasurement1 = "INSERT INTO Magnetometer_measurement
(id_Users,id_Magnetometer)
                             VALUES ($id_Users, $id_Measurement)";

      $insertMeasurement2 = "INSERT INTO Barometer_measurement
(id_Users,id_Barometer)
                             VALUES ($id_Users, $id_Measurement)";
```

```
     $insertMeasurement3 = "INSERT INTO Accelerometer_measurement
(id_Users,id_Accelerometer)
                                VALUES ($id_Users, $id_Measurement)";
      $insertMeasurement4 = "INSERT INTO Proximity_measurement
(id_Users,id_Proximity)
                                VALUES ($id_Users, $id_Measurement)";
      $insertMeasurement5 = "INSERT INTO Step_counter_measurement
(id_Users,id_Step_counter)
                                VALUES ($id_Users, $id_Measurement)";
      $insertMeasurement6 = "INSERT INTO Geolocation_measurement
(id_Users,id_Geolocation)
                                VALUES ($id_Users, $id_Measurement)";
      $insertMeasurement7 = "INSERT INTO Gyroscope_measurement
(id_Users,id_Gyroscope)
                                VALUES ($id_Users, $id_Measurement)";
      $correlationsrecsins =
array($insertMeasurement1,$insertMeasurement2,$insertMeasurement3,$insertMeasurem
ent4,$insertMeasurement5,$insertMeasurement6,$insertMeasurement7);
      $insertResult = pg_query($db_conn, $correlationsrecsins[$i]);
      if (!$insertResult) {
          echo pg_last_error($db_conn);
      } else {
          //echo "Data inserted into Magnetometer_measurement successfully\n";
      }
    } else {
      // Handle empty values if needed
    }
    }
   }
  }
set_time_limit(100);
$db_conn = connection();
FinalRelations_create($db_conn);
TmpRelations_create($db_conn);
Relations_init($db_conn);
correlation_init($db_conn);
pg_close($dbConn);
// Redirect to the main page (index.php)
header("Location: index.php");
exit();
?>
```

- **Περιγραφή: Το παρακάτω αρχείο είναι υπεύθυνο για εισάγει τα δεδομένα απο τα .csv αρχεία στις προσωρινές σχέσεις.**

Αρχείο : copy.php

```php
<?php
function copyRecordstoTmpTables($db_conn)
{
// Array of Temporary Tables names
$tableNames = array('sensors_tmp','Geolocation_tmp','Users_tmp');
// Relative Path directing to the corresponding .csv files
$csvFilePaths = array('/home/Data/2023-24/sensorsF.csv','/home/Data/2023-
24/coordinatesF.csv','/home/Data/2023-24/users.csv');
// Connecting errors checking
if (!$db_conn) {
    die("Connection failed: " . pg_last_error());
}

for($i = 0;$i<count($csvFilePaths);$i++){
    // Read the CSV file into an array
    $rows = file($csvFilePaths[$i], FILE_IGNORE_NEW_LINES);
    // Skip the header row if present
    array_shift($rows);
    // Copying .csv files records inside the Temporary table
    pg_copy_from($db_conn, $tableNames[$i], $rows, ",","");
}
}
?>
```

- **Περιγραφή: Το παρακάτω αρχείο είναι υπεύθυνο για την σύνδεση με την βάση δεδομένων**

Αρχείο : connection.php

```php
<?php
function connection(){
    $conn = pg_connect("host=localhost dbname=db1u14 user=db1u14
password=ste4nV84");
    return $conn;
  }
?>
```

- **Περιγραφή: Τα παρακάτω αρχεία είναι υπεύθυνα για την λειτουργηκότητα της ιντερνετικής σελίδας.**

Αρχεία εισαγωγής/διαγραφής στην σχέση Users:

Αρχείο : insert.php

```
<?php
// Check if the form was submitted
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    // Retrieve variables from the form
    $userId = $_POST["userid"];
    $username = $_POST["username"];
    $email = $_POST["email"];
    $password = $_POST["password"];
    $first_name = $_POST["first_name"];
    $surname = $_POST["surname"];
    // Now you can use these variables as needed
    // For example, you can insert them into a database
[redacted]
    // Example database connection and insertion (replace with your actual
database logic)
    $dbConn = pg_connect("host=localhost dbname=db1u14 user=db1u14
password=ste4nV84");
[redacted]
  //Checking if the current ^ insertion is allready inside the Table
  //If this is the case then we ask the user to give input again
  $sql = 'SELECT * FROM Users';

   $rets = pg_query($dbConn, $sql);
   if(!$rets) {
     echo pg_last_error($dbConn);
     exit;
    }

   $cnt = 0; //Counting the fields that are the same with the fields of another
insertion
            //If all the fields are the same, then we ask the user to give input
again
   while($row = pg_fetch_row($rets)) {

        if($row[1] == $userId)
        $cnt++;

        if($row[2] == $username)
        $cnt++;

        if($row[4] == $email)
```

```
        $cnt++;

        if($cnt > 0) //Found a similar insertion
        break;
        // echo "This Insertion already exists! \n";
   }//while

  // if There is no other similar insertion then we add it to the Table
  if($cnt == 0) {
    $query = "INSERT INTO Users (userid, username, email,
password,first_name,surname)
              VALUES ('$userId', '$username', '$email', '$password',
'$first_name','$surname')";

    $result = pg_query($dbConn, $query);
    if ($result) {
       // echo "Record inserted successfully";
    } else {
       // echo "Error inserting record: " . pg_last_error($dbConn);
    }
       // Close the database connection
       pg_close($dbConn);
       header("Location: ../users.php");
       exit;
 }
 else
 {
     echo "This record compromises the functional dependencies";
 }

}
?>
```

Αρχείο : delete.php

```
<?php
// delete.php

// Check if the form was submitted
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    // Retrieve form values
    $userId = $_POST["userid"];
    $username = $_POST["username"];
    $email = $_POST["email"];
    $password = $_POST["password"];
    $first_name = $_POST["first_name"];
```

```
    $surname = $_POST["surname"];
[redacted]
    // Example database connection (replace with your actual connection code)
    $dbConn = pg_connect("host=localhost dbname=db1u14 user=db1u14
password=ste4nV84");
[redacted]
   // $user_id = 0;
   // $measure_id = 0;
    //**DELETE the correlations between the records that corresponds to the given
userid**
    $reccor = "SELECT Users.id, Accelerometer.id FROM Users, Accelerometer WHERE
Users.userid = Accelerometer.userid AND Users.userid = '$userId'";
      //$delreccord = "DELETE FROM Accelerometer_measurement WHERE id_Users =
".$user_id."AND id_Accelerometer =".$measure_id."";
[redacted]
    $reccor2 = "SELECT Users.id, Magnetometer.id FROM Users, Magnetometer WHERE
Users.userid = Magnetometer.userid AND Users.userid = '$userId'";
      //$delreccord2 = "DELETE FROM Magnetometer_measurement WHERE id_Users =
".$user_id."AND id_Magnetometer =".$measure_id."";
[redacted]
    $reccor3 = "SELECT Users.id, Geolocation.id FROM Users, Geolocation WHERE
Users.userid = Geolocation.userid AND Users.userid = '$userId'";
     // $delreccord3 = "DELETE FROM Geolocation_measurement WHERE id_Users =
".$user_id."AND id_Geolocation =".$measure_id."";
[redacted]
    $reccor4 = "SELECT Users.id, Gyroscope.id FROM Users, Gyroscope WHERE
Users.userid = Gyroscope.userid AND Users.userid = '$userId'";
     // $delreccord4 = "DELETE FROM Gyroscope_measurement WHERE id_Users =
".$user_id."AND id_Gyroscope =".$measure_id."";
[redacted]
    $reccor5 = "SELECT Users.id, Step_counter.id FROM Users, Step_counter WHERE
Users.userid = Step_counter.userid AND Users.userid = '$userId'";
     // $delreccord5 = "DELETE FROM Step_counter_measurement WHERE id_Users =
".$user_id."AND id_Step_counter =".$measure_id."";
[redacted]
    $reccor6 = "SELECT Users.id, Proximity.id FROM Users, Proximity WHERE
Users.userid = Proximity.userid AND Users.userid = '$userId'";
      //$delreccord6 = "DELETE FROM Proximity_measurement WHERE id_Users =
".$user_id."AND id_Proximity =".$measure_id."";
[redacted]
    $reccor7 = "SELECT Users.id, Barometer.id FROM Users, Barometer WHERE
Users.userid = Barometer.userid AND Users.userid = '$userId'";
     // $delreccord7 = "DELETE FROM Barometer_measurement WHERE id_Users =
".$user_id."AND id_Barometer =".$measure_id."";
```

