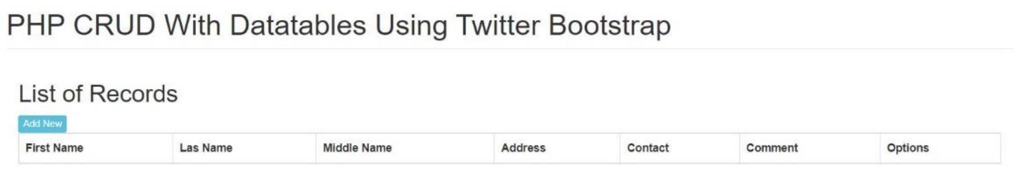
Maintenance Project

- **You need to enhance a program that performs CRUD (create, read, update, delete) operations. The user wants to keep the deleted and modified records in the database and to be accessible when needed, as in the banking system or social media. You need to modify this program to accommodate the new requirements.**

- I have used the PHP framework to implement the new requirement in existing code that performs CRUD operations.
- The existing code reference: https://sourcecodehero.com/crud-operations-in-php-with-source-code/
- The modification has been done in the existing code and I have attached the code snippet at the end of this question.
- Here, I have attached a screenshot of the step-by-step process.

1. As you can see in UI, we do not have any record.



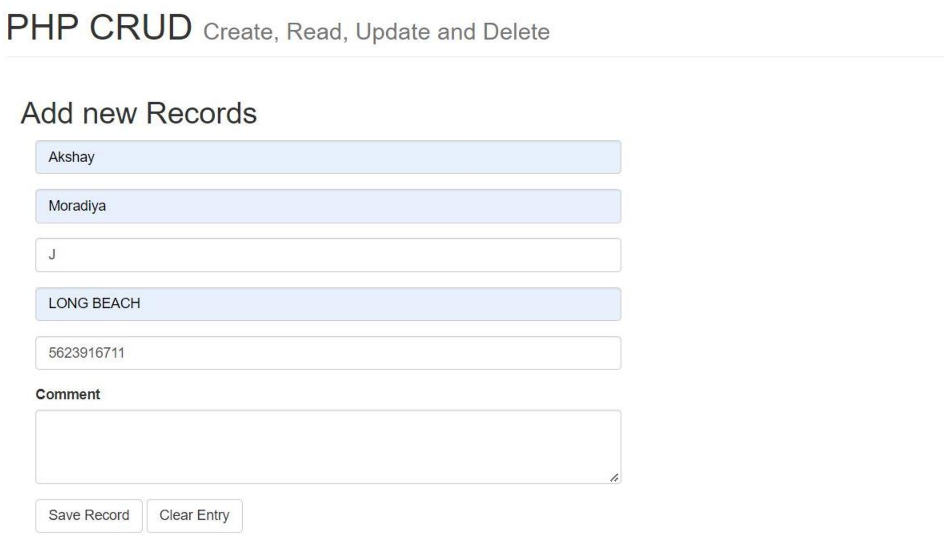
PHP CRUD With Datatables Using Twitter Bootstrap

List of Records

Add New

First Name	Las Name	Middle Name	Address	Contact	Comment	Options

2. Let's add one record by pressing "Add New" button and save the record.



PHP CRUD Create, Read, Update and Delete

Add new Records

Akshay

Moradiya

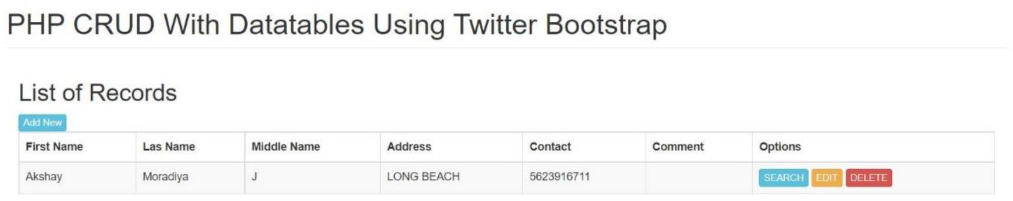
J

LONG BEACH

5623916711

Comment

3. As you can see in the screenshot, the record has been added.



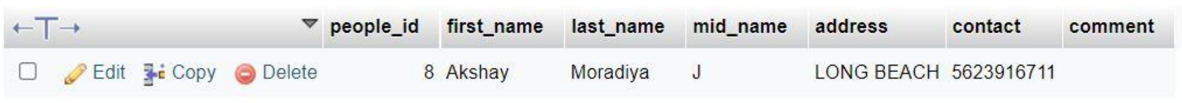
PHP CRUD With Datatables Using Twitter Bootstrap

