Block 2: Back-end development

St. Alphonsus Primary School

Report

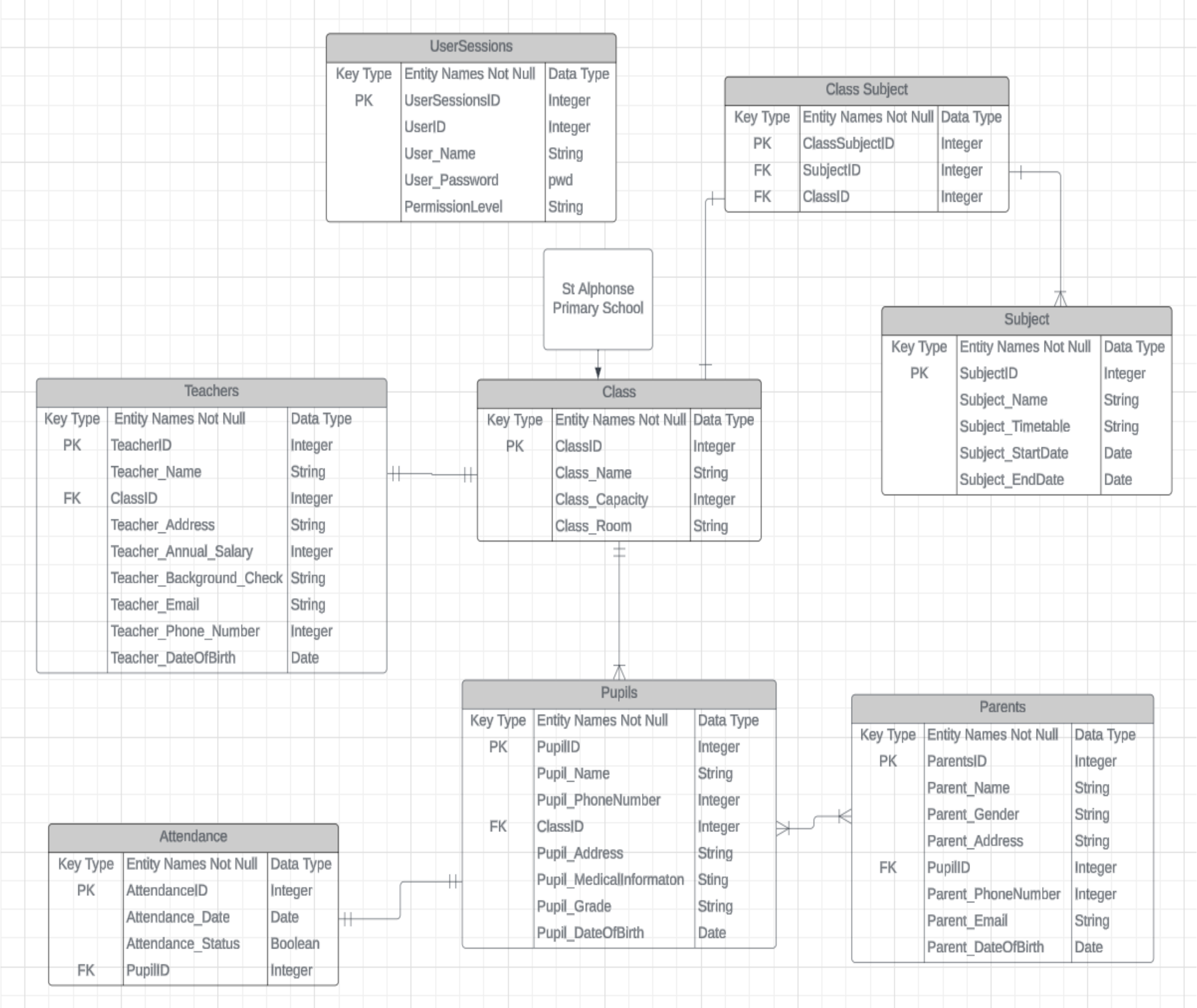
By Toseef Ashraf Parveen

Table of Content

Introduction

The assignment for this project was to create a back-end development database for St. Alphonsus primary school because previously they were using a pen and paper model which was very time consuming and inefficient. For this reason, the assignment was to create a web-based digital system, including a relational database to hold all the data digitally and make the system more efficient. Before making the database, an entity relationship diagram had to be made to not only describe the relationship between tables but to also get a preview of what the database would look like.

ER diagram:



UserSessions

Key Type	Entity Names Not Null	Data Type
PK	UserSessionsID	Integer
	UserID	Integer
	User_Name	String
	User_Password	pwd
	PermissionLevel	String

Class Subject

Key Type	Entity Names Not Null	Data Type
PK	ClassSubjectID	Integer
FK	SubjectID	Integer
FK	ClassID	Integer

Subject

Key Type	Entity Names Not Null	Data Type
PK	SubjectID	Integer
	Subject_Name	String
	Subject_Timetable	String
	Subject_StartDate	Date
	Subject_EndDate	Date

Teachers

Key Type	Entity Names Not Null	Data Type
PK	TeacherID	Integer
	Teacher_Name	String
FK	ClassID	Integer
	Teacher_Address	String
	Teacher_Annual_Salary	Integer
	Teacher_Background_Check	String
	Teacher_Email	String
	Teacher_Phone_Number	Integer
	Teacher_DateOfBirth	Date

Class

Key Type	Entity Names Not Null	Data Type
PK	ClassID	Integer
	Class_Name	String
	Class_Capacity	Integer
	Class_Room	String

Pupils

Key Type	Entity Names Not Null	Data Type
PK	PupilID	Integer
	Pupil_Name	String
	Pupil_PhoneNumber	Integer
FK	ClassID	Integer
	Pupil_Address	String
	Pupil_MedicalInformaton	Sting
	Pupil_Grade	String
	Pupil_DateOfBirth	Date

Parents

Key Type	Entity Names Not Null	Data Type
PK	ParentsID	Integer
	Parent_Name	String
	Parent_Gender	String
	Parent_Address	String
FK	PupilID	Integer
	Parent_PhoneNumber	Integer
	Parent_Email	String
	Parent_DateOfBirth	Date

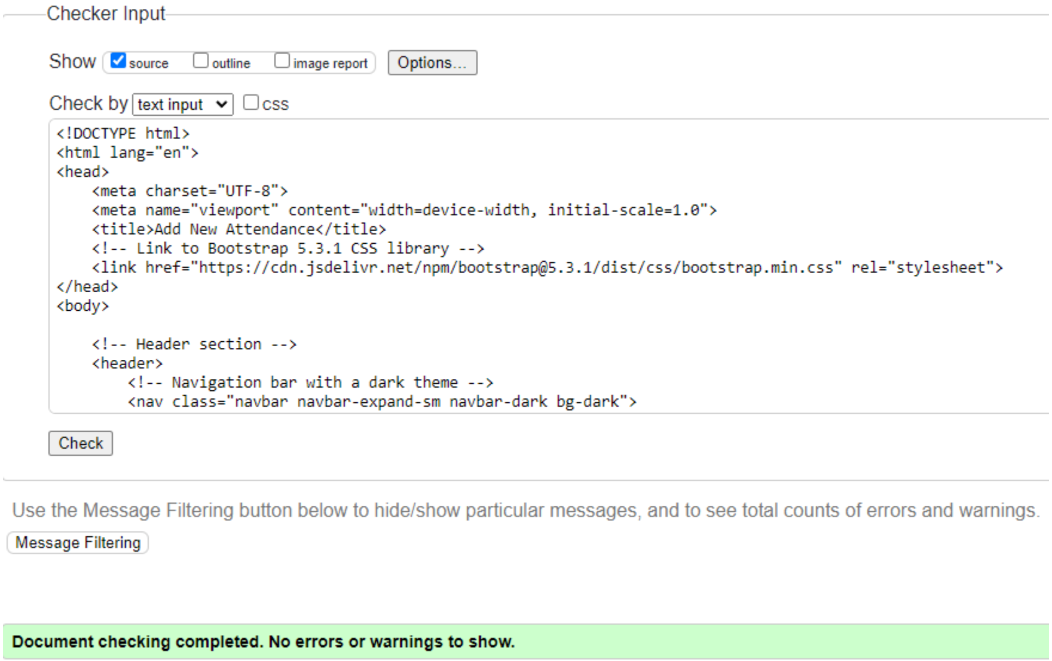
Attendance

Key Type	Entity Names Not Null	Data Type
PK	AttendanceID	Integer
	Attendance_Date	Date
	Attendance_Status	Boolean
FK	PupilID	Integer

Figure 1

PHP Code:

HTML Validation:



Checker Input

Show ☑ source □ outline □ image report Options...

Check by text input □ css

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Add New Attendance</title>
    <!-- Link to Bootstrap 5.3.1 CSS library -->
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.1/dist/css/bootstrap.min.css" rel="stylesheet">
</head>
<body>

    <!-- Header section -->
    <header>
        <!-- Navigation bar with a dark theme -->
        <nav class="navbar navbar-expand-sm navbar-dark bg-dark">
```

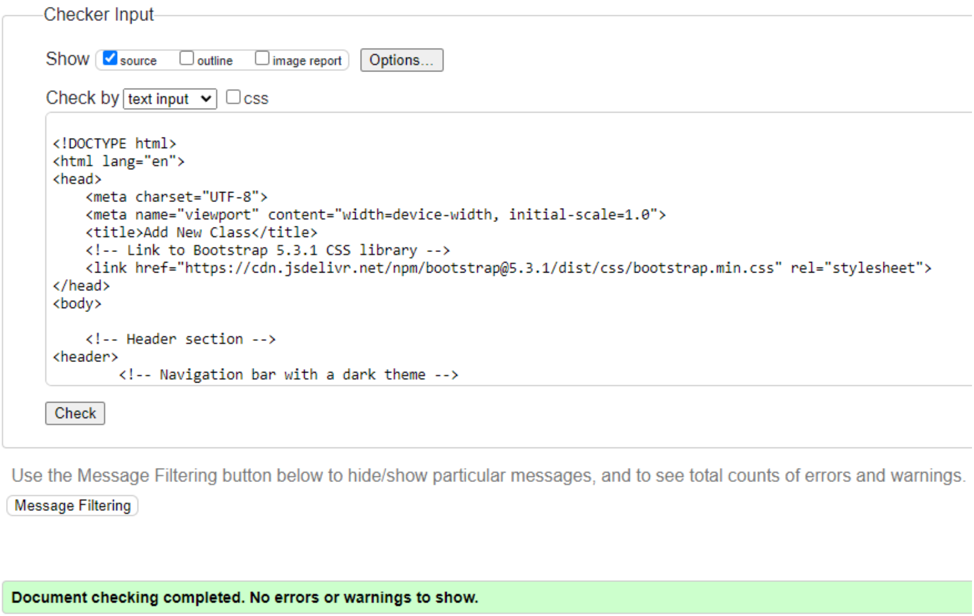
Check

Use the Message Filtering button below to hide/show particular messages, and to see total counts of errors and warnings.

Message Filtering

Document checking completed. No errors or warnings to show.

Figure 2



Checker Input

Show ☑ source □ outline □ image report Options...

Check by text input □ css

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Add New Class</title>
    <!-- Link to Bootstrap 5.3.1 CSS library -->
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.1/dist/css/bootstrap.min.css" rel="stylesheet">
</head>
<body>

    <!-- Header section -->
<header>
        <!-- Navigation bar with a dark theme -->
```

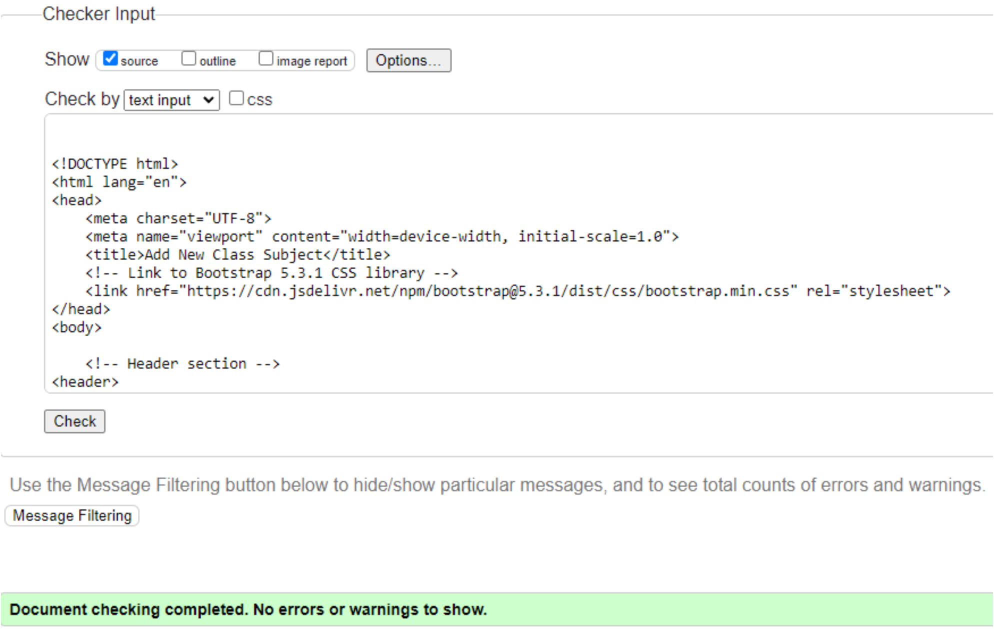
Check

Use the Message Filtering button below to hide/show particular messages, and to see total counts of errors and warnings.