```
    $relation =
array($reccor,$reccor2,$reccor3,$reccor4,$reccor5,$reccor6,$reccor7);
    for($i=0;$i<count($relation);$i++){
      $ret = pg_query($dbConn, $relation[$i]);
      while($row = pg_fetch_row($ret)) {
        $user_id = $row[0];
        $measure_id = $row[1];
        $delreccord = "DELETE FROM Accelerometer_measurement WHERE id_Users =
".$user_id."AND id_Accelerometer =".$measure_id."";
        $delreccord2 = "DELETE FROM Magnetometer_measurement WHERE id_Users =
".$user_id."AND id_Magnetometer =".$measure_id."";
        $delreccord3 = "DELETE FROM Geolocation_measurement WHERE id_Users =
".$user_id."AND id_Geolocation =".$measure_id."";
        $delreccord4 = "DELETE FROM Gyroscope_measurement WHERE id_Users =
".$user_id."AND id_Gyroscope =".$measure_id."";
        $delreccord5 = "DELETE FROM Step_counter_measurement WHERE id_Users =
".$user_id."AND id_Step_counter =".$measure_id."";
        $delreccord6 = "DELETE FROM Proximity_measurement WHERE id_Users =
".$user_id."AND id_Proximity =".$measure_id."";
        $delreccord7 = "DELETE FROM Barometer_measurement WHERE id_Users =
".$user_id."AND id_Barometer =".$measure_id."";
```

```
        echo "". $user_id."<br>".$measure_id.""."<br>";
```

```
        switch ($relation[$i]){

          case $reccor:
            $cor =pg_query($dbConn, $delreccord);
            if(!$cor) {
              echo pg_last_error($dbConn);
              } else {
              echo "Succesful deletion\n";
              }
            break;

          case $reccor2:
            $cor =pg_query($dbConn, $delreccord2);
            if(!$cor) {
              echo pg_last_error($dbConn);
              } else {
              echo "Succesful deletion\n";
              }
            break;
```

```
case $reccor3:
  $cor =pg_query($dbConn, $delreccord3);
  if(!$cor) {
    echo pg_last_error($dbConn);
    } else {
    echo "Succesful deletion\n";
    }
  break;
```

```
case $reccor4:
  $cor =pg_query($dbConn, $delreccord4);
  if(!$cor) {
    echo pg_last_error($dbConn);
    } else {
    echo "Succesful deletion\n";
    }
  break;

case $reccor5:
  $cor =pg_query($dbConn, $delreccord5);
  if(!$cor) {
    echo pg_last_error($dbConn);
    } else {
    echo "Succesful deletion\n";
    }
  break;

 case $reccor6:
  $cor =pg_query($dbConn, $delreccord6);
  if(!$cor) {
    echo pg_last_error($dbConn);
    } else {
    echo "Succesful deletion\n";
    }
  break;

 case $reccor7:
  $cor =pg_query($dbConn, $delreccord7);
  if(!$cor) {
    echo pg_last_error($dbConn);
    } else {
    echo "Succesful deletion\n";
    }
  break;
```

```
    }
    if(!$ret) {
    echo pg_last_error($dbConn);
    } else {
    echo "Table created successfully\n";
    }
  }
  }
```

```
  // **DELETE the records of the final tables the records that correspond to
the given userid**
  $query = "DELETE FROM Users
            WHERE userid = '$userId'
              AND username = '$username'
              AND email = '$email'
              AND password = '$password'
              AND first_name = '$first_name'
              AND surname = '$surname'" ;


  $query2 = "DELETE FROM Magnetometer
             WHERE userid = '$userId'";


  $query3 = "DELETE FROM Accelerometer
             WHERE userid = '$userId'";
```

```
  $query4 = "DELETE FROM Geolocation
             WHERE userid = '$userId'";
```

```
  $query5 = "DELETE FROM Gyroscope
             WHERE userid = '$userId'";


  $query6 = "DELETE FROM Step_counter
             WHERE userid = '$userId'";
```

```
  $query7 = "DELETE FROM Proximity
             WHERE userid = '$userId'";
```

```
  $query8 = "DELETE FROM Barometer
```

```
                WHERE userid = '$userId'";
```

```
     //**Execution** of SQL queries
     $relations =
array($query,$query2,$query3,$query4,$query5,$query6,$query7,$query8);
     for($i=0;$i<count($relations);$i++){
       $ret = pg_query($dbConn, $relations[$i]);
       echo "".$relations[$i];
       if(!$ret) {
       echo pg_last_error($dbConn);
       } else {
        echo "Table created successfully\n";
       }
     }
```

```
    // Close the database connection
    pg_close($dbConn);
    header("Location: ../users.php");
    exit;
}
?>
```

Αρχεία εισαγωγής/διαγραφής στην σχέση Step_counter:

Αρχείο : insert.php

```
<?php
// insert.php
```

```
// Check if the form was submitted
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    // Retrieve variables from the form
    $userId = $_POST["userid"];
    $footsteps = $_POST["footsteps"];
    $timestamp = $_POST["timestamp"];
```

```
    // Now you can use these variables as needed
    // For example, you can insert them into a database
```

```
    // Example database connection and insertion (replace with your actual
database logic)
```

```
    $dbConn = pg_connect("host=localhost dbname=db1u14 user=db1u14
password=ste4nV84");

    //**MULTITUDE**
    //Before we add the record inside the Desired sensor Table we must ensure
that the given userId exists inside Users Table for multitude reasons
    $multitude = "SELECT * FROM Users WHERE userid = '$userId'";

    $ret = pg_query($dbConn, $multitude);
    if(!$ret) {
    echo pg_last_error($dbConn);
    exit;
    }

    $multitudecnt = 0; //Counting the fields that are the same with the fields of
another insertion
          //If all the fields are the same, then we ask the user to give input
again
    while($row = pg_fetch_row($ret)) {
      $multitudecnt++;
    }


$sql = 'SELECT *
FROM Steps_counter';

if($multitude > 0){
$rets = pg_query($dbConn, $sql);
if(!$rets) {
echo pg_last_error($dbConn);
exit;
}