List of Records

Add New

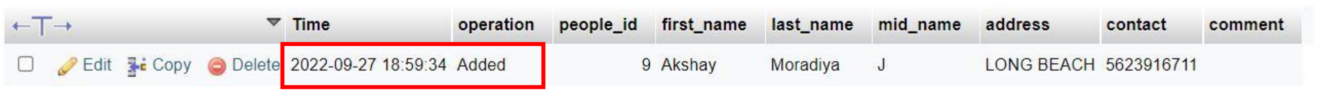
First Name	Las Name	Middle Name	Address	Contact	Comment	Options
Akshay	Moradiya	J	LONG BEACH	5623916711		SEARCH EDIT DELETE

4. The following screenshot shows the person table which has the latest data



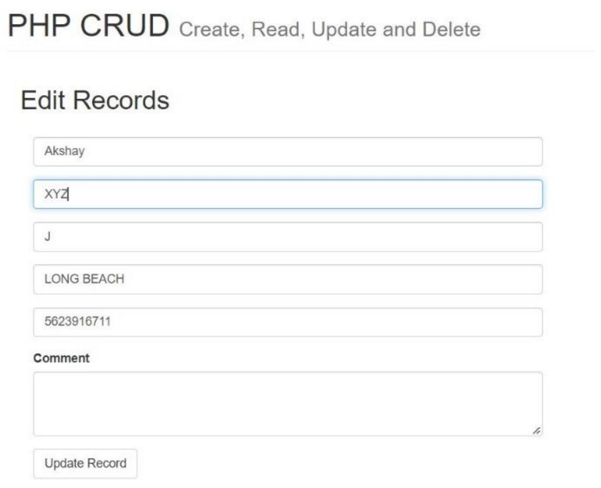
	people_id	first_name	last_name	mid_name	address	contact	comment
Edit Copy Delete	8	Akshay	Moradiya	J	LONG BEACH	5623916711	

5. The following table is a history table that has the history of the operation. You will find an additional two columns name time and operation that states the following operation is done with the timestamp.



	Time	operation	people_id	first_name	last_name	mid_name	address	contact	comment
Edit Copy Delete	2022-09-27 18:59:34	Added	9	Akshay	Moradiya	J	LONG BEACH	5623916711	

6. Let us edit the existing record. I am changing my last name to “XYZ”



PHP CRUD Create, Read, Update and Delete

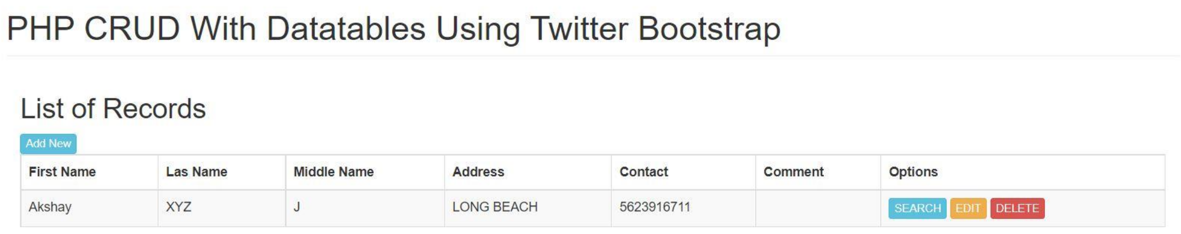
Edit Records

Akshay
XYZ
J
LONG BEACH
5623916711

Comment

Update Record

7. As you can see, on the UI the change has been seen



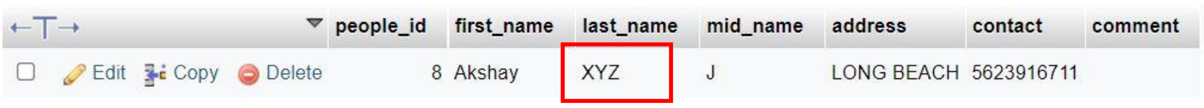
PHP CRUD With Datatables Using Twitter Bootstrap

