Screnario-1

```
<?php
$conn = mysqli_connect("localhost", "root", "", "class_db");

# Change the POST to GET.
# We need to change it because we are sending data through URL.
$id = $_GET['id'];
$sql = "SELECT * FROM students WHERE id = $id";
$res = mysqli_query($conn, $sql);
$r = mysqli_fetch_assoc($res);
echo $r['first_name'];
?>
```

Scenario-2

```
<?php
$conn = mysqli_connect("localhost","root","","class_db");
$fname = $_POST['fname'];
#Not inside the quotes because fname is a variable
#It show error if we not put ' ' around $fname
$sql = "SELECT * FROM students WHERE first_name = '$fname' ";
$res = mysqli_query($conn, $sql);
?>
```

Scenario-3

```
<?php
$conn = mysqli_connect("localhost","root","","class_db");
```

```
#If we put user input directly in sql is not safe
$stmt = $conn->prepare("SELECT * FROM students WHERE age = ?");
$stmt->bind_param("i", $age);
$stmt->execute();
?>
```

Scenario-4

```
<?php
$conn = mysqli_connect("localhost","root","","class_db");
# We check if the fields are not empty before inserting
if (!empty($_POST['fname']) && !empty($_POST['lname'])) {
    $first = $_POST['fname'];
    $last = $_POST['lname'];
    # Only insert when both have values
    $sql = "INSERT INTO students (first_name, last_name) VALUES ('$first', '$last')";
    mysqli_query($conn, $sql);
    echo "Inserted!";
} else {
    # one or both fields are empty
    echo "Please fill out both first and last name.";
}
?>
```

Scenario-5

```
<?php
$conn = mysqli_connect("localhost","root","","class_db");
```

```
# The POST key was misspelled, so we fix it to 'email'
$email = $_POST['email'];

$sql = "SELECT * FROM students WHERE email='$email'";
$res = mysqli_query($conn, $sql);
?>
```

Scenario-6

```
<?php
$conn = mysqli_connect("localhost","root","","class_db");
#We convert it into integer to prevent harmful input like ?id=0 OR 1=1.
$id = intval($_GET['id']);
$sql = "DELETE FROM students WHERE id = $id";
mysqli_query($conn, $sql);
?>
```

Scenario-7

```
<?php
$conn = mysqli_connect("localhost","root","","class_db");
$id = $_POST['id'];
$email = $_POST['email'];
# Wrap the email in quotes since it's a string ' '
# Add error handling to avoid showing "Updated " if the query fails
$sql = "UPDATE students SET email='$email' WHERE id=$id";
if (!$res = mysqli_query($conn, $sql)) {
    echo "Error updating!";
}
?>
```

Scenario-8

```
<?php
$conn = mysqli_connect("localhost","root","","class_db");
$res = mysqli_query($conn,"SELECT * FROM students");
# Use a while loop to fetch and display all students
# Instead of fetching just a single row
while ($row = mysqli_fetch_assoc($res)) {
    echo $row['email'] . "<br>";
}
?>
```

Scenario-9

```
<?php
# Get the 'id' value from the URL query string
$id = $_GET['id'];
?>
<a href="view.php?id=3">View Student</a>
```

Scenario-10

```
<?php
$age = $_POST['age'];
# Use the correct variable 'age' (was misspelled as 'aeg' before)
```

```
$sql = "SELECT * FROM students WHERE age = $age";
?>
```

Scenario-11

```
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title> Scenario 11 </title>
  </head>
  <body>
    <form method="GET" action="save.php">
      <input name="email">
    </form>
  </body>
</html>


<?php
$email = $_GET['email'];
?>
```

Scenario-12

```
<?php
$id = $_GET['id'];
#'id' is a number, so we don't put quotes around it in the SQL query
$sql = "SELECT * FROM students WHERE id = $id";
?>
```

```
}
```

Scenario-13

```
<?php
$newEmail = $_POST['email'];
# Use WHERE to update only the specific student, otherwise all rows would change
$sql = "UPDATE students SET email='$newEmail' WHERE student_id=$id";
mysqli_query($conn,$sql);
?>
```

Scenario-14

```
<?php
$data = $_POST;
# Make sure array keys are correct and string values are wrapped in quotes for SQL
$sql = "INSERT INTO students (first_name, last_name, email)
   VALUES ('{$data['first_name']}', '{$data['last_name']}', '{$data['email']}')";
?>
```

Scenario-15

```
<?php
# Get the page number from the URL
$page = $_GET['page'];
#  Convert it to an integer to prevent invalid input
$page = intval($page);
# Make sure the page number is not negative
if ($page < 0) {
   $page = 0;
}
```

```
# Calculate how many records to skip for pagination

$limit = 5;

$offset = $page * $limit;

# Get a limited set of students for the current page

$sql = "SELECT * FROM students LIMIT $offset, $limit";

?>
```