$duplicate = false;
$cnt = 0; //Counting the fields that are the same with the fields of another
insertion
      //If all the fields are the same, then we ask the user to give input again
while($row = pg_fetch_row($rets)) {

  if($row[1] == $userId)
  {
    $cnt++;
  }

  if($row[2] == $footsteps)
```

```
  {
    $cnt++;
  }
```

```
  if(strtotime($row[3]) == $timestamp)
  {
    $cnt++;
  }
```

```
if($cnt == 3) //Found a similar insertion
{
 // echo "This Insertion already exists! \n";
  break;
}
else
{
 $cnt = 0; //Reset for each repetetion
}
```

```
}//while
```

```
if($cnt == 0) // if There is no other similar insertion then we add it to the
Table
{
    $query = "INSERT INTO Step_counter (userid, footsteps, time_stamp)
              VALUES ('$userId', $footsteps,
TO_TIMESTAMP($timeStamp)::TIMESTAMP))";
```

```
    $result = pg_query($dbConn, $query);
```

```
    if ($result) {
        echo "Record inserted successfully";
    } else {
        echo "Error inserting record: " . pg_last_error($dbConn);
    }
```

```
    //**UPDATING THE CORRESPONDING CORRELATION**
    $Idpairs =  "SELECT Users.id AS id_u, Step_counter.id AS id_m
    FROM Users, Step_counter
    WHERE Users.userid = Step_counter.userid AND Users.usersid = '$userid' AND
Step_counter.userid = '$userid'";
```

```
    $result = pg_query($db_conn, $Idpairs);
```

```
    if(!$result) {
    echo pg_last_error($db_conn);
    } else {
    //echo "Table created successfully\n";
    }

    // Fetch the result row
    while ($row = pg_fetch_assoc($result)){
    $id_Users = $row['id_u'];
    $id_Measurement = $row['id_m'];

    // Check if values are not empty
    if (!empty($id_Users) && !empty($id_Measurement)) {
    $insertMeasurement = "INSERT INTO Step_counter_measurement
(id_Users,id_Step_counter)
                          VALUES ($id_Users, $id_Measurement)";

    $insertResult = pg_query($db_conn, $insertMeasurement);

    if (!$insertResult) {
        echo pg_last_error($db_conn);
    } else {
        //echo "Data inserted into Magnetometer_measurement successfully\n";
    }
    }
    }

    // Close the database connection
    pg_close($dbConn);
    header("Location: ../step-counter_sensor.php");
    exit;
}
else
   {
     echo "This record compromises the functional dependencies";
   }
}//multitude
else
{
  echo "No user with that userId";
}
}
?>
```

Αρχείο : delete.php

```php
<?php
// delete.php

// Check if the form was submitted
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    // Retrieve form values
    $userId = $_POST["userid"];
    $footsteps = $_POST["footsteps"];
    $timestamp = $_POST["timestamp"];

    // Example database connection (replace with your actual connection code)
    $dbConn = pg_connect("host=localhost dbname=db1u14 user=db1u14 password=ste4nV84");

    $reccor = "SELECT Users.id, Step_counter.id FROM Users, Step_counter WHERE Users.userid = Step_counter.userid AND Users.userid = '$userId'";

    $ret = pg_query($dbConn, $reccor);
    while($row = pg_fetch_row($ret)) {
        $user_id = $row[0];
        $measure_id = $row[1];
        $delreccord = "DELETE FROM Step_counter_measurement WHERE id_Users = ".$user_id."AND id_Step_counter =".$measure_id."";

        $cor =pg_query($dbConn, $delreccord);
        if(!$cor) {
          echo pg_last_error($dbConn);
        } else {
          echo "Succesful deletion\n";
        }
    }

    // Construct the DELETE statement
    $query = "DELETE FROM Steps_counter
              WHERE userid = '$userId'
                AND proximity_boolean = '$footsteps'
                AND gyroscope_y = TO_TIMESTAMP($timeStamp)::TIMESTAMP)";

    // Execute the DELETE statement
    $result = pg_query($dbConn, $query);

    if ($result) {
        echo "Records deleted successfully";
    } else {
```

```
        echo "Error deleting records: " . pg_last_error($dbConn);
    }

    // Close the database connection
    pg_close($dbConn);
    header("Locaiton: ../step-counter_sensor.php");
    exit;
}
?>
```

```
<?php
// insert.php

// Check if the form was submitted
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    // Retrieve variables from the form
    $userId = $_POST["userid"];
    $proximity_boolean = $_POST["proximity"];
    $timestamp = $_POST["timestamp"];

    // Now you can use these variables as needed
    // For example, you can insert them into a database

    // Example database connection and insertion (replace with your actual
database logic)
    $dbConn = pg_connect("host=localhost dbname=db1u14 user=db1u14
password=ste4nV84");

//**MULTITUDE**
    //Before we add the record inside the Desired sensor Table we must ensure
that the given userId exists inside Users Table for multitude reasons
    $multitude = "SELECT * FROM Users WHERE userid = '$userId'";

    $ret = pg_query($dbConn, $multitude);
    if(!$ret) {
    echo pg_last_error($dbConn);
    exit;
    }
```

```
    $multitudecnt = 0; //Counting the fields that are the same with the fields of another insertion
          //If all the fields are the same, then we ask the user to give input again
    while($row = pg_fetch_row($ret)) {
      $multitudecnt++;
    }
```

```
    $sql = 'SELECT *
FROM Proximity';
```

```
if($multitude > 0){
$rets = pg_query($dbConn, $sql);
if(!$rets) {
echo pg_last_error($dbConn);
exit;
}
```

```
$duplicate = false;
$cnt = 0; //Counting the fields that are the same with the fields of another insertion
      //If all the fields are the same, then we ask the user to give input again
while($row = pg_fetch_row($rets)) {
```

```
  if($row[1] == $userId)
  {
    $cnt++;
  }
```

```
  if($row[2] == $proximity_boolean)
  {
    $cnt++;
  }
```

```
  if(strtotime($row[3]) == $timestamp)
  {
    $cnt++;
  }
```

```
if($cnt == 3) //Found a similar insertion
{
 // echo "This Insertion already exists! \n";
```

```
  break;
}
else
{
 $cnt = 0; //Reset for each repetetion
}
[redacted]
}//while
[redacted]
if($cnt == 0) // if There is no other similar insertion then we add it to the
Table
{
[redacted]
    $query = "INSERT INTO Proximity (userid, proximity_boolean, time_stamp)
              VALUES ('$userId', '$proximity_boolean',
TO_TIMESTAMP($timeStamp)::TIMESTAMP))";
[redacted]
    $result = pg_query($dbConn, $query);
[redacted]
    if ($result) {
        echo "Record inserted successfully";
    } else {
        echo "Error inserting record: " . pg_last_error($dbConn);
    }
[redacted]
    //**UPDATING THE CORRESPONDING CORRELATION**
    $Idpairs =  "SELECT Users.id AS id_u, Proximity.id AS id_m
    FROM Users, Proximity
    WHERE Users.userid = Proximity.userid AND Users.usersid = '$userid' AND
Proximity.userid = '$userid'";
[redacted]
    $result = pg_query($db_conn, $Idpairs);
    if(!$result) {
    echo pg_last_error($db_conn);
    } else {
    //echo "Table created successfully\n";
    }
[redacted]
    // Fetch the result row
    while ($row = pg_fetch_assoc($result)){
    $id_Users = $row['id_u'];
    $id_Measurement = $row['id_m'];
```