List of Records

Add New

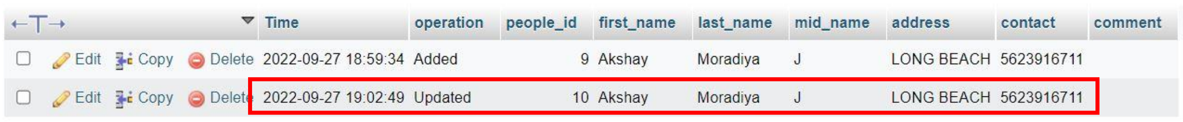
First Name	Las Name	Middle Name	Address	Contact	Comment	Options
Akshay	XYZ	J	LONG BEACH	5623916711		SEARCH EDIT DELETE

8. In person table, the change can be seen.



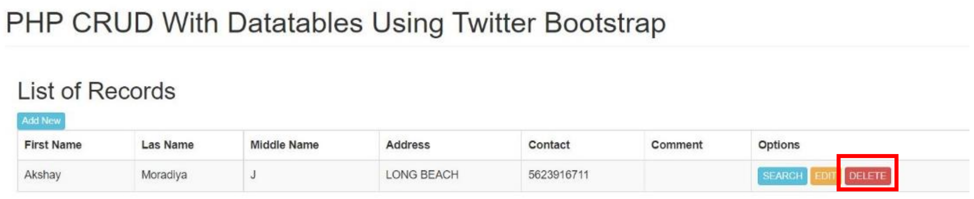
←T→	people_id	first_name	last_name	mid_name	address	contact	comment
☐ Edit Copy Delete	8	Akshay	XYZ	J	LONG BEACH	5623916711	

9. Now, let us look at the history table. It should contain a history of all operations



←T→	Time	operation	people_id	first_name	last_name	mid_name	address	contact	comment
☐ Edit Copy Delete	2022-09-27 18:59:34	Added	9	Akshay	Moradiya	J	LONG BEACH	5623916711	
☐ Edit Copy Delete	2022-09-27 19:02:49	Updated	10	Akshay	Moradiya	J	LONG BEACH	5623916711	

10. Now, let us perform the delete operation.



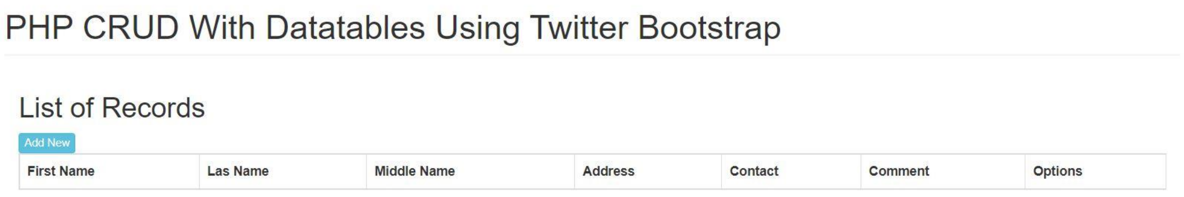
PHP CRUD With Datatables Using Twitter Bootstrap

List of Records

Add New

First Name	Las Name	Middle Name	Address	Contact	Comment	Options
Akshay	Moradiya	J	LONG BEACH	5623916711		SEARCH EDIT DELETE

11. As can be seen in image, the record has been deleted



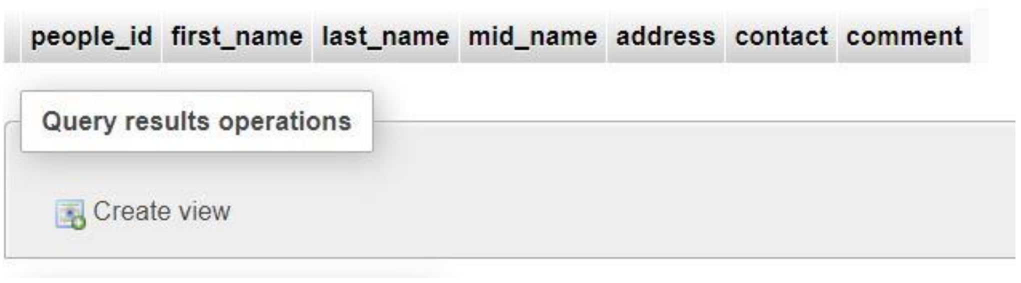
PHP CRUD With Datatables Using Twitter Bootstrap