Message Filtering

Document checking completed. No errors or warnings to show.

Figure 3



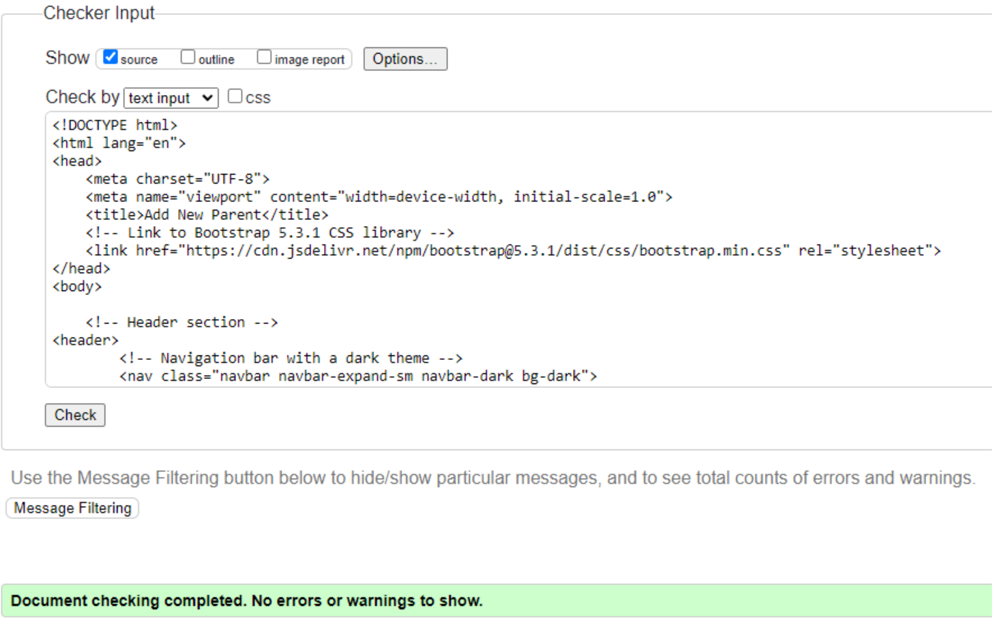
Checker Input
Show ☑ source □ outline □ image report
Options...
Check by text input □ css
Check
Message Filtering

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Add New Class Subject</title>
    <!-- Link to Bootstrap 5.3.1 CSS library -->
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.1/dist/css/bootstrap.min.css" rel="stylesheet">
</head>
<body>

    <!-- Header section -->
<header>
```

Use the Message Filtering button below to hide/show particular messages, and to see total counts of errors and warnings.

Document checking completed. No errors or warnings to show.

Figure 4



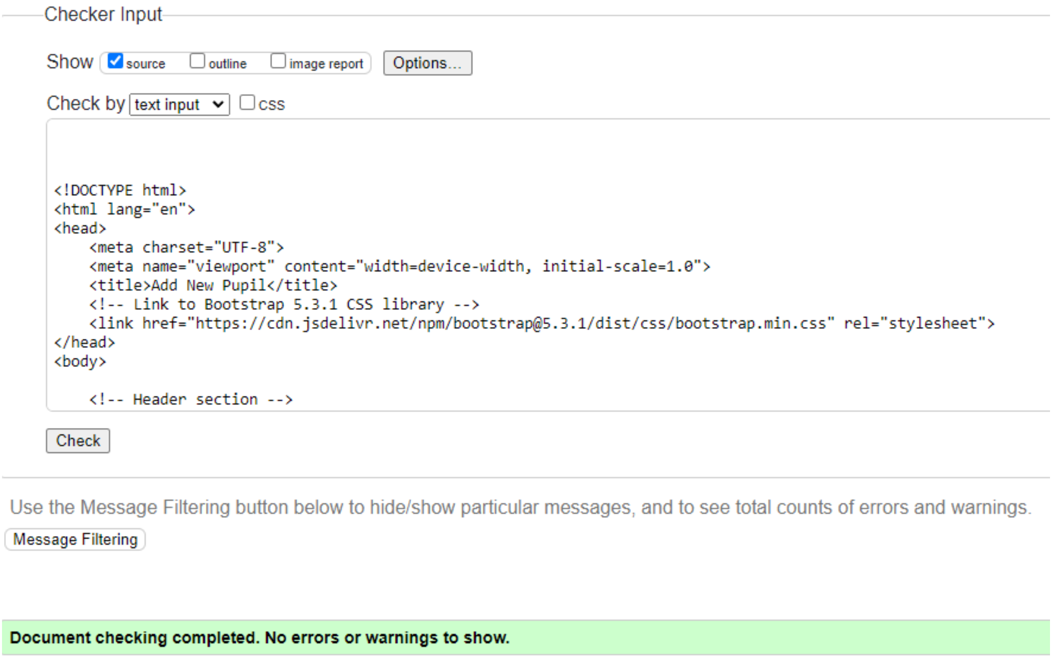
Checker Input
Show ☑ source □ outline □ image report
Options...
Check by text input □ css
Check
Message Filtering

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Add New Parent</title>
    <!-- Link to Bootstrap 5.3.1 CSS library -->
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.1/dist/css/bootstrap.min.css" rel="stylesheet">
</head>
<body>

    <!-- Header section -->
<header>
        <!-- Navigation bar with a dark theme -->
        <nav class="navbar navbar-expand-sm navbar-dark bg-dark">
```

Use the Message Filtering button below to hide/show particular messages, and to see total counts of errors and warnings.

Document checking completed. No errors or warnings to show.

Figure 5

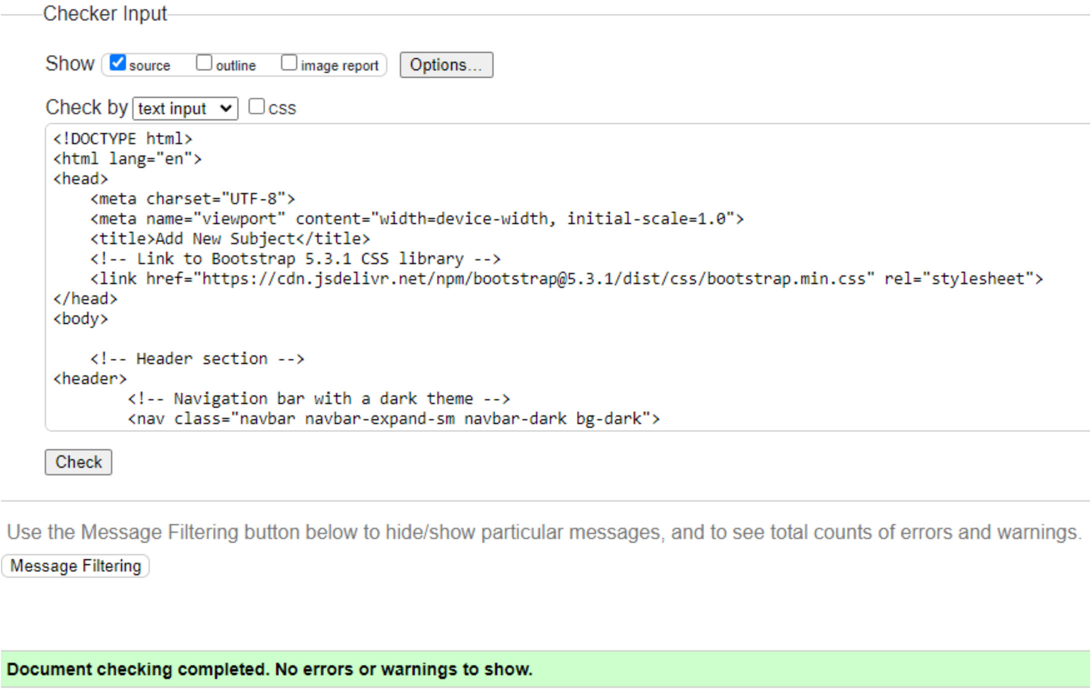

Checker Input
Show ☑ source ☐ outline ☐ image report
Options...
Check by text input ☐ css
Check
Message Filtering

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Add New Pupil</title>
    <!-- Link to Bootstrap 5.3.1 CSS library -->
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.1/dist/css/bootstrap.min.css" rel="stylesheet">
</head>
<body>

    <!-- Header section -->
```

Use the Message Filtering button below to hide/show particular messages, and to see total counts of errors and warnings.

Document checking completed. No errors or warnings to show.

Figure 6

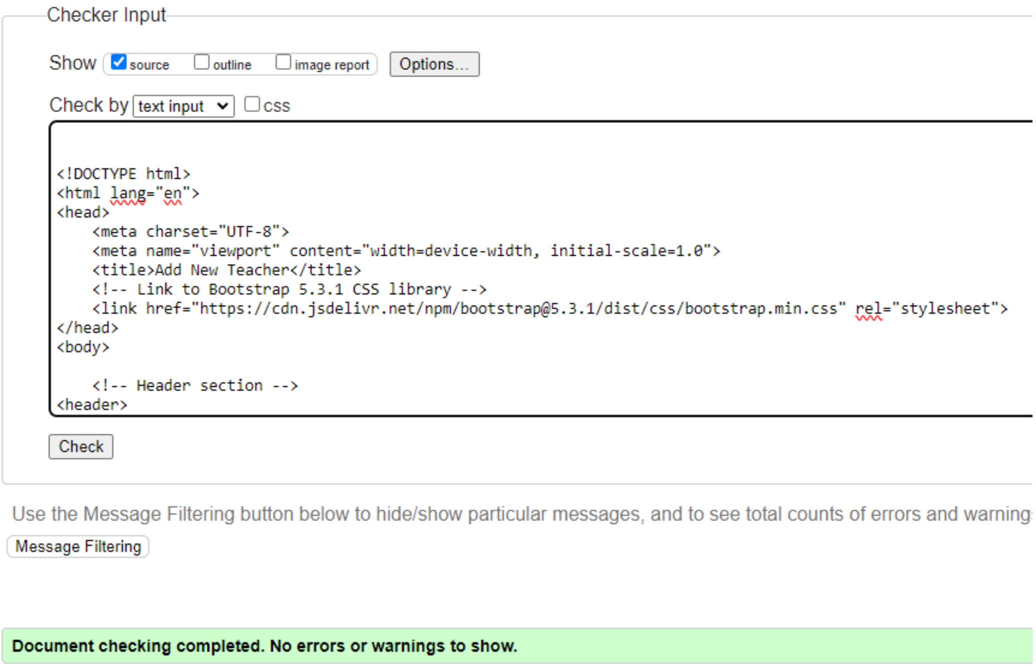

Checker Input
Show ☑ source ☐ outline ☐ image report
Options...
Check by text input ☐ css
Check
Message Filtering

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Add New Subject</title>
    <!-- Link to Bootstrap 5.3.1 CSS library -->
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.1/dist/css/bootstrap.min.css" rel="stylesheet">
</head>
<body>

    <!-- Header section -->
<header>
        <!-- Navigation bar with a dark theme -->
        <nav class="navbar navbar-expand-sm navbar-dark bg-dark">
```

Use the Message Filtering button below to hide/show particular messages, and to see total counts of errors and warnings.

Document checking completed. No errors or warnings to show.