```
    // Check if values are not empty
    if (!empty($id_Users) && !empty($id_Measurement)) {
    $insertMeasurement = "INSERT INTO Proximity_measurement
(id_Users,id_Proximity)
                          VALUES ($id_Users, $id_Measurement)";

    $insertResult = pg_query($db_conn, $insertMeasurement);

    if (!$insertResult) {
        echo pg_last_error($db_conn);
    } else {
        //echo "Data inserted into Magnetometer_measurement successfully\n";
    }
    }
    }

    // Close the database connection
    pg_close($dbConn);
    header("Locaiton: ../proximity_sensor.php");
    exit;
}
else
   {
     echo "This record compromises the functional dependencies";
   }
}//multitude
else
{
  echo "No user with that userId";
}
}
?>
```

Αρχείο : delete.php

```
<?php
// delete.php

// Check if the form was submitted
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    // Retrieve form values
    $userId = $_POST["userid"];
    $proximity_boolean = $_POST["proximity"];
    $timestamp = $_POST["timestamp"];
```

```
    // Example database connection (replace with your actual connection code)
    $dbConn = pg_connect("host=localhost dbname=db1u14 user=db1u14
password=ste4nV84");

    $reccor = "SELECT Users.id, Proximity.id FROM Users, Proximity WHERE
Users.userid = Proximity.userid AND Users.userid = '$userId'";

    $ret = pg_query($dbConn, $reccor);
    while($row = pg_fetch_row($ret)) {
        $user_id = $row[0];
        $measure_id = $row[1];
        $delreccord = "DELETE FROM Proximity_measurement WHERE id_Users =
".$user_id."AND id_Proximity =".$measure_id."";

        $cor =pg_query($dbConn, $delreccord);
        if(!$cor) {
          echo pg_last_error($dbConn);
        } else {
          echo "Succesful deletion\n";
        }
    }

    // Construct the DELETE statement
    $query = "DELETE FROM Proximity
              WHERE userid = '$userId'
                AND proximity_boolean = '$proximity_boolean'
                AND gyroscope_y = TO_TIMESTAMP($timeStamp)::TIMESTAMP)";

    // Execute the DELETE statement
    $result = pg_query($dbConn, $query);

    if ($result) {
        echo "Records deleted successfully";
    } else {
        echo "Error deleting records: " . pg_last_error($dbConn);
    }

    // Close the database connection
    pg_close($dbConn);
    header("Location: ../proximity_sensor.php");
    exit;
}
?>
```

Αρχεία εισαγωγής/διαγραφής στην σχέση Magnetometer

Αρχείο : insert.php

```
<?php
// insert.php

// Check if the form was submitted
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    // Retrieve variables from the form
    $userId = $_POST["userid"];
    $magnetometerX = $_POST["magnetometer_x"];
    $magnetometerY = $_POST["magnetometer_y"];
    $magnetometerZ = $_POST["magnetometer_z"];
    $timeStamp = $_POST["time_stamp"];

    // Now you can use these variables as needed
    // For example, you can insert them into a database

    // Example database connection and insertion (replace with your actual
database logic)
    $dbConn = pg_connect("host=localhost dbname=your_database user=your_user
password=your_password");

//Before we add the record inside the Desired sensor Table we must ensure that
the given userId exists inside Users Table for multitude reasons
$multitude = "SELECT * FROM Users WHERE userid = '$userId'";

$ret = pg_query($dbConn, $multitude);
if(!$ret) {
echo pg_last_error($dbConn);
exit;
}

$multitudecnt = 0; //Counting the fields that are the same with the fields of
another insertion
      //If all the fields are the same, then we ask the user to give input again
while($row = pg_fetch_row($ret)) {
  $multitudecnt++;
}

    $sql = 'SELECT *
FROM Magnetometer';
```

```
if($multitude > 0){
$rets = pg_query($dbConn, $sql);
if(!$rets) {
echo pg_last_error($dbConn);
exit;
}

$duplicate = false;
$cnt = 0; //Counting the fields that are the same with the fields of another
insertion
      //If all the fields are the same, then we ask the user to give input again
while($row = pg_fetch_row($rets)) {

  if($row[1] == $userId)
  {
    $cnt++;
  }

  if($row[2] == $magnetometerX)
  {
    $cnt++;
  }

  if($row[3] == $magnetometerY)
  {
    $cnt++;
  }

  if($row[4] == $magnetometerZ)
  {
    $cnt++;
  }

  if(strtotime($row[5]) == $timestamp)
  {
    $cnt++;
  }

if($cnt == 5) //Found a similar insertion
{
 // echo "This Insertion already exists! \n";
  break;
}
```

```
else
{
 $cnt = 0; //Reset for each repetetion
}
[redacted]
}//while
[redacted]
if($cnt == 0) // if There is no other similar insertion then we add it to the
Table
{
[redacted]
    $query = "INSERT INTO Magnetometer (userid, magnetometer_x, magnetometer_y,
magnetometer_z, time_stamp)
              VALUES ('$userId', CAST('$magnetometerX'AS double precision),
CAST('$magnetometerY'AS double precision), CAST('$magnetometerZ'AS double
precision), TO_TIMESTAMP($timeStamp)::TIMESTAMP))";
[redacted]
    $result = pg_query($dbConn, $query);
[redacted]
    if ($result) {
        echo "Record inserted successfully";
    } else {
        echo "Error inserting record: " . pg_last_error($dbConn);
    }
[redacted]
    //**UPDATING THE CORRESPONDING CORRELATION**
    $Idpairs =  "SELECT Users.id AS id_u, Magnetometer.id AS id_m
    FROM Users, Magnetometer
    WHERE Users.userid = Magnetometer.userid AND Users.usersid = '$userid' AND
Magnetometer.userid = '$userid'";
[redacted]
    $result = pg_query($db_conn, $Idpairs);
    if(!$result) {
    echo pg_last_error($db_conn);
    } else {
    //echo "Table created successfully\n";
    }
[redacted]
    // Fetch the result row
    while ($row = pg_fetch_assoc($result)){
    $id_Users = $row['id_u'];
    $id_Measurement = $row['id_m'];
```

```
    // Check if values are not empty
    if (!empty($id_Users) && !empty($id_Measurement)) {
    $insertMeasurement = "INSERT INTO Magnetometer_measurement
(id_Users,id_Magnetometer)
                          VALUES ($id_Users, $id_Measurement)";

    $insertResult = pg_query($db_conn, $insertMeasurement);

    if (!$insertResult) {
        echo pg_last_error($db_conn);
    } else {
        //echo "Data inserted into Magnetometer_measurement successfully\n";
    }
    }
    }

    // Close the database connection
    pg_close($dbConn);
    header("Location: ../magnetometer_sensor.php");
    exit;
}
else
   {
     echo "This record compromises the functional dependencies";
   }
}//multitude
else
{
  echo "No user with that userId";
}
}
?>
```

Αρχείο : delete.php

```
<?php
// delete.php

// Check if the form was submitted
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    // Retrieve form values
    $userId = $_POST["userid"];
    $magnetometerX = $_POST["magnetometer_x"];
    $magnetometerY = $_POST["magnetometer_y"];
```

```
    $magnetometerZ = $_POST["magnetometer_z"];
    $timeStamp = $_POST["time_stamp"];

    // Example database connection (replace with your actual connection code)
    $dbConn = pg_connect("host=localhost dbname=your_database user=your_user
password=your_password");