List of Records

Add New

First Name	Las Name	Middle Name	Address	Contact	Comment	Options

12. Person table has nothing as we deleted the record

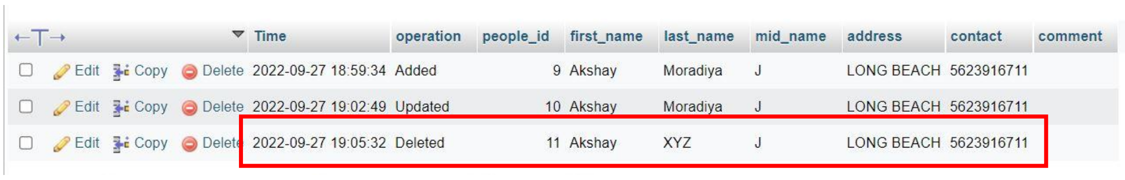


people_id	first_name	last_name	mid_name	address	contact	comment

Query results operations

Create view

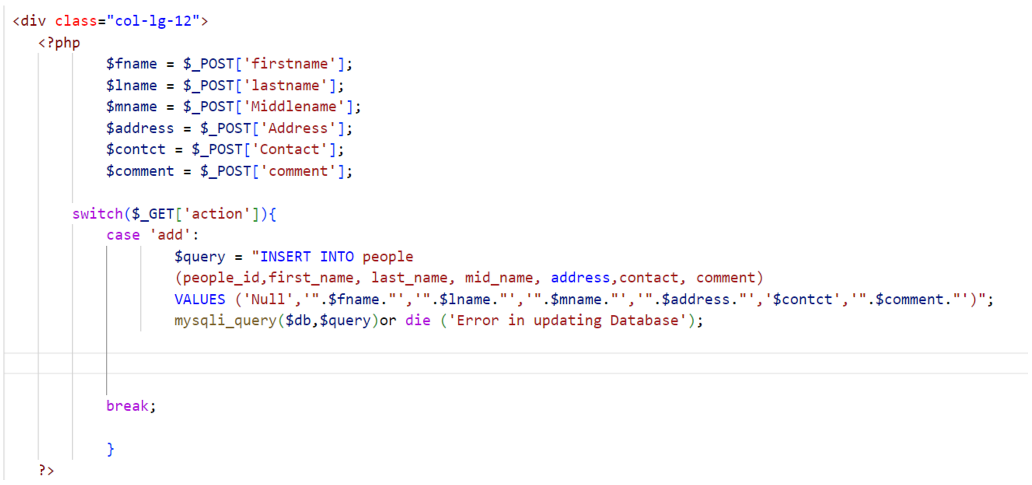
13. Let us look at the history table, it should have the history of deleted record



←T→	Time	operation	people_id	first_name	last_name	mid_name	address	contact	comment
☐ Edit Copy Delete	2022-09-27 18:59:34	Added	9	Akshay	Moradiya	J	LONG BEACH	5623916711	
☐ Edit Copy Delete	2022-09-27 19:02:49	Updated	10	Akshay	Moradiya	J	LONG BEACH	5623916711	
☐ Edit Copy Delete	2022-09-27 19:05:32	Deleted	11	Akshay	XYZ	J	LONG BEACH	5623916711	

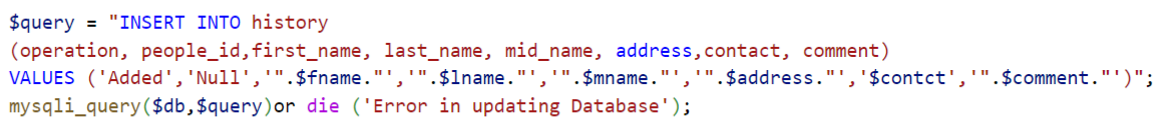
- Code snippet for each operation.