Figure 7



Show ☑ source □ outline □ image report

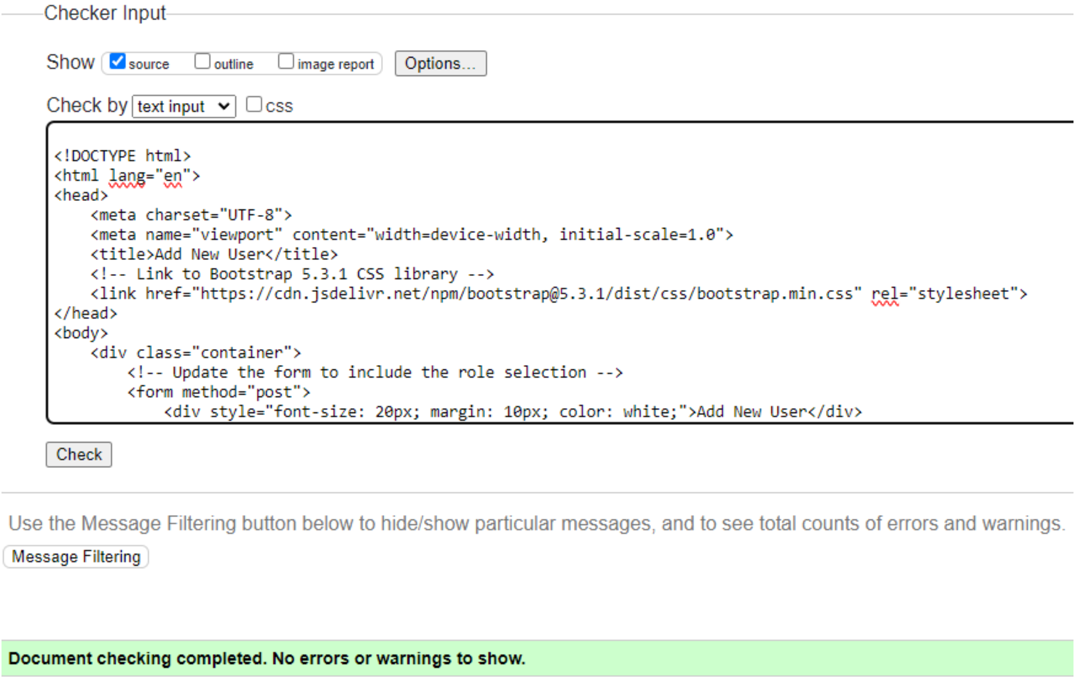
Check by text input □ css

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Add New Teacher</title>
    <!-- Link to Bootstrap 5.3.1 CSS library -->
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.1/dist/css/bootstrap.min.css" rel="stylesheet">
</head>
<body>

    <!-- Header section -->
<header>
```

Use the Message Filtering button below to hide/show particular messages, and to see total counts of errors and warning

Document checking completed. No errors or warnings to show.

Figure 8



Show ☑ source □ outline □ image report

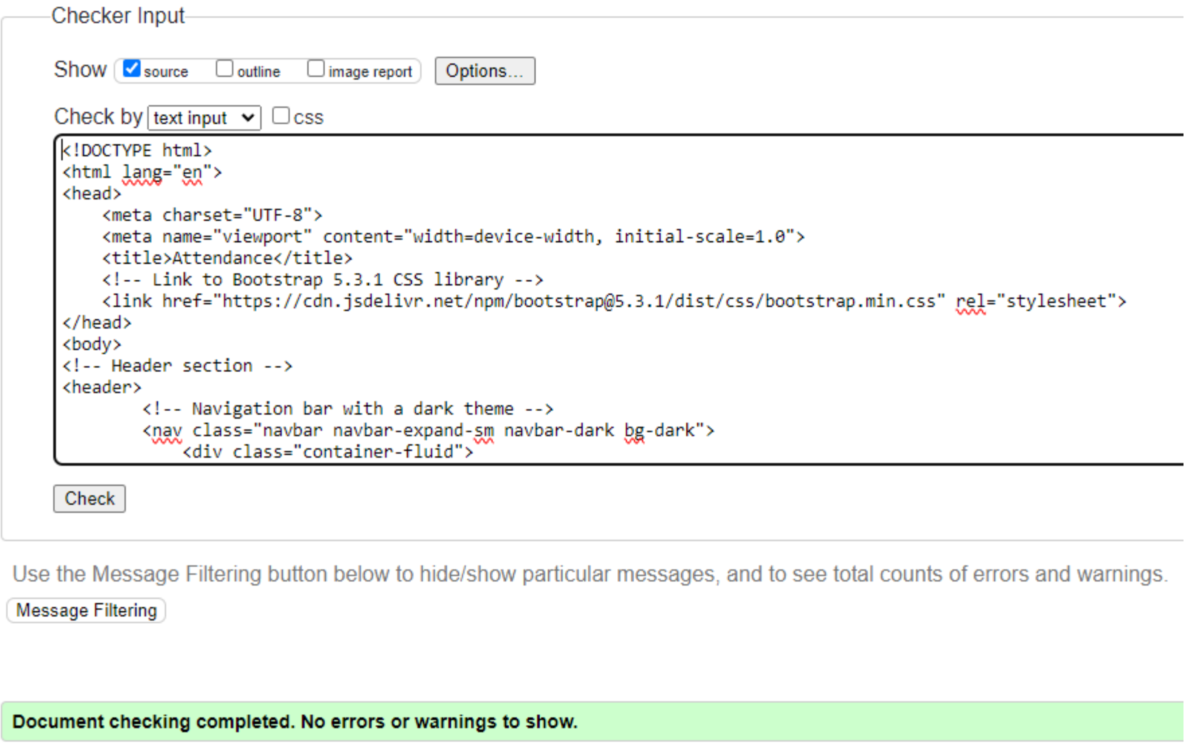
Check by text input □ css

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Add New User</title>
    <!-- Link to Bootstrap 5.3.1 CSS library -->
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.1/dist/css/bootstrap.min.css" rel="stylesheet">
</head>
<body>
    <div class="container">
        <!-- Update the form to include the role selection -->
        <form method="post">
            <div style="font-size: 20px; margin: 10px; color: white;">Add New User</div>
```

Use the Message Filtering button below to hide/show particular messages, and to see total counts of errors and warnings.

Document checking completed. No errors or warnings to show.

Figure 9

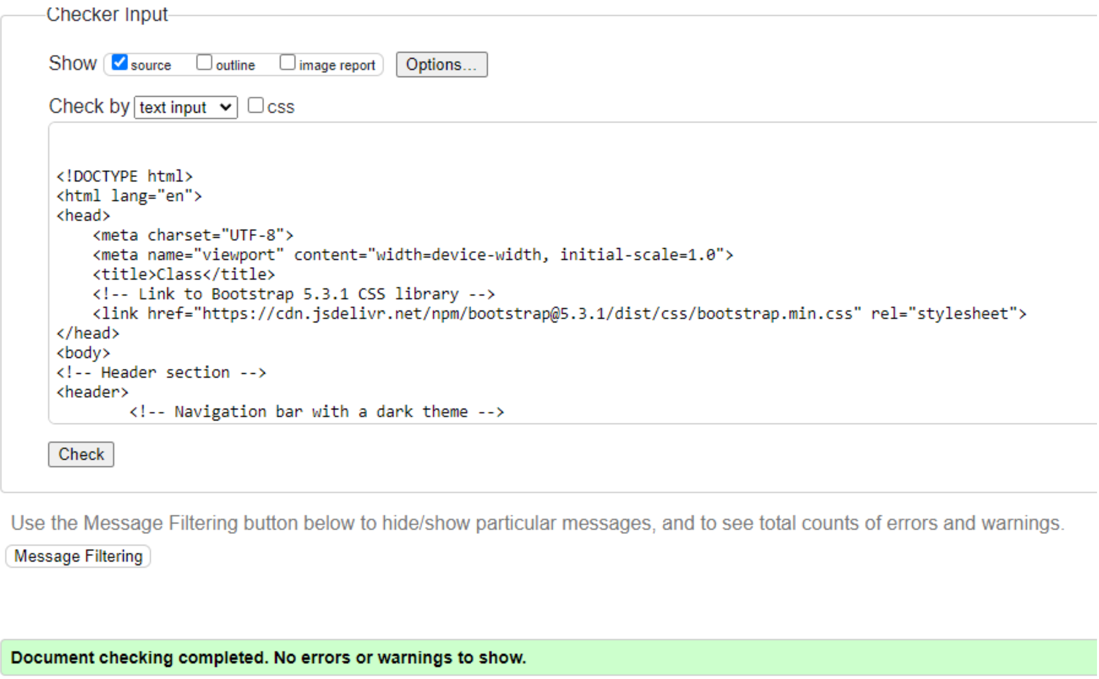

Checker Input
Show ☑ source □ outline □ image report Options...
Check by text input □ css
Check
Message Filtering

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Attendance</title>
    <!-- Link to Bootstrap 5.3.1 CSS library -->
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.1/dist/css/bootstrap.min.css" rel="stylesheet">
</head>
<body>
<!-- Header section -->
<header>
        <!-- Navigation bar with a dark theme -->
        <nav class="navbar navbar-expand-sm navbar-dark bg-dark">
            <div class="container-fluid">
```

Use the Message Filtering button below to hide/show particular messages, and to see total counts of errors and warnings.

Document checking completed. No errors or warnings to show.

Figure 10

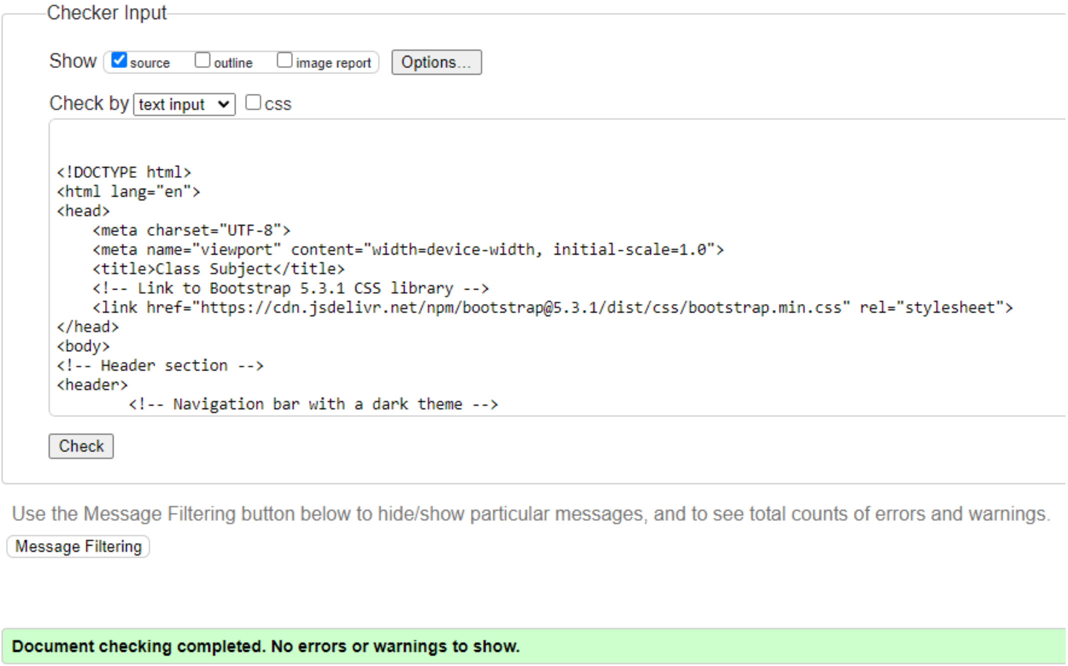

Checker Input
Show ☑ source □ outline □ image report Options...
Check by text input □ css
Check
Message Filtering

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Class</title>
    <!-- Link to Bootstrap 5.3.1 CSS library -->
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.1/dist/css/bootstrap.min.css" rel="stylesheet">
</head>
<body>
<!-- Header section -->
<header>
        <!-- Navigation bar with a dark theme -->
```

Use the Message Filtering button below to hide/show particular messages, and to see total counts of errors and warnings.

Document checking completed. No errors or warnings to show.