    $reccor = "SELECT Users.id, Magnetometer.id FROM Users, Magnetometer WHERE
Users.userid = Magnetometer.userid AND Users.userid = '$userId'";

    $ret = pg_query($dbConn, $reccor);
    while($row = pg_fetch_row($ret)) {
        $user_id = $row[0];
        $measure_id = $row[1];
        $delreccord = "DELETE FROM Magnetometer_measurement WHERE id_Users =
".$user_id."AND id_Magnetometer =".$measure_id."";

        $cor =pg_query($dbConn, $delreccord);
        if(!$cor) {
          echo pg_last_error($dbConn);
        } else {
          echo "Succesful deletion\n";
        }
    }

    // Construct the DELETE statement
    $query = "DELETE FROM Magnetometer
              WHERE userid = '$userId'
                AND magnetometer_x = CAST('$magnetometerX'AS double precision)
                AND magnetometer_y = CAST('$magnetometerY'AS double precision)
                AND magnetometer_z = CAST('$magnetometerZ'AS double precision)
                AND time_stamp = TO_TIMESTAMP($timeStamp)::TIMESTAMP)";

    // Execute the DELETE statement
    $result = pg_query($dbConn, $query);

    if ($result) {
        echo "Records deleted successfully";
    } else {
        echo "Error deleting records: " . pg_last_error($dbConn);
    }

    // Close the database connection
    pg_close($dbConn);
    header("Location: ../magnetometer_sensor.php");
```

```
    exit;
}
?>
```

Αρχεία εισαγωγής/διαγραφής στην σχέση Gyroscope:

Αρχείο : insert.php

```
<?php
// insert.php

// Check if the form was submitted
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    // Retrieve variables from the form
    $userId = $_POST["userid"];
    $gyroscopeX = $_POST["gyroscope_x"];
    $gyroscopeY = $_POST["gyroscope_y"];
    $gyroscopeZ = $_POST["gyroscope_z"];
    $timeStamp = $_POST["time_stamp"];

    // Now you can use these variables as needed
    // For example, you can insert them into a database

    // Example database connection and insertion (replace with your actual
database logic)
    $dbConn = pg_connect("host=localhost dbname=db1u14 user=db1u14
password=ste4nV84");

   //**MULTITUDE**
    //Before we add the record inside the Desired sensor Table we must ensure
that the given userId exists inside Users Table for multitude reasons
    $multitude = "SELECT * FROM Users WHERE userid = '$userId'";

    $ret = pg_query($dbConn, $multitude);
    if(!$ret) {
    echo pg_last_error($dbConn);
    exit;
    }

    $multitudecnt = 0; //Counting the fields that are the same with the fields of
another insertion
          //If all the fields are the same, then we ask the user to give input
again
    while($row = pg_fetch_row($ret)) {
      $multitudecnt++;
```

```
    }
    $sql = 'SELECT *
    FROM Gyroscope';

    if($multitude > 0){
    $rets = pg_query($dbConn, $sql);
    if(!$rets) {
    echo pg_last_error($dbConn);
    exit;
    }

    $duplicate = false;
    $cnt = 0; //Counting the fields that are the same with the fields of another
insertion
          //If all the fields are the same, then we ask the user to give input
again
    while($row = pg_fetch_row($rets)) {

      if($row[1] == $userId)
      {
        $cnt++;
      }

      if($row[2] == $gyroscopeX)
      {
        $cnt++;
      }

      if($row[3] == $gyroscopeY)
      {
        $cnt++;
      }

      if($row[4] == $gyroscopeZ)
      {
        $cnt++;
      }

      if(strtotime($row[5]) == $timeStamp)
      {
        $cnt++;
      }
```

```
    if($cnt == 5) //Found a similar insertion
    {
     // echo "This Insertion already exists! \n";
      break;
    }
    else
    {
     $cnt = 0; //Reset for each repetetion
    }

    }//while


    if($cnt == 0) // if There is no other similar insertion then we add it to the
Table
    {
[redacted]
    $query = "INSERT INTO Gyroscope (userid, gyroscope_x, gyroscope_y,
gyroscope_z, time_stamp)
              VALUES ('$userId', CAST('$gyroscopeX'AS double precision),
CAST('$gyroscopeY'AS double precision), CAST('$gyroscopeZ'AS double precision),
TO_TIMESTAMP($timeStamp)::TIMESTAMP))";
[redacted]
    $result = pg_query($dbConn, $query);
[redacted]
    if ($result) {
       // echo "Record inserted successfully";
    } else {
      //  echo "Error inserting record: " . pg_last_error($dbConn);
    }
[redacted]
     //**UPDATING THE CORRESPONDING CORRELATION**
     $Idpairs =  "SELECT Users.id AS id_u, Gyroscope.id AS id_m
     FROM Users, Gyroscope
     WHERE Users.userid = Gyroscope.userid AND Users.usersid = '$userid' AND
Gyroscope.userid = '$userid'";


    $result = pg_query($db_conn, $Idpairs);
    if(!$result) {
    echo pg_last_error($db_conn);
    } else {
    //echo "Table created successfully\n";
    }
```

```
    // Fetch the result row
    while ($row = pg_fetch_assoc($result)){
    $id_Users = $row['id_u'];
    $id_Measurement = $row['id_m'];

    // Check if values are not empty
    if (!empty($id_Users) && !empty($id_Measurement)) {
    $insertMeasurement = "INSERT INTO Gyroscope_measurement
(id_Users,id_Gyroscope)
                         VALUES ($id_Users, $id_Measurement)";

    $insertResult = pg_query($db_conn, $insertMeasurement);

    if (!$insertResult) {
        echo pg_last_error($db_conn);
    } else {
        //echo "Data inserted into Magnetometer_measurement successfully\n";
    }
    }
    }
```

```
   }
   else
   {
     echo "This record compromises the functional dependencies";
   }
  } //multitude
  else
{
  echo "No user with that userId";
}
    // Close the database connection
    pg_close($dbConn);
    header("Location: ../gyroscope_sensor.php");
    exit;
}
?>
```

Αρχείο : delete.php

```
<?php
// delete.php
```

```
// Check if the form was submitted
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    // Retrieve form values
```

```
    $userId = $_POST["userid"];
    $gyroscopeX = $_POST["gyroscope_x"];
    $gyroscopeY = $_POST["gyroscope_y"];
    $gyroscopeZ = $_POST["gyroscope_z"];
    $timeStamp = $_POST["time_stamp"];

    // Example database connection (replace with your actual connection code)
    $dbConn = pg_connect("host=localhost dbname=db1u14 user=db1u14
password=ste4nV84");

    $reccor = "SELECT Users.id, Gyroscope.id FROM Users, Gyroscope WHERE
Users.userid = Gyroscope.userid AND Users.userid = '$userId'";

    $ret = pg_query($dbConn, $reccor);
    while($row = pg_fetch_row($ret)) {
        $user_id = $row[0];
        $measure_id = $row[1];
        $delreccord = "DELETE FROM Gyroscope_measurement WHERE id_Users =
".$user_id."AND id_Gyroscope =".$measure_id."";

        $cor =pg_query($dbConn, $delreccord);
        if(!$cor) {
          echo pg_last_error($dbConn);
        } else {
          echo "Succesful deletion\n";
        }
    }