1. Adding a new record in person table.



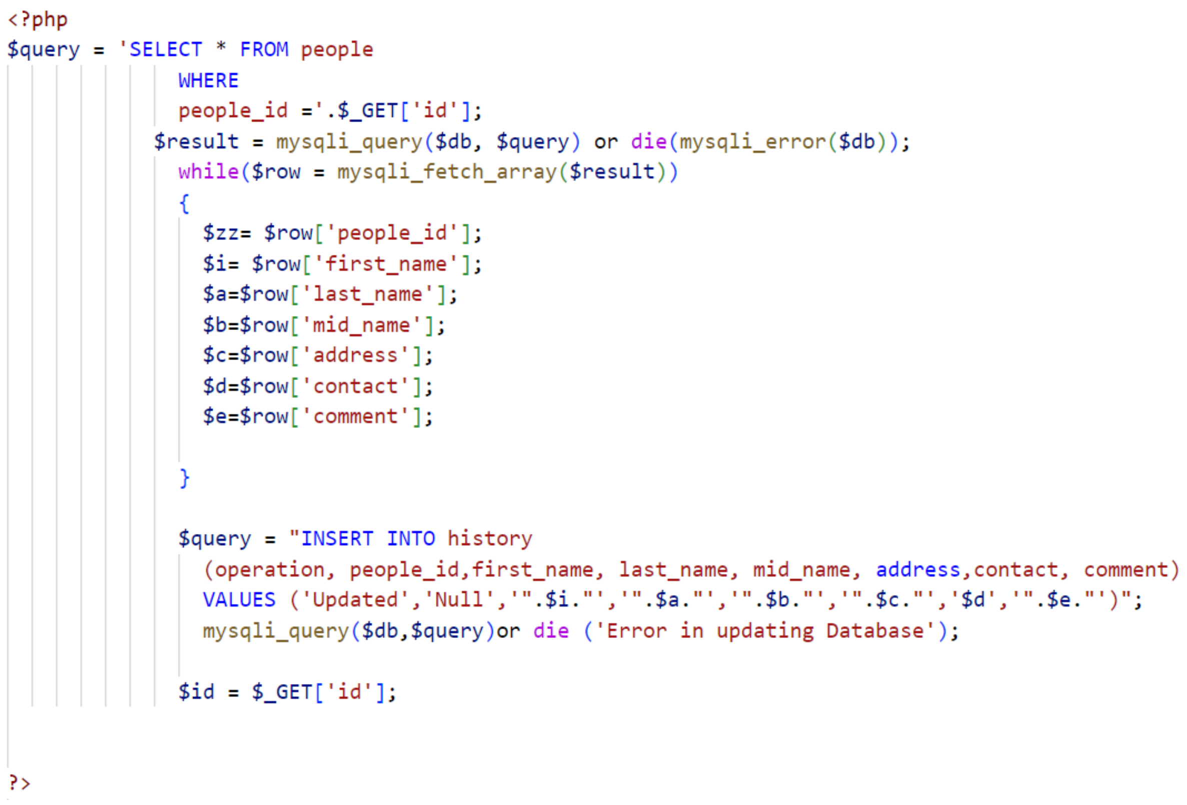
```
<div class="col-lg-12">
   <?php
        $fname = $_POST['firstname'];
        $lname = $_POST['lastname'];
        $mname = $_POST['Middlename'];
        $address = $_POST['Address'];
        $contct = $_POST['Contact'];
        $comment = $_POST['comment'];

    switch($_GET['action']){
        case 'add':
                $query = "INSERT INTO people
                (people_id,first_name, last_name, mid_name, address,contact, comment)
                VALUES ('Null','".$fname."','".$lname."','".$mname."','".$address."','$contct','".$comment."')";
                mysqli_query($db,$query)or die ('Error in updating Database');


        break;

        }
   ?>
```

2. Adding a new record in the history table to keep the history of the performed operation. This snippet is for history of “Add” operation.