Figure 11

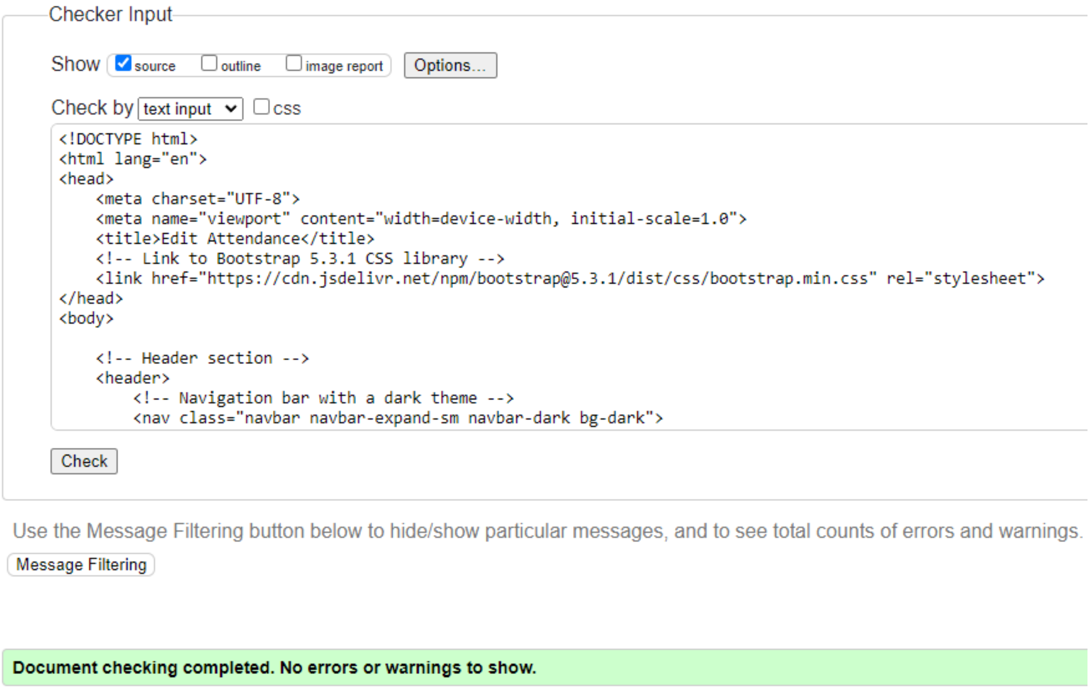

Checker Input
Show ☑ source □ outline □ image report
Options...
Check by text input □ css
Check
Message Filtering

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Class Subject</title>
    <!-- Link to Bootstrap 5.3.1 CSS library -->
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.1/dist/css/bootstrap.min.css" rel="stylesheet">
</head>
<body>
<!-- Header section -->
<header>
        <!-- Navigation bar with a dark theme -->
```

Use the Message Filtering button below to hide/show particular messages, and to see total counts of errors and warnings.

Document checking completed. No errors or warnings to show.

Figure 12

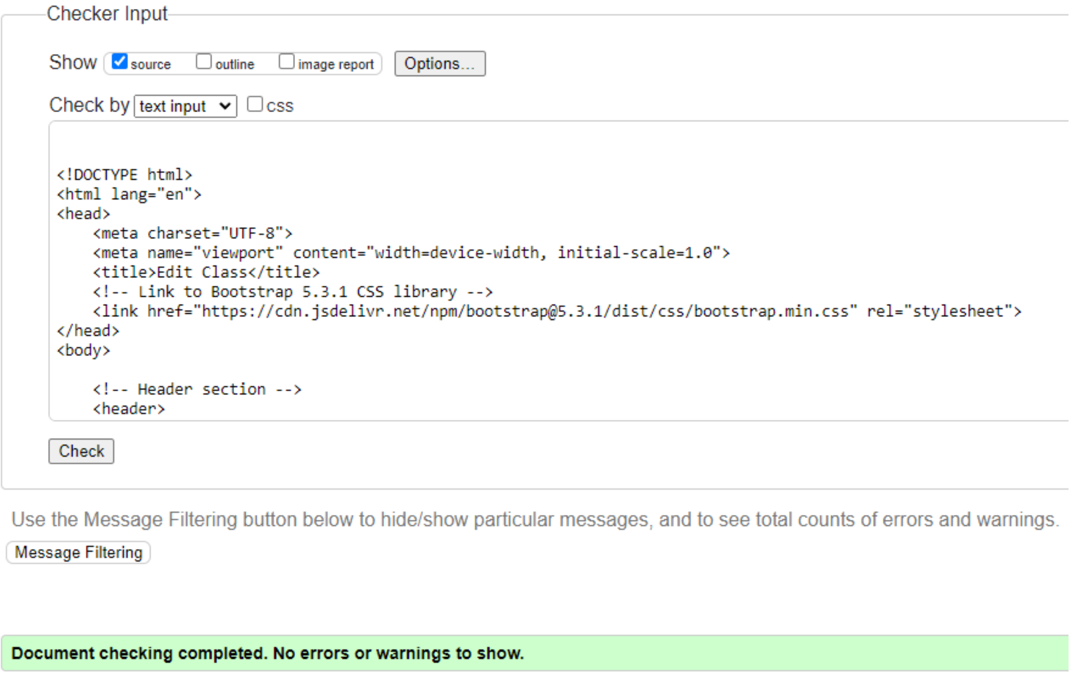

Checker Input
Show ☑ source □ outline □ image report
Options...
Check by text input □ css
Check
Message Filtering

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Edit Attendance</title>
    <!-- Link to Bootstrap 5.3.1 CSS library -->
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.1/dist/css/bootstrap.min.css" rel="stylesheet">
</head>
<body>

    <!-- Header section -->
    <header>
        <!-- Navigation bar with a dark theme -->
        <nav class="navbar navbar-expand-sm navbar-dark bg-dark">
```

Use the Message Filtering button below to hide/show particular messages, and to see total counts of errors and warnings.

Document checking completed. No errors or warnings to show.

Figure 13



Checker Input

Show ☑ source □ outline □ image report Options...

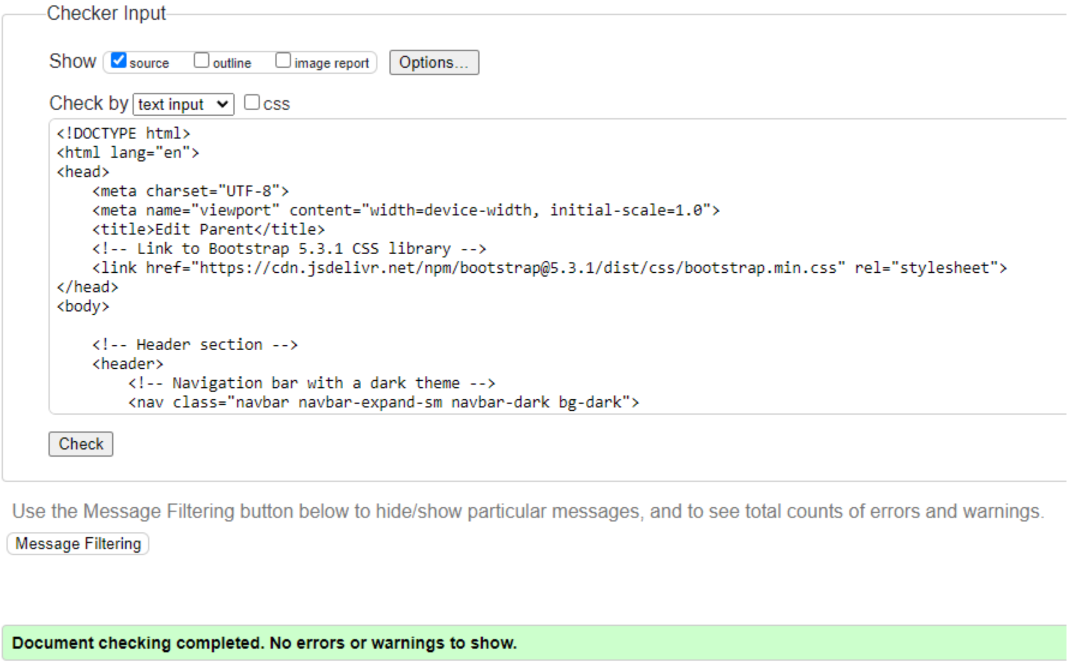
Check by text input □ css

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Edit Class</title>
    <!-- Link to Bootstrap 5.3.1 CSS library -->
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.1/dist/css/bootstrap.min.css" rel="stylesheet">
</head>
<body>

    <!-- Header section -->
    <header>
```

Use the Message Filtering button below to hide/show particular messages, and to see total counts of errors and warnings.

Document checking completed. No errors or warnings to show.

Figure 14



Checker Input

Show ☑ source □ outline □ image report Options...

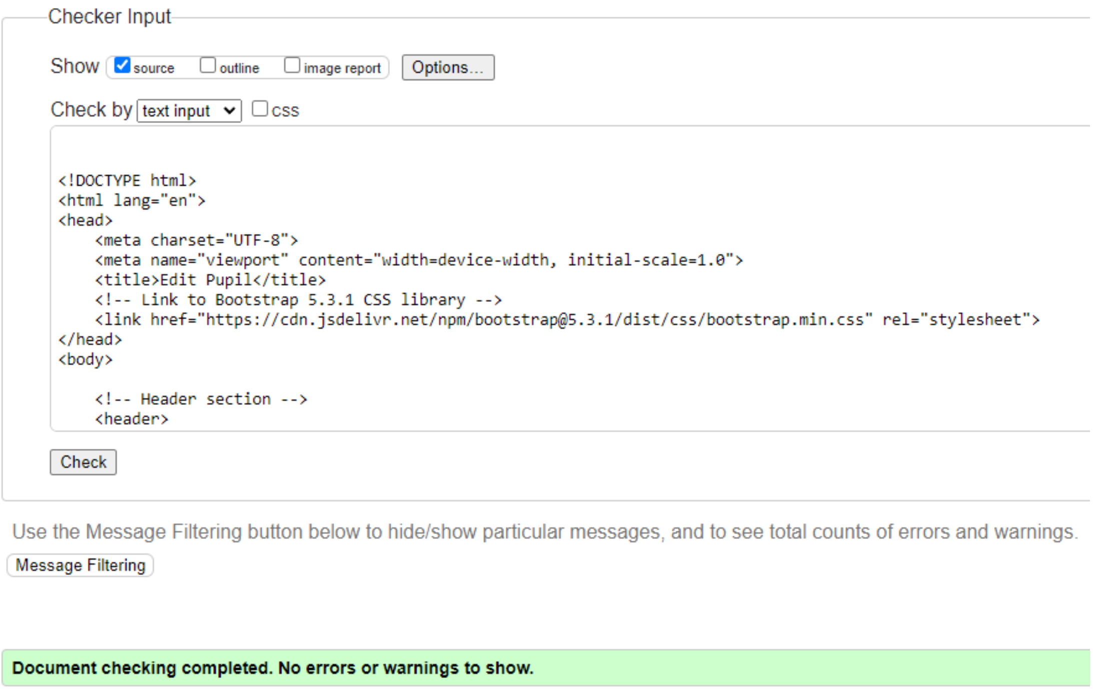
Check by text input □ css

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Edit Parent</title>
    <!-- Link to Bootstrap 5.3.1 CSS library -->
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.1/dist/css/bootstrap.min.css" rel="stylesheet">
</head>
<body>

    <!-- Header section -->
    <header>
        <!-- Navigation bar with a dark theme -->
        <nav class="navbar navbar-expand-sm navbar-dark bg-dark">
```

Use the Message Filtering button below to hide/show particular messages, and to see total counts of errors and warnings.

Document checking completed. No errors or warnings to show.

Figure 16



Show ☑ source □ outline □ image report

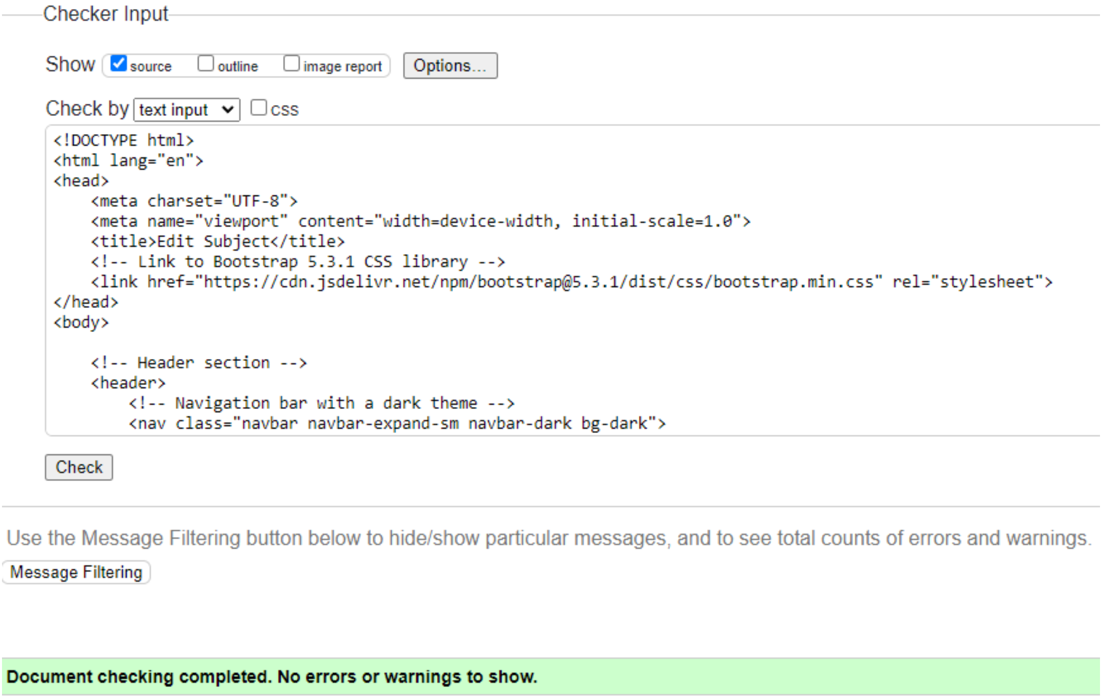
Check by text input □ css

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Edit Pupil</title>
    <!-- Link to Bootstrap 5.3.1 CSS library -->
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.1/dist/css/bootstrap.min.css" rel="stylesheet">
</head>
<body>