    // Construct the DELETE statement
    $query = "DELETE FROM Gyroscope
              WHERE userid = '$userId'
                AND gyroscope_x = CAST('$gyroscopeX'AS double precision)
                AND gyroscope_y = CAST('$gyroscopeY'AS double precision)
                AND gyroscope_z = CAST('$gyroscopeZ'AS double precision)
                AND time_stamp = TO_TIMESTAMP($timeStamp)::TIMESTAMP)";

    // Execute the DELETE statement
    $result = pg_query($dbConn, $query);

    if ($result) {
        echo "Records deleted successfully";
    } else {
        echo "Error deleting records: " . pg_last_error($dbConn);
    }
```

```
    // Close the database connection
    pg_close($dbConn);
    header("Location: ../gyroscope_sensor.php");
    exit;
}
?>
```

Αρχεία εισαγωγής/διαγραφής στην σχέση Geolocation:

Αρχείο : insert.php

```
<?php
// insert.php

// Check if the form was submitted
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    // Retrieve variables from the form
    $userId = $_POST["userid"];
    $longtitude = $_POST["longtitude"];
    $altitude = $_POST["altitude"];
    $altitude = $_POST["latitude"];
    $timestamp   = $_POST["time_stamp"];

    // Now you can use these variables as needed
    // For example, you can insert them into a database

    // Example database connection and insertion (replace with your actual
database logic)
    $dbConn = pg_connect("host=localhost dbname=db1u14 user=db1u14
password=ste4nV84");

    //**MULTITUDE**
    //Before we add the record inside the Desired sensor Table we must ensure
that the given userId exists inside Users Table for multitude reasons
    $multitude = "SELECT * FROM Users WHERE userid = '$userId'";

    $ret = pg_query($dbConn, $multitude);
    if(!$ret) {
    echo pg_last_error($dbConn);
    exit;
    }
```

```
    $multitudecnt = 0; //Counting the fields that are the same with the fields of
another insertion
          //If all the fields are the same, then we ask the user to give input
again
    while($row = pg_fetch_row($ret)) {
      $multitudecnt++;
    }
```

[redacted]

```
    $sql = 'SELECT *
    FROM Geolocation';
```

[redacted]

```
    if($multitude > 0){
    $rets = pg_query($dbConn, $sql);
    if(!$rets) {
    echo pg_last_error($dbConn);
    exit;
    }

    $duplicate = false;
    $cnt = 0; //Counting the fields that are the same with the fields of another
insertion
          //If all the fields are the same, then we ask the user to give input
again
    while($row = pg_fetch_row($rets)) {

      if($row[1] == $userId)
      {
        $cnt++;
      }

      if($row[2] == $longtitude)
      {
        $cnt++;
      }

      if($row[3] == $latitude)
      {
        $cnt++;
      }

      if($row[4] == $altitude)
      {
```

```
        $cnt++;
      }

      if(strtotime($row[5]) == $timestamp)
      {
        $cnt++;
      }

    if($cnt == 5) //Found a similar insertion
    {
     // echo "This Insertion already exists! \n";
      break;
    }
    else
    {
     $cnt = 0; //Reset for each repetetion
    }

    }//while


    if($cnt == 0) // if There is no other similar insertion then we add it to the Table
    {
[redacted]
    $query = "INSERT INTO Geolocation (userid,longtitude,altitude,latitude,time_stamp)
              VALUES ('$userId', CAST('$longtitude' AS double precision),CAST('$altitude' AS double precision),CAST('$latitude' AS double precision), TO_TIMESTAMP($timeStamp)::TIMESTAMP))";
[redacted]
    $result = pg_query($dbConn, $query);
[redacted]
    if ($result) {
        echo "Record inserted successfully";
    } else {
        echo "Error inserting record: " . pg_last_error($dbConn);
    }
[redacted]
    //**UPDATING THE CORRESPONDING CORRELATION**
    $Idpairs =  "SELECT Users.id AS id_u, Geolocation.id AS id_m
    FROM Users, Geolocation
    WHERE Users.userid = Geolocation.userid AND Users.usersid = '$userid' AND Geolocation.userid = '$userid'";
```

```
    $result = pg_query($db_conn, $Idpairs);
    if(!$result) {
    echo pg_last_error($db_conn);
    } else {
    //echo "Table created successfully\n";
    }

    // Fetch the result row
    while ($row = pg_fetch_assoc($result)){
    $id_Users = $row['id_u'];
    $id_Measurement = $row['id_m'];

    // Check if values are not empty
    if (!empty($id_Users) && !empty($id_Measurement)) {
    $insertMeasurement = "INSERT INTO Geolocation_measurement
(id_Users,id_Geolocation)
                    VALUES ($id_Users, $id_Measurement)";

    $insertResult = pg_query($db_conn, $insertMeasurement);

    if (!$insertResult) {
    echo pg_last_error($db_conn);
    } else {
    //echo "Data inserted into Magnetometer_measurement successfully\n";
    }
    }
    }

    // Close the database connection
    pg_close($dbConn);
    header("Locaiotn: ../geolocation_sensor.php");
    exit;
}
else
   {
     echo "This record compromises the functional dependencies";
   }
}
else
{
  echo "No user with that userId";
```

```
}
}
?>
```

Αρχείο : delete.php

```
<?php
// delete.php

// Check if the form was submitted
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    // Retrieve form values
    $userId = $_POST["userid"];
    $longtitude = $_POST["longtitude"];
    $altitude = $_POST["altitude"];
    $altitude = $_POST["latitude"];
    $timestamp   = $_POST["time_stamp"];

    // Example database connection (replace with your actual connection code)
    $dbConn = pg_connect("host=localhost dbname=db1u14 user=db1u14
password=ste4nV84");

    $reccor = "SELECT Users.id, Geolocation.id FROM Users, Geolocation WHERE
Users.userid = Geolocation.userid AND Users.userid = '$userId'";

    $ret = pg_query($dbConn, $reccor);
    while($row = pg_fetch_row($ret)) {
        $user_id = $row[0];
        $measure_id = $row[1];
        $delreccord = "DELETE FROM Geolocation_measurement WHERE id_Users =
".$user_id."AND id_Geolocation =".$measure_id."";

        $cor =pg_query($dbConn, $delreccord);
        if(!$cor) {
          echo pg_last_error($dbConn);
        } else {
          echo "Succesful deletion\n";
        }
    }

    // Construct the DELETE statement
    $query = "DELETE FROM Geolocation
              WHERE userid = '$userId'
                AND longtitude = $longtitude
                AND latitude = $latitude
                AND altitude = $altitude
```

```
                AND time_stamp = TO_TIMESTAMP($timeStamp)::TIMESTAMP)";

    // Execute the DELETE statement
    $result = pg_query($dbConn, $query);

    if ($result) {
        echo "Records deleted successfully";
    } else {
        echo "Error deleting records: " . pg_last_error($dbConn);
    }

    // Close the database connection
    pg_close($dbConn);
    header("Location: ../geolocation_sensor.php");
    exit;
}
?>
```

Αρχεία εισαγωγής/διαγραφής στην σχέση Barometer:

Αρχείο : insert.php

```
<?php
// insert.php

// Check if the form was submitted
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    // Retrieve variables from the form
    $userId = $_POST["userid"];
    $pressure = $_POST["pressure"];
    $timestamp = $_POST["time_stamp"];

    // Now you can use these variables as needed
    // For example, you can insert them into a database

    // Example database connection and insertion (replace with your actual
database logic)
    $dbConn = pg_connect("host=localhost dbname=db1u14 user=db1u14
password=ste4nV84");
```