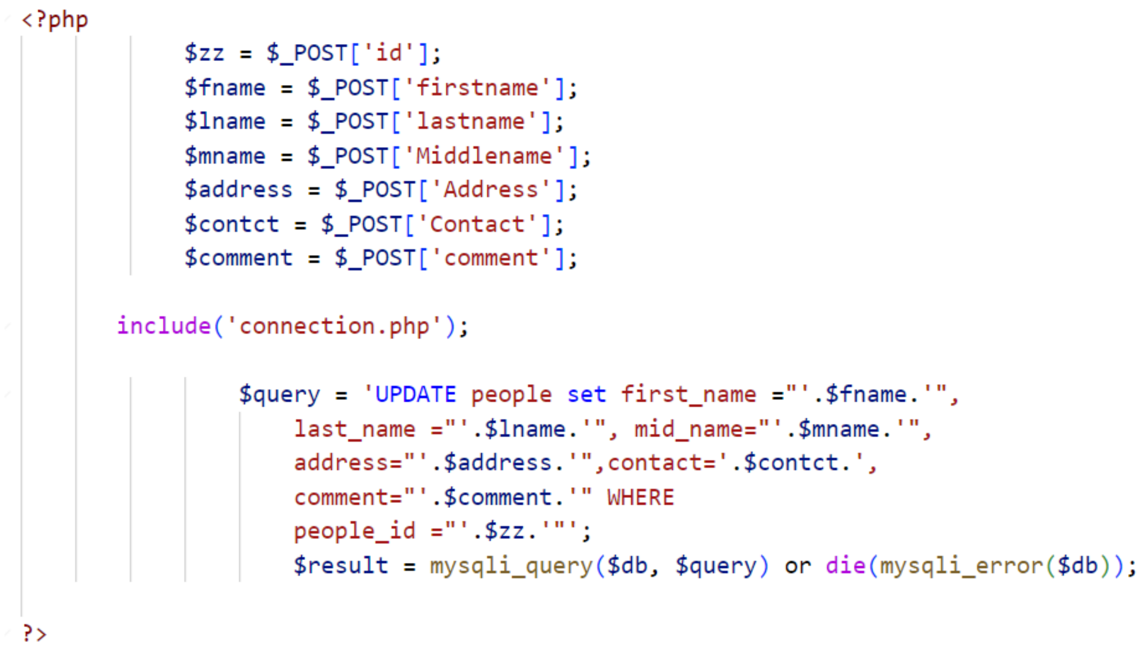
```
$query = "INSERT INTO history
(operation, people_id,first_name, last_name, mid_name, address,contact, comment)
VALUES ('Added','Null','".$fname."','".$lname."','".$mname."','".$address."','$contct','".$comment."')";
mysqli_query($db,$query)or die ('Error in updating Database');
```

3. For update operation, we need to add the record that is being modified(before modification) to the history table. So, we can check later from the history table that particular this record was modified.



```
<?php
$query = 'SELECT * FROM people
              WHERE
              people_id ='.$_GET['id'];
            $result = mysqli_query($db, $query) or die(mysqli_error($db));
              while($row = mysqli_fetch_array($result))
              {
                $zz= $row['people_id'];
                $i= $row['first_name'];
                $a=$row['last_name'];
                $b=$row['mid_name'];
                $c=$row['address'];
                $d=$row['contact'];
                $e=$row['comment'];

              }

              $query = "INSERT INTO history
                (operation, people_id,first_name, last_name, mid_name, address,contact, comment)
                VALUES ('Updated','Null','".$i."','".$a."','".$b."','".$c."','$d','".$e."')";
                mysqli_query($db,$query)or die ('Error in updating Database');

              $id = $_GET['id'];

?>
```

