

# DOCUMENTATION

**Github link:** <https://github.com/Rencelagsil/CRUD.git>

# CRUD FUNCTIONALITIES


## Student List

Add New Student

ID	First Name	Middle Name	Last Name	Student ID	Email	Strand	Level	Actions
107	Rence	Diamoda	Lagsil	2311600102	lagsilrence@gmail.com	HUMSS	Grade 12	<div>EditDelete</div>
108	Justine	A.	Moril	12345	fauzi@gmail.com	STEM	Grade 11	<div>EditDelete</div>
109	Jayvee	C.	Bajarias	2311600105	jbj@gmail.com	ABM	Grade 11	<div>EditDelete</div>
110	Raph	Sho	Visayas	2311600103	raph@gmail.com	HUMSS	Grade 11	<div>EditDelete</div>
111	Johar	Datu	Gogo	2311600104	gogo@gmail.com	ABM	Grade 11	<div>EditDelete</div>

# CREATE (ADD NEW STUDENT)

## Add New Student

First Name	Middle Name	Last Name	dd/mm/yyyy 	Student ID	Street Address	City
State / Province	Country	ZIP Code	E-mail	ABM ▼	Grade 11 ▼	Add Student
Back						

- **Function:** Allows users to add new students.
- **Inputs:** First Name, Last Name, Birthdate, Student ID, Street Address, City, State Province, Country, Zip Code, Email, Strand, and Level.  
**Output:** New student record saved in the database.

# READ (VIEW STUDENT LIST)

## Student List

[Add New Student](#)

ID	First Name	Middle Name	Last Name	Student ID	Email	Strand	Level	Actions
107	Rence	Diamoda	Lagsil	2311600102	lagsilrence@gmail.com	HUMSS	Grade 12	<a href="#">Edit</a> <a href="#">Delete</a>
108	Justine	A.	Moril	12345	fauzi@gmail.com	STEM	Grade 11	<a href="#">Edit</a> <a href="#">Delete</a>
109	Jayvee	C.	Bajarias	2311600105	jbj@gmail.com	ABM	Grade 11	<a href="#">Edit</a> <a href="#">Delete</a>
110	Raph	Sho	Visayas	2311600103	raph@gmail.com	HUMSS	Grade 11	<a href="#">Edit</a> <a href="#">Delete</a>
111	Johar	Datu	Gogo	2311600104	gogo@gmail.com	ABM	Grade 11	<a href="#">Edit</a> <a href="#">Delete</a>

- **Function:** Displays all registered students.  
**Features:** View all students, search, and sort records.

# EDIT OR UPDATE ( EDIT STUDENT DETAILS)

## Edit Student

<input type="text" value="Rence"/>	<input type="text" value="Diamoda"/>	<input type="text" value="Lagsil"/>	<input type="text" value="HUMSS ▼"/>	<input type="text" value="Grade 12 ▼"/>	<input type="button" value="Update"/>
------------------------------------	--------------------------------------	-------------------------------------	--------------------------------------	---	---------------------------------------

- **Function:** Allows updating student information.  
User clicks "Edit" to modify student details, then if click the update automatically update changes and saved in the database.

# DELETE (REMOVE STUDENT RECORD)
















## Student List

Add New Student

ID	First Name	Middle Name	Last Name	Student ID	Email	Strand	Level	Actions
107	Rence	Diamoda	Lagsil	2311600102	lagsilrence@gmail.com	HUMSS	Grade 12	<a href="#">Edit</a> <a href="#">Delete</a>
108	Justine	A.	Moril	12345	fauzi@gmail.com	STEM	Grade 11	<a href="#">Edit</a> <a href="#">Delete</a>
109	Jayvee	C.	Bajarias	2311600105	jbj@gmail.com	ABM	Grade 11	<a href="#">Edit</a> <a href="#">Delete</a>
110	Raph	Sho	Visayas	2311600103	raph@gmail.com	HUMSS	Grade 11	<a href="#">Edit</a> <a href="#">Delete</a>
111	Johar	Datu	Gogo	2311600104	gogo@gmail.com	ABM	Grade 11	<a href="#">Edit</a> <a href="#">Delete</a>

- **Function:** Deletes a student record permanently.  
User clicks "Delete", confirmation appears.  
If confirmed, student record is deleted from the database.

# DATABASE SCHEMA DIAGRAM

student_registration	students
	
id : int(11)	
	first_name : varchar(50)
	middle_name : varchar(50)
	last_name : varchar(50)
	birthdate : date
	student_id : varchar(100)
	street_address : varchar(100)
	city : varchar(100)
	state_province : varchar(100)
	country : varchar(100)
	zip_code : varchar(100)
	email : varchar(100)
	strand : varchar(100)
	level : varchar(100)