```
//Before we add the record inside the Desired sensor Table we must ensure that
the given userId exists inside Users Table for multitude reasons
$multitude = "SELECT * FROM Users WHERE userid = '$userId'";
[redacted]
$ret = pg_query($dbConn, $multitude);
if(!$ret) {
echo pg_last_error($dbConn);
exit;
}
[redacted]
$multitudecnt = 0; //Counting the fields that are the same with the fields of
another insertion
      //If all the fields are the same, then we ask the user to give input again
while($row = pg_fetch_row($ret)) {
  $multitudecnt++;
}
[redacted]
    $sql = 'SELECT *
            FROM Barometer';
[redacted]
if($multitude > 0){
[redacted]
$rets = pg_query($dbConn, $sql);
if(!$rets) {
echo pg_last_error($dbConn);
exit;
}
[redacted]
$duplicate = false;
$cnt = 0; //Counting the fields that are the same with the fields of another
insertion
      //If all the fields are the same, then we ask the user to give input again
while($row = pg_fetch_row($rets)) {
[redacted]
  if($row[1] == $userId)
  {
    $cnt++;
  }
[redacted]
  if($row[2] == $pressure)
  {
    $cnt++;
  }
[redacted]
  if(strtotime($row[3]) == $timestamp)
```

```
  {
    $cnt++;
  }
```

```
if($cnt == 3) //Found a similar insertion
{
 // echo "This Insertion already exists! \n";
  break;
}
else
{
 $cnt = 0; //Reset for each repetetion
}
```

```
}//while
```

```
if($cnt == 0) // if There is no other similar insertion then we add it to the
Table
{
```

```
    $query = "INSERT INTO Barometer (userid, pressure, time_stamp)
              VALUES ('$userId', $pressure,
TO_TIMESTAMP($timeStamp)::TIMESTAMP))";
```

```
    $result = pg_query($dbConn, $query);
```

```
    if ($result) {
        echo "Record inserted successfully";
    } else {
        echo "Error inserting record: " . pg_last_error($dbConn);
    }
```

```
      //**UPDATING THE CORRESPONDING CORRELATION**
    $Idpairs =  "SELECT Users.id AS id_u, Barometer.id AS id_m
    FROM Users, Barometer
    WHERE Users.userid = Barometer.userid AND Users.usersid = '$userid' AND
Barometer.userid = '$userid'";
```

```
$result = pg_query($db_conn, $Idpairs);
if(!$result) {
echo pg_last_error($db_conn);
} else {
```

```
//echo "Table created successfully\n";
}

// Fetch the result row
while ($row = pg_fetch_assoc($result)){
$id_Users = $row['id_u'];
$id_Measurement = $row['id_m'];

// Check if values are not empty
if (!empty($id_Users) && !empty($id_Measurement)) {
$insertMeasurement = "INSERT INTO Barometer_measurement (id_Users,id_Barometer)
                    VALUES ($id_Users, $id_Measurement)";

$insertResult = pg_query($db_conn, $insertMeasurement);

if (!$insertResult) {
    echo pg_last_error($db_conn);
} else {
    //echo "Data inserted into Magnetometer_measurement successfully\n";
}
}
}

    // Close the database connection
    pg_close($dbConn);
    header("Location: ../barometer_sensor.php");
    exit;
}
else
   {
     echo "This record compromises the functional dependencies";
   }
}
else
{
  echo "No user with that userId";
}
}
?>
```

Αρχείο : delete.php

```
<?php
// delete.php

// Check if the form was submitted
```

```php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    // Retrieve form values
    $userId = $_POST["userid"];
    $pressure = $_POST["pressure"];
    $timestamp = $_POST["time_stamp"];

    // Example database connection (replace with your actual connection code)
    $dbConn = pg_connect("host=localhost dbname=db1u14 user=db1u14
password=ste4nV84");

    $reccor = "SELECT Users.id, Barometer.id FROM Users, Barometer WHERE
Users.userid = Barometer.userid AND Users.userid = '$userId'";

    $ret = pg_query($dbConn, $reccor);
    while($row = pg_fetch_row($ret)) {
        $user_id = $row[0];
        $measure_id = $row[1];
        $delreccord = "DELETE FROM Barometer_measurement WHERE id_Users =
".$user_id."AND id_Barometer =".$measure_id."";

        $cor =pg_query($dbConn, $delreccord);
        if(!$cor) {
          echo pg_last_error($dbConn);
        } else {
          echo "Succesful deletion\n";
        }
    }

    // Construct the DELETE statement
    $query = "DELETE FROM Barometer
              WHERE userid = '$userId'
                AND pressure = $pressure
                AND time_stamp = TO_TIMESTAMP($timeStamp)::TIMESTAMP)" ;

    // Execute the DELETE statement
    $result = pg_query($dbConn, $query);

    if ($result) {
        echo "Records deleted successfully";
    } else {
        echo "Error deleting records: " . pg_last_error($dbConn);
    }

    // Close the database connection
    pg_close($dbConn);
```

```
    header("Location: ../barometer_sensor.php");
    exit;
}
?>
```

Αρχεία εισαγωγής/διαγραφής στην σχέση Accelerometer:

Αρχείο : insert.php

```
<?php
// insert.php

// Check if the form was submitted
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    // Retrieve variables from the form
    $userId = $_POST["userid"];
    $accelerometerX = $_POST["accelerometer_x"];
    $accelerometerY = $_POST["accelerometer_y"];
    $accelerometerZ = $_POST["accelerometer_z"];
    $timeStamp = $_POST["time_stamp"];

    // Now you can use these variables as needed
    // For example, you can insert them into a database
   // echo "".$userId;
    // Example database connection and insertion (replace with your actual
database logic)
    $dbConn = pg_connect("host=localhost dbname=db1u14 user=db1u14
password=ste4nV84");

    //Before we add the record inside the Desired sensor Table we must ensure
that the given userId exists inside Users Table for multitude reasons
    $multitude = "SELECT * FROM Users WHERE userid = '$userId'";

    $ret = pg_query($dbConn, $multitude);
    if(!$ret) {
    echo pg_last_error($dbConn);
    exit;
    }

    $multitudecnt = 0; //Counting the fields that are the same with the fields of
another insertion
          //If all the fields are the same, then we ask the user to give input
again
    if($row = pg_fetch_row($ret)) {
      $multitudecnt++;
```

```
      //echo "<tr><td>" . $row[1] ;
    }
```

```
    $sql = 'SELECT *
            FROM Accelerometer';
```

```
 if($multitudecnt > 0){
    $rets = pg_query($dbConn, $sql);
    if(!$rets) {
    echo pg_last_error($dbConn);
    exit;
    }

    $duplicate = false;
    $cnt = 0; //Counting the fields that are the same with the fields of another
insertion
          //If all the fields are the same, then we ask the user to give input
again
    while($row = pg_fetch_row($rets)) {

      if($row[1] == $userId)
      {
        $cnt++;
      }

      if($row[2] == $accelerometerX)
      {
        $cnt++;
      }

      if($row[3] == $accelerometerY)
      {
        $cnt++;
      }

      if($row[4] == $accelerometerZ)
      {
        $cnt++;
      }

      if(strtotime($row[5]) == $timeStamp)
      {
        $cnt++;
      }
```

```
    if($cnt == 5) //Found a similar insertion
    {
     // echo "This Insertion already exists! \n";
      break;
    }
    else
    {
     $cnt = 0; //Reset for each repetetion
    }