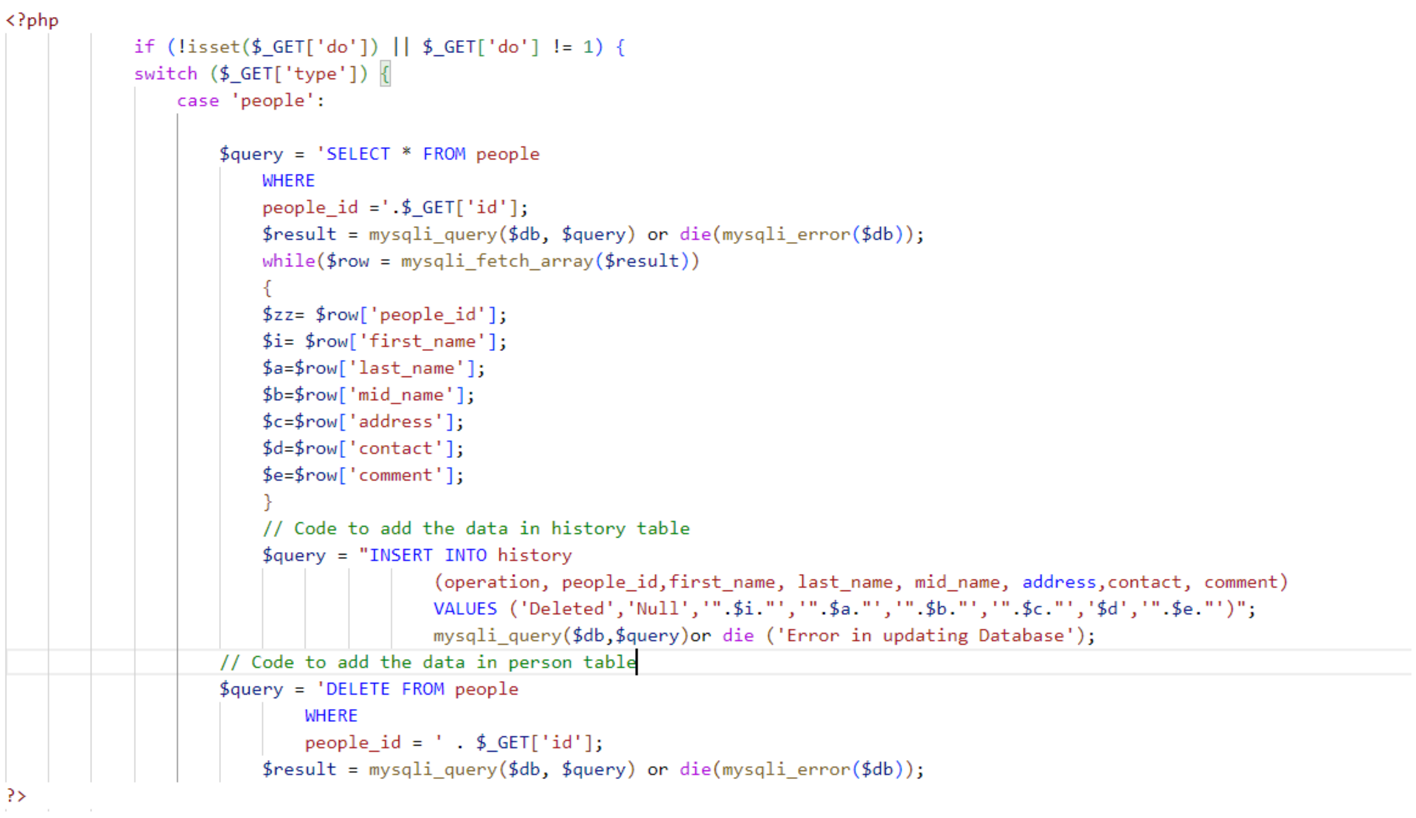
4. Code to update the record in person table.



```
<?php
            $zz = $_POST['id'];
            $fname = $_POST['firstname'];
            $lname = $_POST['lastname'];
            $mname = $_POST['Middlename'];
            $address = $_POST['Address'];
            $contct = $_POST['Contact'];
            $comment = $_POST['comment'];

        include('connection.php');

                $query = 'UPDATE people set first_name ="'.$fname.'",
                    last_name ="'.$lname.'", mid_name="'.$mname.'",
                    address="'.$address.'",contact='.$contct.',
                    comment="'.$comment.'" WHERE
                    people_id ="'.$zz.'"';
                    $result = mysqli_query($db, $query) or die(mysqli_error($db));

?>
```

5. For the delete operation as well, first, we will need to add the record which is being deleted to the history table then we will need to delete the record from the person table.

Code to add the record in the history table which is being deleted. And then, the code to delete the record from the person table.



```
<?php
        if (!isset($_GET['do']) || $_GET['do'] != 1) {
        switch ($_GET['type']) {
            case 'people':

                $query = 'SELECT * FROM people
                    WHERE
                    people_id ='.$_GET['id'];
                    $result = mysqli_query($db, $query) or die(mysqli_error($db));
                    while($row = mysqli_fetch_array($result))
                    {
                    $zz= $row['people_id'];
                    $i= $row['first_name'];
                    $a=$row['last_name'];
                    $b=$row['mid_name'];
                    $c=$row['address'];
                    $d=$row['contact'];
                    $e=$row['comment'];
                    }
                    // Code to add the data in history table
                    $query = "INSERT INTO history
                                    (operation, people_id,first_name, last_name, mid_name, address,contact, comment)
                                    VALUES ('Deleted','Null','".$i."','".$a."','".$b."','".$c."','$d','".$e."')";
                                    mysqli_query($db,$query)or die ('Error in updating Database');
                // Code to add the data in person table
                $query = 'DELETE FROM people
                        WHERE
                        people_id = ' . $_GET['id'];
                    $result = mysqli_query($db, $query) or die(mysqli_error($db));
?>
```

- **Determine the type of maintenance. Describe what maintenance-related activities you performed/did not perform (like testing/ regression testing). Describe the reason why an activity was not performed. Describe the challenges you faced while maintaining the program.**

- **Definition**: Software maintenance is the process of making changes, modifications, and updating existing software to meet customer needs. The main motive is to improve the software's performance.