    <!-- Header section -->
    <header>
```

Use the Message Filtering button below to hide/show particular messages, and to see total counts of errors and warnings.

Document checking completed. No errors or warnings to show.

Figure 17



Show ☑ source □ outline □ image report

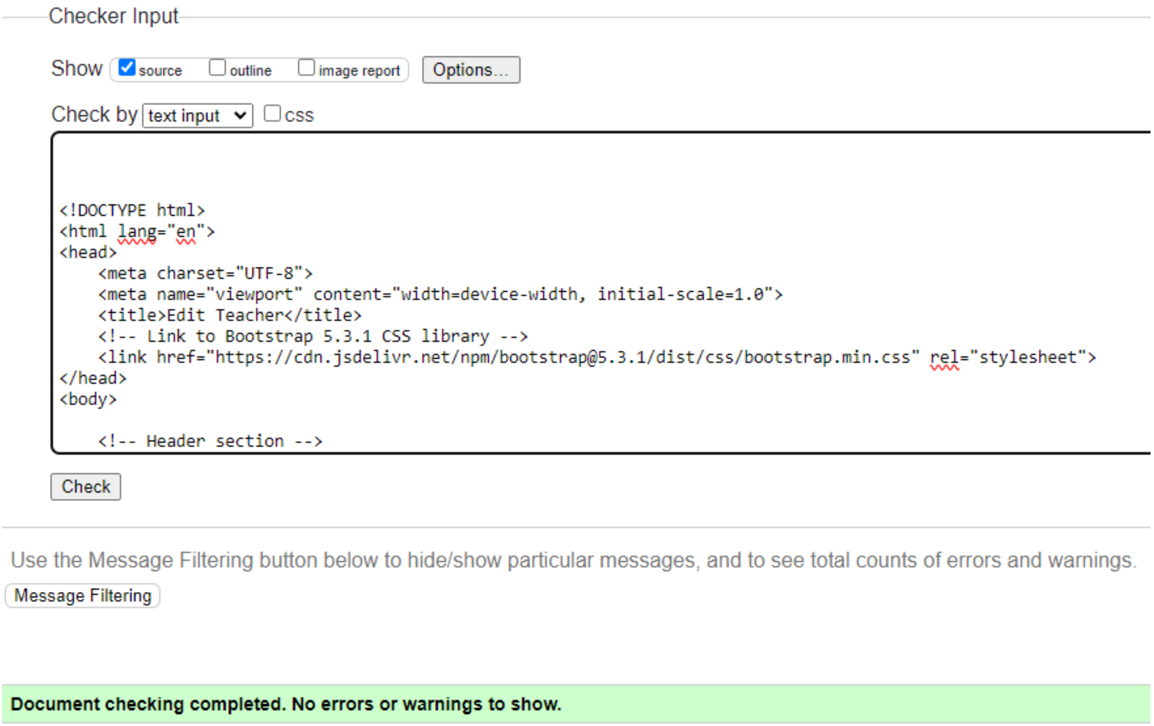
Check by text input □ css

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Edit Subject</title>
    <!-- Link to Bootstrap 5.3.1 CSS library -->
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.1/dist/css/bootstrap.min.css" rel="stylesheet">
</head>
<body>

    <!-- Header section -->
    <header>
        <!-- Navigation bar with a dark theme -->
        <nav class="navbar navbar-expand-sm navbar-dark bg-dark">
```

Use the Message Filtering button below to hide/show particular messages, and to see total counts of errors and warnings.

Document checking completed. No errors or warnings to show.

Figure 18

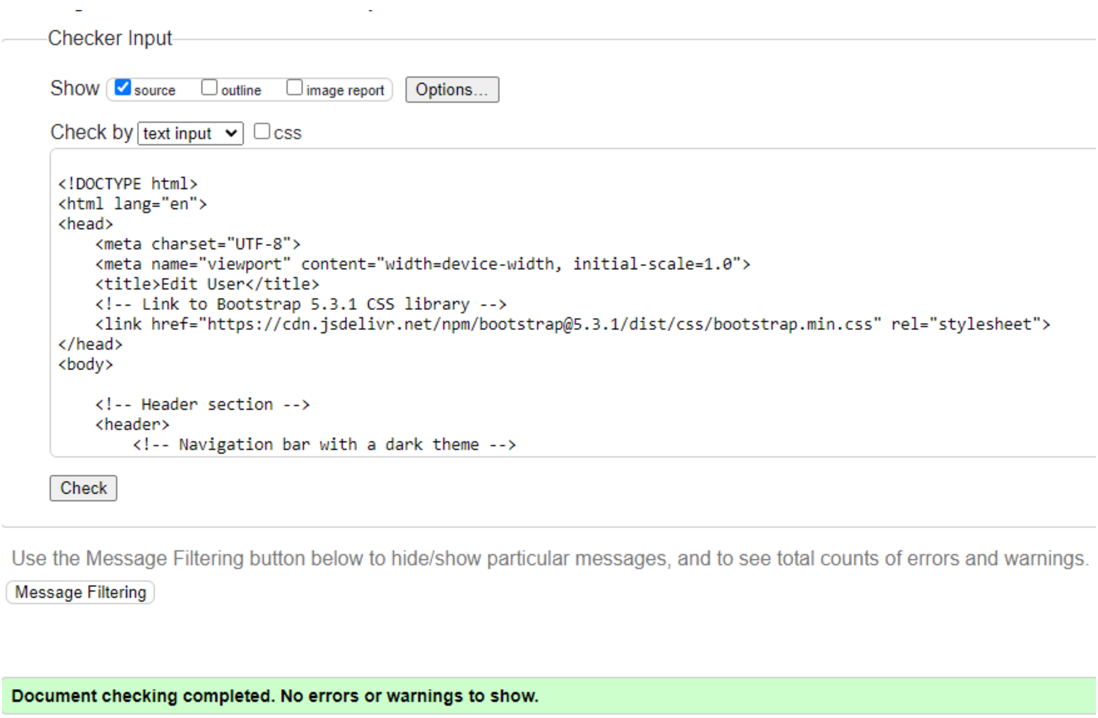

Checker Input
Show ☑ source □ outline □ image report
Options...
Check by text input □ css
Check
Message Filtering

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Edit Teacher</title>
    <!-- Link to Bootstrap 5.3.1 CSS library -->
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.1/dist/css/bootstrap.min.css" rel="stylesheet">
</head>
<body>

    <!-- Header section -->
```

Use the Message Filtering button below to hide/show particular messages, and to see total counts of errors and warnings.

Document checking completed. No errors or warnings to show.

Figure 19

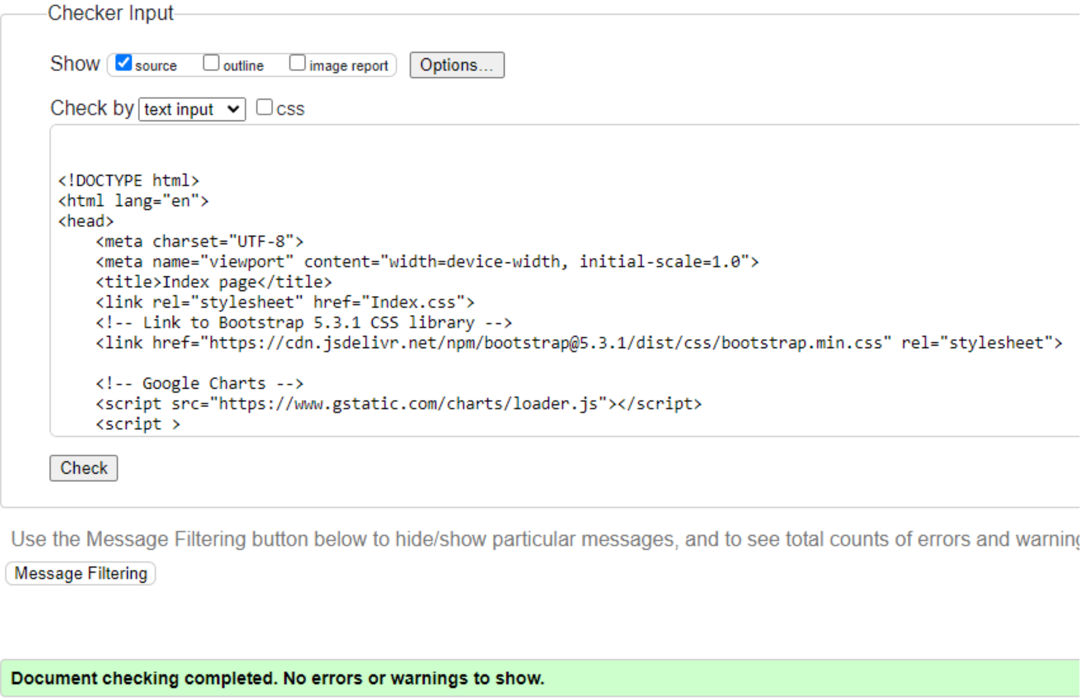

Checker Input
Show ☑ source □ outline □ image report
Options...
Check by text input □ css
Check
Message Filtering

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Edit User</title>
    <!-- Link to Bootstrap 5.3.1 CSS library -->
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.1/dist/css/bootstrap.min.css" rel="stylesheet">
</head>
<body>

    <!-- Header section -->
    <header>
        <!-- Navigation bar with a dark theme -->
```

Use the Message Filtering button below to hide/show particular messages, and to see total counts of errors and warnings.

Document checking completed. No errors or warnings to show.

Figure 20



Show ☑ source □ outline □ image report

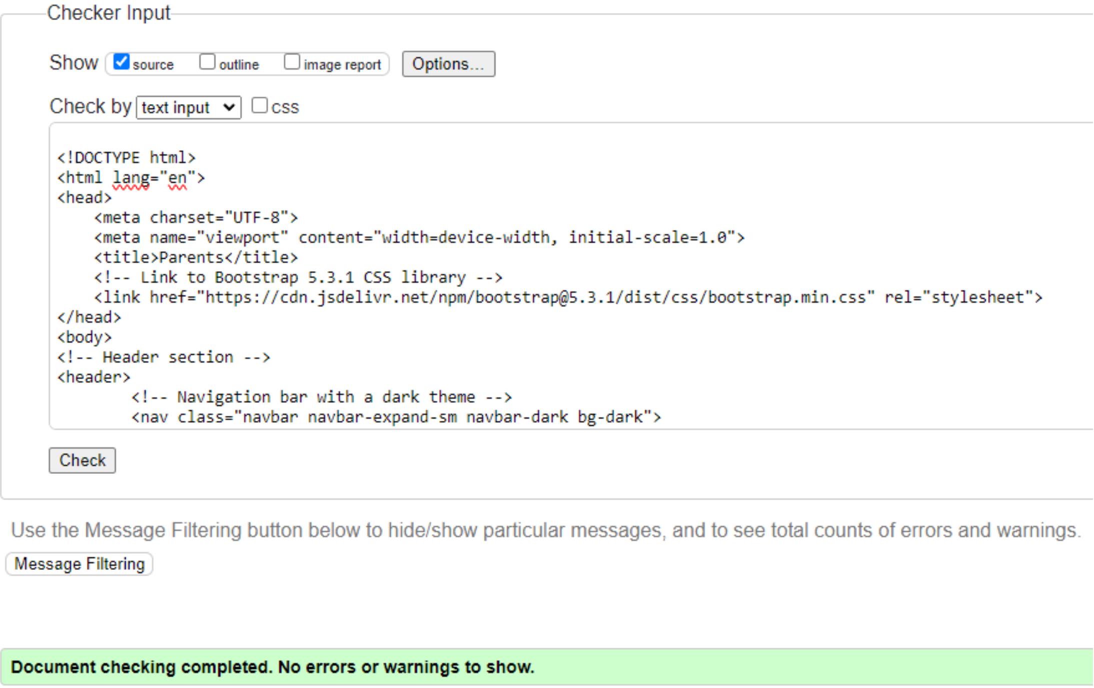
Check by text input ▾ □ css

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Index page</title>
    <link rel="stylesheet" href="Index.css">
    <!-- Link to Bootstrap 5.3.1 CSS library -->
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.1/dist/css/bootstrap.min.css" rel="stylesheet">