    }//while


    if($cnt == 0) // if There is no other similar insertion then we add it to the
Table
    {
     echo "".$cnt;
[redacted]
    $query = "INSERT INTO Accelerometer (userid, accelerometer_x,
accelerometer_y, accelerometer_z, time_stamp)
              VALUES ('$userId', CAST('$accelerometerX' AS double precision),
CAST('$accelerometerY' AS double precision), CAST('$accelerometerZ' AS double
precision), TO_TIMESTAMP($timeStamp)::TIMESTAMP)";
[redacted]
    $result = pg_query($dbConn, $query);
[redacted]
    if ($result) {
        echo "Record inserted successfully";
    } else {
        echo "Error inserting record: " . pg_last_error($dbConn);
    }
[redacted]
    //**UPDATING THE CORRESPONDING CORRELATION**
    $Idpairs =  "SELECT Users.id AS id_u, Accelerometer.id AS id_m
                 FROM Users, Accelerometer
                 WHERE Users.userid = Accelerometer.userid AND Users.userid =
'$userId' AND Accelerometer.userid = '$userId'";
[redacted]

    $result = pg_query($dbConn, $Idpairs);
    if(!$result) {
      echo pg_last_error($dbConn);
    } else {
```

```
      //echo "Table created successfully\n";
    }

    // Fetch the result row
    while ($row = pg_fetch_assoc($result)){
      $id_Users = $row['id_u'];
      $id_Measurement = $row['id_m'];

      // Check if values are not empty
    if (!empty($id_Users) && !empty($id_Measurement)) {
      $insertMeasurement = "INSERT INTO Accelerometer_measurement
(id_Users,id_Accelerometer)
                                VALUES ($id_Users, $id_Measurement)";

      $insertResult = pg_query($dbConn, $insertMeasurement);

      if (!$insertResult) {
          echo pg_last_error($dbConn);
      } else {
          //echo "Data inserted into Magnetometer_measurement successfully\n";
      }
    }
   }

    // Close the database connection
    pg_close($dbConn);
   // header("Location: ../accelerometer_sensor.php");
    exit;
 }
 else
   {
     echo "This record compromises the functional dependencies";
   }

}//if multitude
else
{
  echo "No user with that userId";
}
}
?>
```

Αρχείο : delete.php

```
<?php
// delete.php
```

```
// Check if the form was submitted
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    // Retrieve form values
    $userId = $_POST["userid"];
    $accelerometerX = $_POST["accelerometer_x"];
    $accelerometerY = $_POST["accelerometer_y"];
    $accelerometerZ = $_POST["accelerometer_z"];
    $timeStamp = $_POST["time_stamp"];

    // Example database connection (replace with your actual connection code)
    $dbConn = pg_connect("host=localhost dbname=db1u14 user=db1u14
password=ste4nV84");

    $reccor = "SELECT Users.id, Accelerometer.id FROM Users, Accelerometer WHERE
Users.userid = Accelerometer.userid AND Users.userid = '$userId'";

    $ret = pg_query($dbConn, $reccor);
    while($row = pg_fetch_row($ret)) {
        $user_id = $row[0];
        $measure_id = $row[1];
        $delreccord = "DELETE FROM Accelerometer_measurement WHERE id_Users =
".$user_id."AND id_Accelerometer =".$measure_id."";

        $cor =pg_query($dbConn, $delreccord);
        if(!$cor) {
          echo pg_last_error($dbConn);
        } else {
          echo "Succesful deletion\n";
        }
    }

    // Construct the DELETE statement
    $query = "DELETE FROM Accelerometer
              WHERE userid = '$userId'
                AND accelerometer_x = CAST('$accelerometerX' AS double precision)
                AND accelerometer_y = CAST('$accelerometerY' AS double precision)
                AND accelerometer_z = CAST('$accelerometerZ' AS double precision)
                AND timestamp = TO_TIMESTAMP($timeStamp)::TIMESTAMP)";

    // Execute the DELETE statement
```

```
    $result = pg_query($dbConn, $query);

    if ($result) {
        echo "Records deleted successfully";
    } else {
        echo "Error deleting records: " . pg_last_error($dbConn);
    }

    // Close the database connection
    pg_close($dbConn);
    header("accelerometer_sensor.php");
}
?>
```

Αρχείο για το CSS style:

Αρχείο : style.css

```
/* general container */
.background-container {
    background-color: black;
    color: white; /* Set text color to white or another contrasting color */
    padding: 20px; /* Adjust padding as needed */
}

/* The side navigation menu (used: universally)*/

.sidebar {
margin: 0;
padding: 0;
width: 200px;
background-color: #f1f1f1;
position: fixed;
height: 100%;
overflow: auto;
}

/* Sidebar links */
.sidebar a {
display: block;
color: black;
padding: 16px;
text-decoration: none;
```

```
}

/* Active/current link */
.sidebar a.active {
background-color: #04AA6D;
color: white;
}

/* Links on mouse-over */
.sidebar a:hover:not(.active) {
background-color: #555;
color: white;
}

/* Page content. The value of the margin-left property should match the value of
the sidebar's width property */
div.content {
margin-left: 200px;
padding: 1px 16px;
height: 1000px;
}

.menu-text{

font-family: 'Impact';

}

/*Hoverable Dropdown*/
/* Style for the dropdown container */
/* Style for the dropdown container */
.dropdown {
position: relative;
display: inline-block;
}

/* Style for the dropdown content (hidden by default) */
.dropdown-content {
display: none;
position: absolute;
background-color: #f9f9f9;
min-width: 160px;
box-shadow: 0 8px 16px rgba(0, 0, 0, 0.2);
z-index: 1;
}
```

```
/* Style for the dropdown links */
.dropdown-content a {
color: black;
padding: 12px 16px;
display: block;
text-decoration: none;
}
```

```
/* Change color on hover */
.dropdown-content a:hover {
background-color: #ddd;
}
```

```
/* Show the dropdown content when hovering over the dropdown container */
.dropdown:hover .dropdown-content {
display: block;
}
```

```
/* Index.php CONTENTS */
/*Index.php BUTTON for the DATABASES INITIALISATION */
.content {
display: flex;
justify-content: center;
align-items: center;
flex-grow: 1;
transition: background-color 0.3s ease;
max-height: 100vh;
}
```

```
.content button {
padding: 15px 20px;
font-size: 16px;
color: white;
background-color: #04AA6D;
}
```

```
.content button:active {
background-color: #555; /* Change to the desired background color when pressed */
}
```

```
/* sensors.php CONTENTS */
/*In this part it takes place the users functionality for Insertion/Deletion*/
table {
border-collapse: collapse;
```

```
width: 50%;
margin: 20px auto;
}

table, th, td {
border: 1px solid black;
padding: 10px;
}

form {
text-align: center;
margin: 20px auto;
}

/* Table Stylish */

th {
background-color: #04AA6D;
}

/* RESPONSIVENESS */

/* On screens that are less than 700px wide, make the sidebar into a topbar */
@media screen and (max-width: 700px) {
.sidebar {
width: 100%;
height: auto;
position: relative;
}
.sidebar a {float: left;}
div.content {margin-left: 0;}
}

/* On screens that are less than 400px, display the bar vertically, instead of
horizontally */
@media screen and (max-width: 400px) {
.sidebar a {
text-align: center;
float: none;
}
}
```