- **There are four types of software maintenance.**

1. Corrective Software Maintenance

- Software defects result from mistakes and flaws in the software's logic, design, and code. These flaws and problems in your software system are fixed by corrective maintenance activity, sometimes known as "bug fixing."
- Following are the requirements when we need to perform corrective maintenance
    - It fails to perform as intended due to serious problems, such as flawed logic flow, improper implementation, invalid or insufficient tests, etc.
    - Users are still being impacted by the software's bugs after it has been deployed.
- Bug reports are typically written by users and provided as criticism to the software's developer. After reviewing the code, the company's engineers and testers update the software to reflect any necessary corrections. Its goal is to make the software better and more functional.
- The maintenance action is preventive or adaptive if you find and fix software flaws before users are aware of them. On the other hand, corrective maintenance activity is when you address the issue after receiving bug reports from the user's perspective.
- If you are spending the majority of time dealing with corrective maintenance task then pay attention to the following:-
    - Adopt a robust testing practice.
    - Develop high-quality code.
    - Focus on the correct implementation of the design specification.
    - Enhance your ability to anticipate the problems.

2. Adaptive Software Maintenance

- Updates and modifications to the software are the goals of adaptive maintenance when:
    - The operating system on which your software runs changes (due to technology, laws, policies, rules, and operating system.)
    - The product must be able to work with updated hardware or software for your customers.

  - You have identified software flaws that could eventually impact your clients.
- Let's say you are using a web application and discovered during a developer preview that it is incompatible with the new Edge browser from Microsoft, which is powered by Chrome. You are carrying out an adaptive software maintenance task if you fix this compatibility issue in your web application before the Microsoft Edge browser's final stable version, which is powered by Chromium.
- Your clients will experience problems if you wait for a stable release of the updated Edge browser, which runs on Microsoft Chromium. The modifications you make next to address those problems are considered corrective maintenance tasks. It is preferable to take adaptive maintenance action early because postponing it will force you to take a more expensive strategy known as corrective maintenance later.
- Keep in mind that changing the environment in which your software operates will also cause changes in other components of your software. For instance, changing the server, processors, compilers, etc. will also impact your software's operation.

3. Perfective Software Maintenance

- When you upgrade a software system to increase its value in response to user requests, this is known as perfective software maintenance. This comprises:
  - Speed optimization
  - Improvement in user interfaces
  - Improvements in software usability
  - Enhancement of software functionality
  - Improvement in software performance
- Some key points about Perfective software maintenance:
- It entails improving software functionality by incorporating new or modified user needs (even when the changes are not considered a defect, error or fault).
  - Customer input frequently, although not always, serves as the catalyst.
  - It is responsible for half of all maintenance tasks.

4. Preventive Software Maintenance

- Making a software change for preventive maintenance will stop faults from happening in the future. By bringing down the complexity, it improves the software's capacity to be maintained. Among the preventive maintenance tasks are:
  - Updating the documentation: Updating the document according to the current state of the system.
  - Optimizing the code: Modifying the code for faster execution of programs or making efficient use of storage space.
  - Reconstructing the code: Transforming the structure of the program by reducing the source code, making it easily understandable.

- **There are various software maintenance-related activities that I have performed.**

- **Functional testing**
- **Unit testing:** As I had added the new unit that adds the data in another table, I had to check whether the implemented unit is working well or not. For that, I had added, updated, and delete some records and it was working well.
- **Integration testing:** According to the task demand I had to integrate another database into the existing system. So, integration testing needed to be performed to check whether the database integration working well or not.
- **Smoke testing:** As smoke testing is for build verification, I did not need to perform this testing because my application did need the build.

- **Non Functional testing:**
- **Reliability:** I had to test the reliability of the application as I wanted to check if the database can handle thousands of data or not. For that, I imported thousands of records and it worked well.
- **Scalability:** As the application is at a low level, I did not need to scale it. So, I did not check the scalability.

- **Regression testing**
- This testing is necessary as we changed and improved the existing code.
- I had to perform this testing to check whether the application works well after adding a new module. It was necessary for me to make sure that the overall stability and functionality of the application remain sustainable before and after adding new features.

- **Challenges that I have faced**
- While implanting the new features, it was hard to understand the flow for me of the new feature.
- I was getting trouble creating a flow on how I should update the history table.
- But, after understanding the structure of small record-keeping systems, I understood the over all flow and developed the new feature.

Project Link :

https://drive.google.com/file/d/1wUUsshofid0vaEBkm6SpcXmxY_UEYwTd/view?usp=sharing