    <!-- Google Charts -->
    <script src="https://www.gstatic.com/charts/loader.js"></script>
    <script >
```

Use the Message Filtering button below to hide/show particular messages, and to see total counts of errors and warnin

Document checking completed. No errors or warnings to show.

Figure 21



Show ☑ source □ outline □ image report

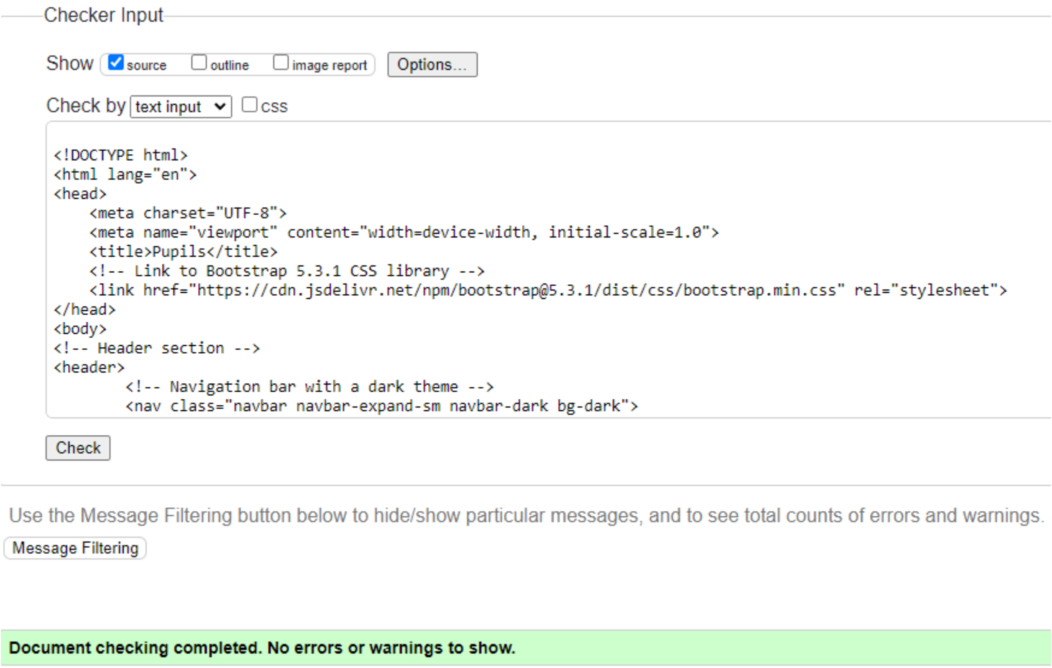
Check by text input ▾ □ css

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Parents</title>
    <!-- Link to Bootstrap 5.3.1 CSS library -->
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.1/dist/css/bootstrap.min.css" rel="stylesheet">
</head>
<body>
<!-- Header section -->
<header>
        <!-- Navigation bar with a dark theme -->
        <nav class="navbar navbar-expand-sm navbar-dark bg-dark">
```

Use the Message Filtering button below to hide/show particular messages, and to see total counts of errors and warnings.

Document checking completed. No errors or warnings to show.

Figure 22

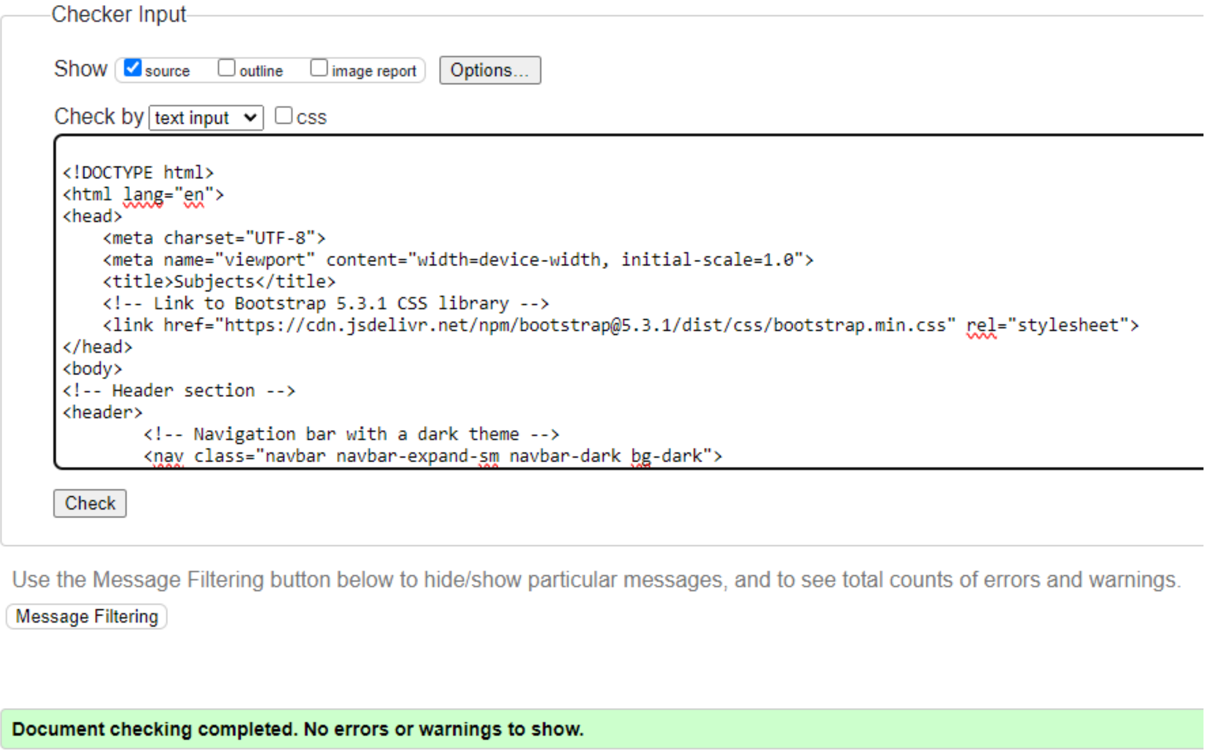

Checker Input
Show ☑ source □ outline □ image report Options...
Check by text input □ css
Check
Message Filtering

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Pupils</title>
    <!-- Link to Bootstrap 5.3.1 CSS library -->
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.1/dist/css/bootstrap.min.css" rel="stylesheet">
</head>
<body>
<!-- Header section -->
<header>
        <!-- Navigation bar with a dark theme -->
        <nav class="navbar navbar-expand-sm navbar-dark bg-dark">
```

Use the Message Filtering button below to hide/show particular messages, and to see total counts of errors and warnings.

Document checking completed. No errors or warnings to show.

Figure 23

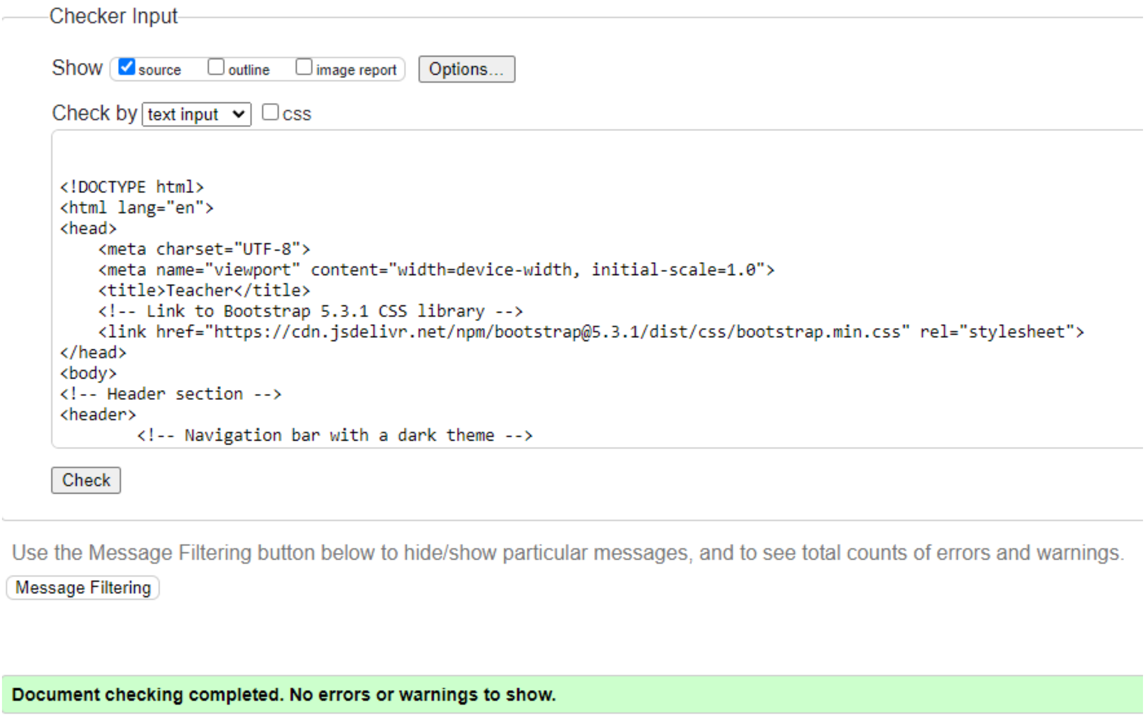

Checker Input
Show ☑ source □ outline □ image report Options...
Check by text input □ css
Check
Message Filtering

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Subjects</title>
    <!-- Link to Bootstrap 5.3.1 CSS library -->
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.1/dist/css/bootstrap.min.css" rel="stylesheet">
</head>
<body>
<!-- Header section -->
<header>
        <!-- Navigation bar with a dark theme -->
        <nav class="navbar navbar-expand-sm navbar-dark bg-dark">
```

Use the Message Filtering button below to hide/show particular messages, and to see total counts of errors and warnings.

Document checking completed. No errors or warnings to show.

Figure 24

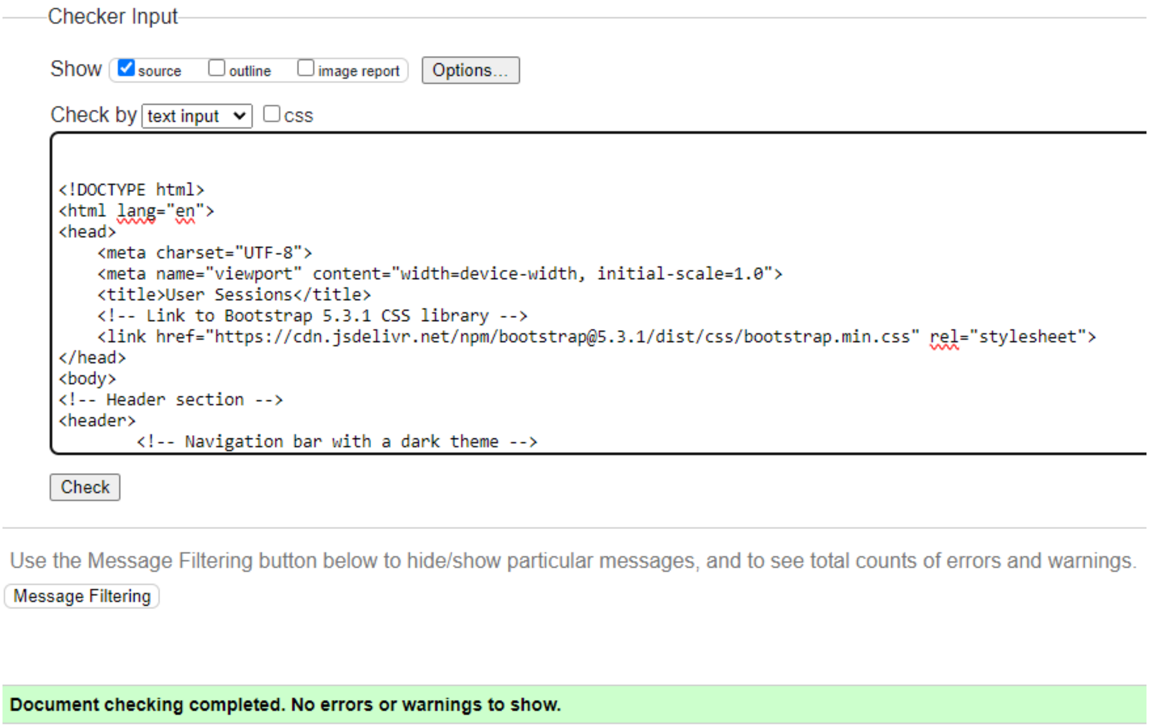

Checker Input
Show ☑ source □ outline □ image report
Options...
Check by text input □ css
Check
Message Filtering

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Teacher</title>
    <!-- Link to Bootstrap 5.3.1 CSS library -->
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.1/dist/css/bootstrap.min.css" rel="stylesheet">
</head>
<body>
<!-- Header section -->
<header>
        <!-- Navigation bar with a dark theme -->
```

Use the Message Filtering button below to hide/show particular messages, and to see total counts of errors and warnings.

Document checking completed. No errors or warnings to show.

Figure 25


Checker Input
Show ☑ source □ outline □ image report
Options...
Check by text input □ css
Check
Message Filtering

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>User Sessions</title>
    <!-- Link to Bootstrap 5.3.1 CSS library -->
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.1/dist/css/bootstrap.min.css" rel="stylesheet">
</head>
<body>
<!-- Header section -->
<header>
        <!-- Navigation bar with a dark theme -->
```