# CODE SNIPPETS FOR CRUD OPERATIONS

- Create (Insert Student – insert\_student.php)

```
insert_student.php
1  <?php
2  include 'db.php'; // Include database connection
3
4  if ($_SERVER["REQUEST_METHOD"] == "POST") {
5      $first_name = $_POST['fName'];
6      $middle_name = $_POST['mName'];
7      $last_name = $_POST['lName'];
8      $birthdate = $_POST['birthdate'];
9      $student_id = $_POST['studentID'];
10     $street = $_POST['street'];
11     $city = $_POST['city'];
12     $state = $_POST['state'];
13     $country = $_POST['country'];
14     $zip = $_POST['zip'];
15     $email = $_POST['email'];
16     $strand = $_POST['strand'];
17     $level = $_POST['level'];
18
19     // Insert data into the database
20     $sql = "INSERT INTO students (first_name, middle_name, last_name, birthdate, student_id, street_address, city, state,
21     | | | | VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)";
22
23     $stmt = $conn->prepare($sql);
24     $stmt->bind_param("ssssssssssss", $first_name, $middle_name, $last_name, $birthdate, $student_id, $street, $city, $
25
26     if ($stmt->execute()) {
27         echo "<script>
28         | | | | alert('Student Added Successfully!');
29         | | | | window.location.href='students.php';
30         | | | | </script>";
31     } else {
32         echo "<script>
```



# READ (DISPLAY STUDENT – STUDENT.PHP)

student.php


```
10 <head>
11 <title>
12 </title>
13 <style>
14 </style>
55 </head>
56 <body>
57 <div class="container">
58 <h1>Student List</h1>
59 <a href="add_student.php"><button class="btn" style="background: #007bff;">Add New Student</button></a>
60 <table>
61 <tr>
62 <th>ID</th>
63 <th>First Name</th>
64 <th>Middle Name</th>
65 <th>Last Name</th>
66 <th>Student ID</th>
67 <th>Email</th>
68 <th>Strand</th>
69 <th>Level</th>
70 <th>Actions</th>
71 </tr>
72 <tr>
73 <td><?= htmlspecialchars($row['id']) ?></td>
74 <td><?= htmlspecialchars($row['first_name']) ?></td>
75 <td><?= htmlspecialchars($row['middle_name']) ?></td>
76 <td><?= htmlspecialchars($row['last_name']) ?></td>
77 <td><?= htmlspecialchars($row['student_id']) ?></td>
78 <td><?= htmlspecialchars($row['email']) ?></td>
79 <td><?= htmlspecialchars($row['strand']) ?></td>
80 <td><?= htmlspecialchars($row['level']) ?></td>
81 <td><?= htmlspecialchars($row['level']) ?></td>
82 <td><?= htmlspecialchars($row['level']) ?></td>
83 <td>
84 <a href="edit_student.php?id=<?= $row['id'] ?>" class="btn edit-btn">Edit</a>
85 <a href="delete_student.php?id=<?= $row['id'] ?>" class="btn delete-btn" onclick="return confirm('Are you sure you want to delete this student?')">Delete</a></td>
86 </tr>
87 </table>
88 </div>
89 </body>
90 </html>
```

# UPDATE (EDIT STUDENT – EDIT\_STUDENT.PHP)

edit\_student.php

```
1  <?php
2  include 'db.php';
3
4  $id = $_GET['id'];
5  $result = $conn->query("SELECT * FROM students WHERE id=$id");
6  $row = $result->fetch_assoc();
7
8  if ($_SERVER["REQUEST_METHOD"] == "POST") {
9      $first_name = $_POST['fName'];
10     $middle_name = $_POST['mName'];
11     $last_name = $_POST['lName'];
12     $strand = $_POST['strand'];
13     $level = $_POST['level'];
14
15     $sql = "UPDATE students SET first_name='$first_name', middle_name='$middle_name', last_name='$last_name', strand='$strand', level='$level' WHERE id=$id";
16
17     if ($conn->query($sql) === TRUE) {
18         echo "<script>alert('Student Updated Successfully'); window.location.href='students.php';</script>";
19     }
20 }
21 ?>
22
23 <!DOCTYPE html>
24 <html lang="en">
25 <head>
26     <meta charset="UTF-8">
27     <title>Edit Student</title>
28 </head>
29 <body>
30     <h1>Edit Student</h1>
31     <form method="POST">
32         <input type="text" name="fName" value="<?= $row['first_name'] ?>" required>
```

## DELETE (REMOVE STUDENT – DELETE\_STUDENT.PHP)

 delete\_student.php

```
1  <?php
2  include 'db.php';
3  $id = $_GET['id'];
4  $conn->query("DELETE FROM students WHERE id=$id");
5  header("Location: students.php");
6  ?>
7  
```

# CHALLENGES & SOLUTIONS

## CHALLENGES

- Redirect issues after adding students
- Redirect issues after adding students
- CRUD operations failing
- UI not user-friendly

## SOLUTIONS

- Used `window.location.href='students.php'` in JavaScript alerts
- Verified MySQL settings in `db.php`
- Used prepared statements to prevent SQL errors
- Improved styling with CSS for better user experience

# CONCLUSION

- The SMS or Student Management System successfully implements CRUD operations, enabling seamless student record management with a secure and user-friendly interface. Using prepared statements ensures data security, while a well-structured database maintains integrity. The responsive UI enhances usability, making it easy to add, view, update, and delete student records. Challenges such as redirection issues, database connectivity errors, and UI inconsistencies were resolved through debugging, error handling, and styling improvements. The system is now fully functional and serves as a strong foundation for future enhancements like search filters, role-based access, and advanced reporting features to further streamline student management.