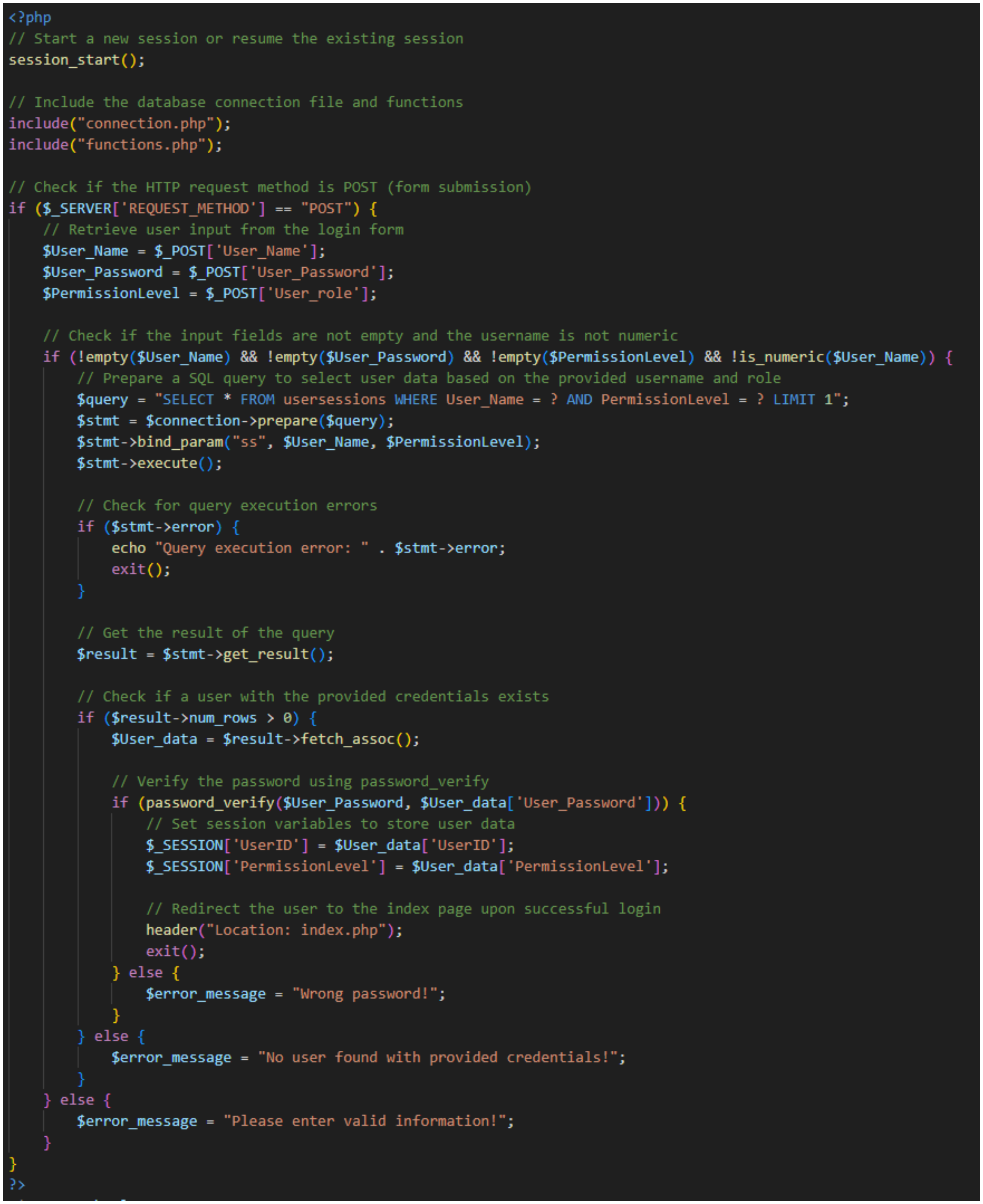
Use the Message Filtering button below to hide/show particular messages, and to see total counts of errors and warnings.

Document checking completed. No errors or warnings to show.

Figure 26

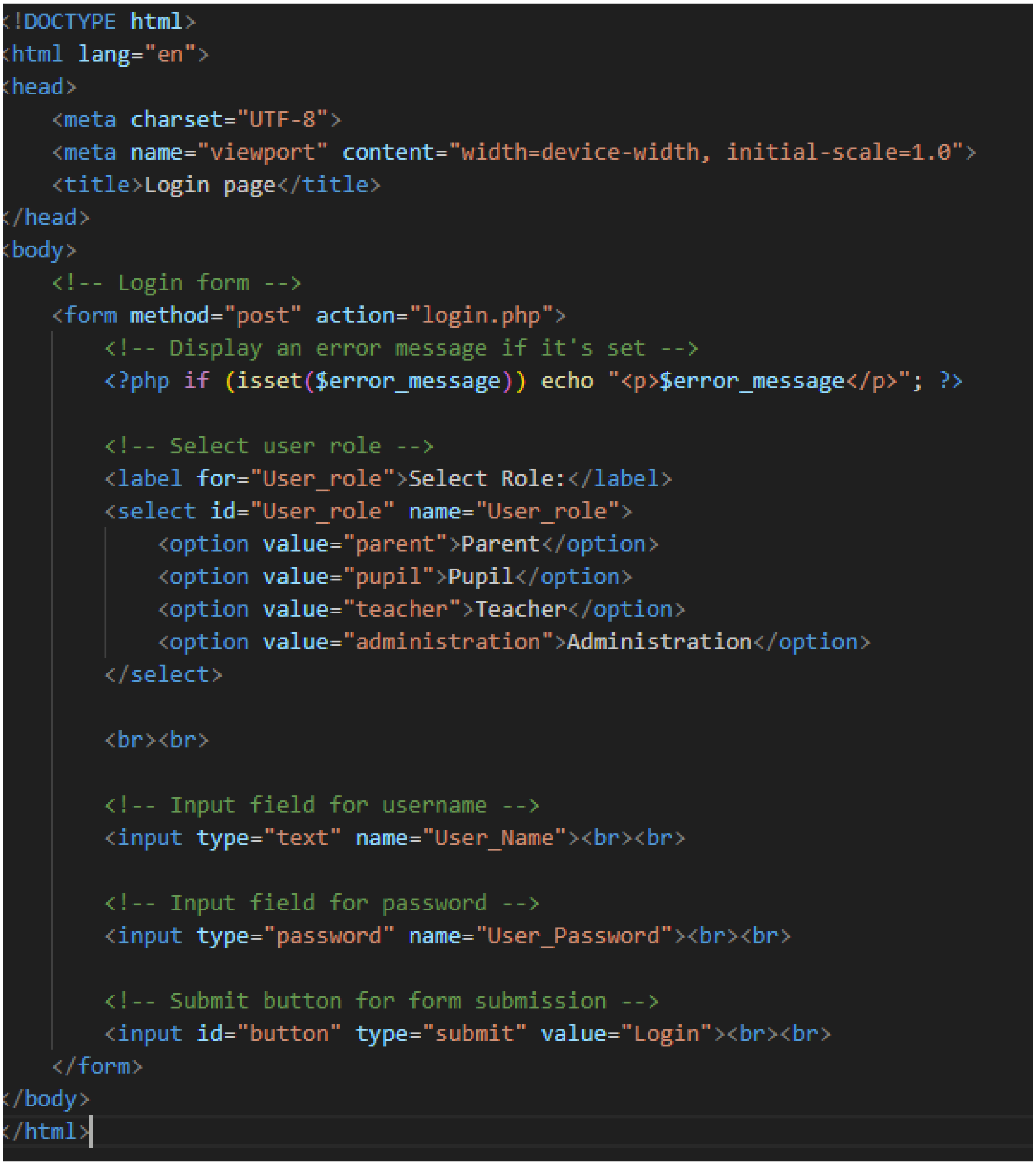
From Figure 1 to Figure 26 it shows that all the HTML is correct and there are no errors, it also shows that the structure is also correct and in the proper way.

Login Page:



```php
<?php
// Start a new session or resume the existing session
session_start();

// Include the database connection file and functions
include("connection.php");
include("functions.php");

// Check if the HTTP request method is POST (form submission)
if ($_SERVER['REQUEST_METHOD'] == "POST") {
    // Retrieve user input from the login form
    $User_Name = $_POST['User_Name'];
    $User_Password = $_POST['User_Password'];
    $PermissionLevel = $_POST['User_role'];

    // Check if the input fields are not empty and the username is not numeric
    if (!empty($User_Name) && !empty($User_Password) && !empty($PermissionLevel) && !is_numeric($User_Name)) {
        // Prepare a SQL query to select user data based on the provided username and role
        $query = "SELECT * FROM usersessions WHERE User_Name = ? AND PermissionLevel = ? LIMIT 1";
        $stmt = $connection->prepare($query);
        $stmt->bind_param("ss", $User_Name, $PermissionLevel);
        $stmt->execute();

        // Check for query execution errors
        if ($stmt->error) {
            echo "Query execution error: " . $stmt->error;
            exit();
        }

        // Get the result of the query
        $result = $stmt->get_result();

        // Check if a user with the provided credentials exists
        if ($result->num_rows > 0) {
            $User_data = $result->fetch_assoc();

            // Verify the password using password_verify
            if (password_verify($User_Password, $User_data['User_Password'])) {
                // Set session variables to store user data
                $_SESSION['UserID'] = $User_data['UserID'];
                $_SESSION['PermissionLevel'] = $User_data['PermissionLevel'];

                // Redirect the user to the index page upon successful login
                header("Location: index.php");
                exit();
            } else {
                $error_message = "Wrong password!";
            }
        } else {
            $error_message = "No user found with provided credentials!";
        }
    } else {
        $error_message = "Please enter valid information!";
    }
}
?>
```

Figure 27

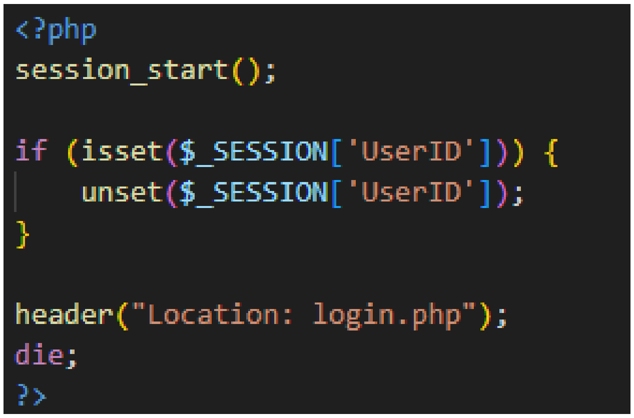


```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Login page</title>
</head>
<body>
    <!-- Login form -->
    <form method="post" action="login.php">
        <!-- Display an error message if it's set -->
        <?php if (isset($error_message)) echo "<p>$error_message</p>"; ?>

        <!-- Select user role -->
        <label for="User_role">Select Role:</label>
        <select id="User_role" name="User_role">
            <option value="parent">Parent</option>
            <option value="pupil">Pupil</option>
            <option value="teacher">Teacher</option>
            <option value="administration">Administration</option>
        </select>

        <br><br>

        <!-- Input field for username -->
        <input type="text" name="User_Name"><br><br>

        <!-- Input field for password -->
        <input type="password" name="User_Password"><br><br>

        <!-- Submit button for form submission -->
        <input id="button" type="submit" value="Login"><br><br>
    </form>
</body>
</html>
```

Figure 28

(*Signup and Login with PHP and MySQL*. [online])The login page consists of connecting to the database, getting the information from the form submission, and checking that it matches the data in the database. If it does match it will let the user proceed, if not, an error message will appear.

Logout Page:

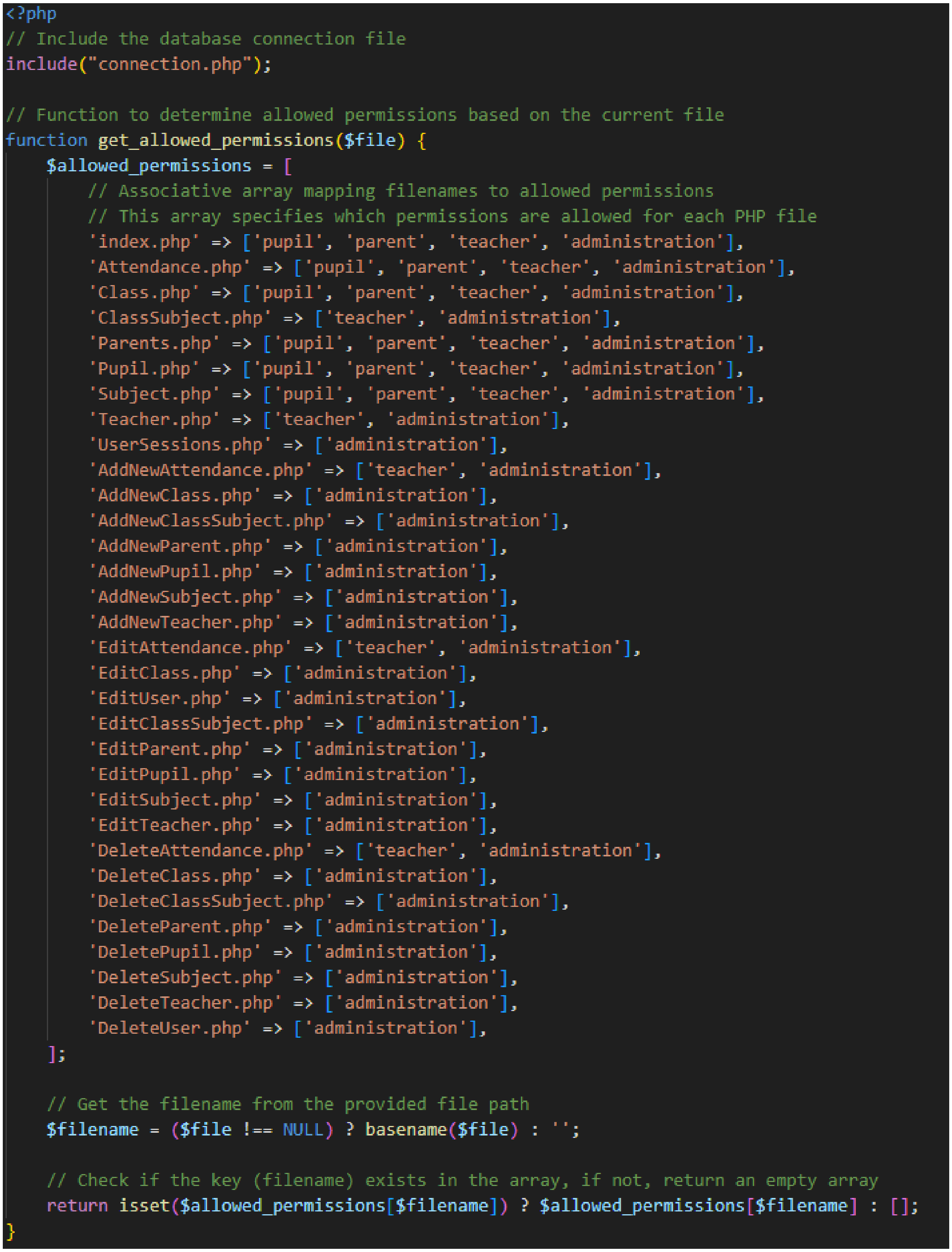


```php
<?php
session_start();

if (isset($_SESSION['UserID'])) {
    unset($_SESSION['UserID']);
}

header("Location: login.php");
die;
?>
```

Figure 29

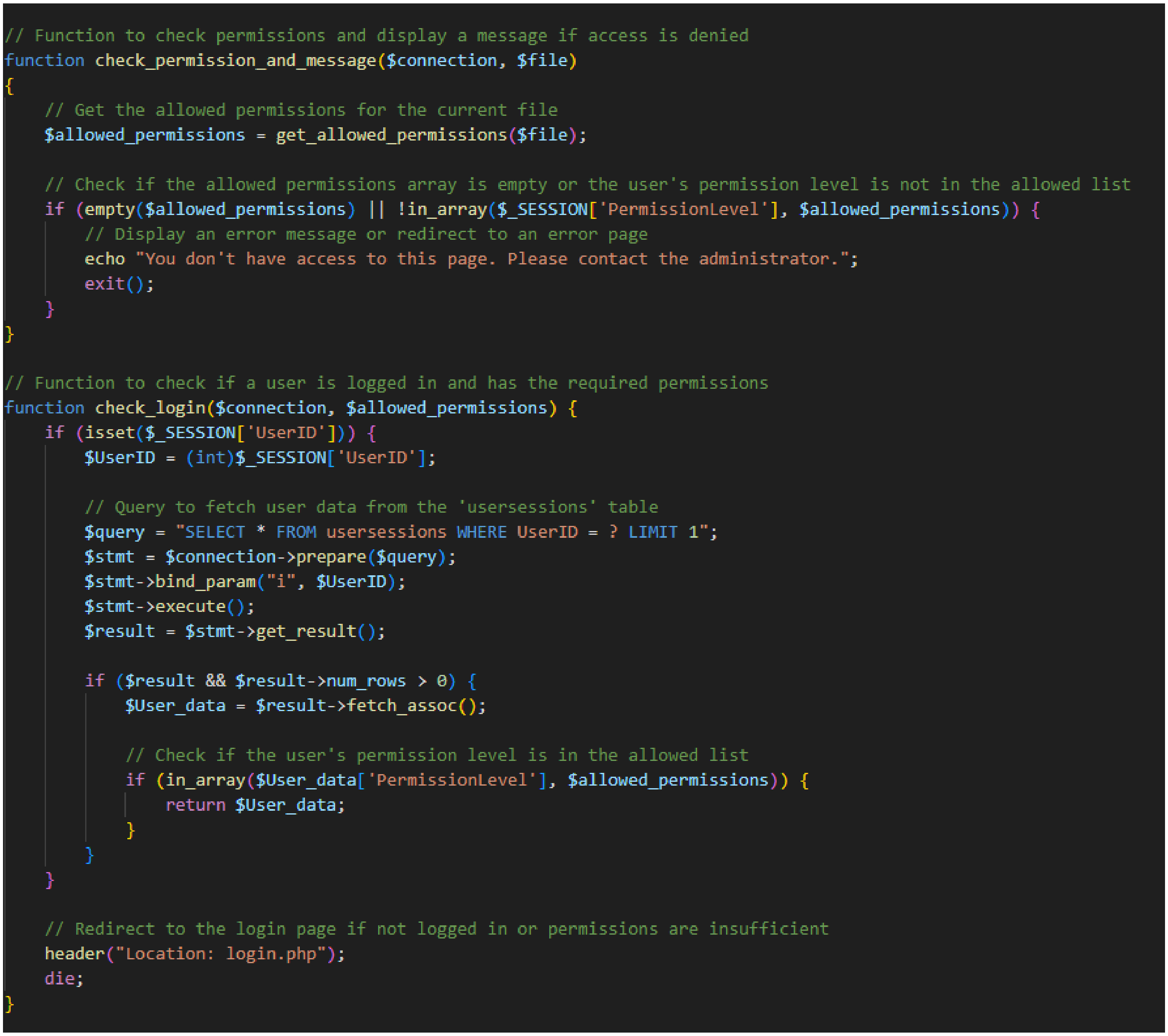
The logout page consists of ending the session.

Functions Page:



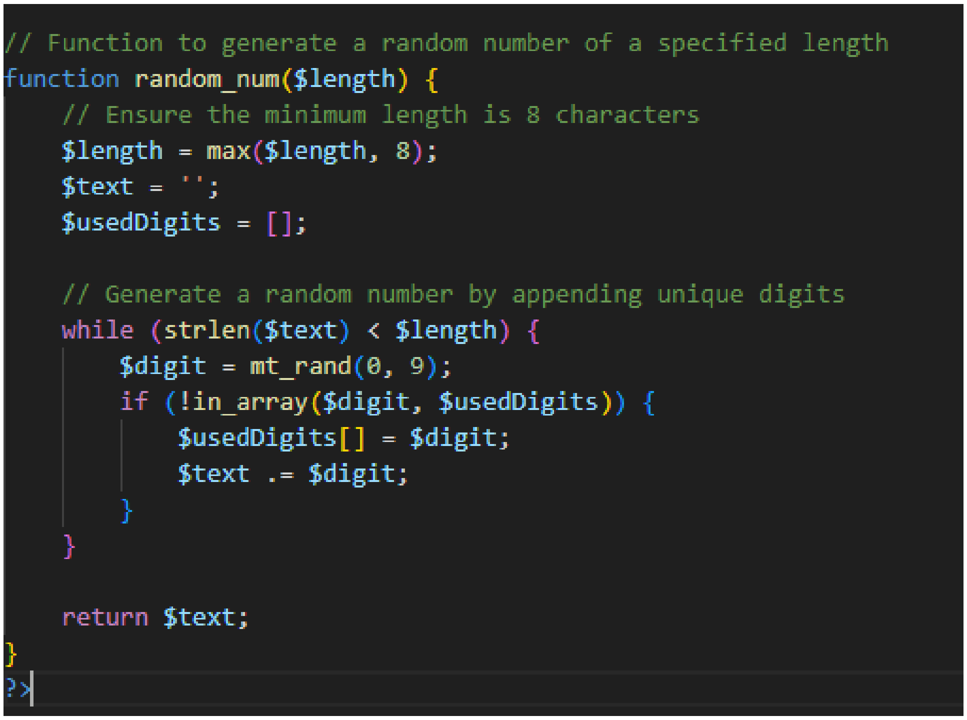
```php
<?php
// Include the database connection file
include("connection.php");

// Function to determine allowed permissions based on the current file
function get_allowed_permissions($file) {
    $allowed_permissions = [
        // Associative array mapping filenames to allowed permissions
        // This array specifies which permissions are allowed for each PHP file
        'index.php' => ['pupil', 'parent', 'teacher', 'administration'],
        'Attendance.php' => ['pupil', 'parent', 'teacher', 'administration'],
        'Class.php' => ['pupil', 'parent', 'teacher', 'administration'],
        'ClassSubject.php' => ['teacher', 'administration'],
        'Parents.php' => ['pupil', 'parent', 'teacher', 'administration'],
        'Pupil.php' => ['pupil', 'parent', 'teacher', 'administration'],
        'Subject.php' => ['pupil', 'parent', 'teacher', 'administration'],
        'Teacher.php' => ['teacher', 'administration'],
        'UserSessions.php' => ['administration'],
        'AddNewAttendance.php' => ['teacher', 'administration'],
        'AddNewClass.php' => ['administration'],
        'AddNewClassSubject.php' => ['administration'],
        'AddNewParent.php' => ['administration'],
        'AddNewPupil.php' => ['administration'],
        'AddNewSubject.php' => ['administration'],
        'AddNewTeacher.php' => ['administration'],
        'EditAttendance.php' => ['teacher', 'administration'],
        'EditClass.php' => ['administration'],
        'EditUser.php' => ['administration'],
        'EditClassSubject.php' => ['administration'],
        'EditParent.php' => ['administration'],
        'EditPupil.php' => ['administration'],
        'EditSubject.php' => ['administration'],
        'EditTeacher.php' => ['administration'],
        'DeleteAttendance.php' => ['teacher', 'administration'],
        'DeleteClass.php' => ['administration'],
        'DeleteClassSubject.php' => ['administration'],
        'DeleteParent.php' => ['administration'],
        'DeletePupil.php' => ['administration'],
        'DeleteSubject.php' => ['administration'],
        'DeleteTeacher.php' => ['administration'],
        'DeleteUser.php' => ['administration'],
    ];

    // Get the filename from the provided file path
    $filename = ($file !== NULL) ? basename($file) : '';

    // Check if the key (filename) exists in the array, if not, return an empty array
    return isset($allowed_permissions[$filename]) ? $allowed_permissions[$filename] : [];
}
```

Figure 30



```php
// Function to check permissions and display a message if access is denied
function check_permission_and_message($connection, $file)
{
    // Get the allowed permissions for the current file
    $allowed_permissions = get_allowed_permissions($file);

    // Check if the allowed permissions array is empty or the user's permission level is not in the allowed list
    if (empty($allowed_permissions) || !in_array($_SESSION['PermissionLevel'], $allowed_permissions)) {
        // Display an error message or redirect to an error page
        echo "You don't have access to this page. Please contact the administrator.";
        exit();
    }
}

// Function to check if a user is logged in and has the required permissions
function check_login($connection, $allowed_permissions) {
    if (isset($_SESSION['UserID'])) {
        $UserID = (int)$_SESSION['UserID'];

        // Query to fetch user data from the 'usersessions' table
        $query = "SELECT * FROM usersessions WHERE UserID = ? LIMIT 1";
        $stmt = $connection->prepare($query);
        $stmt->bind_param("i", $UserID);
        $stmt->execute();
        $result = $stmt->get_result();

        if ($result && $result->num_rows > 0) {
            $User_data = $result->fetch_assoc();

            // Check if the user's permission level is in the allowed list
            if (in_array($User_data['PermissionLevel'], $allowed_permissions)) {
                return $User_data;
            }
        }
    }

    // Redirect to the login page if not logged in or permissions are insufficient
    header("Location: login.php");
    die;
}
```

Figure 31



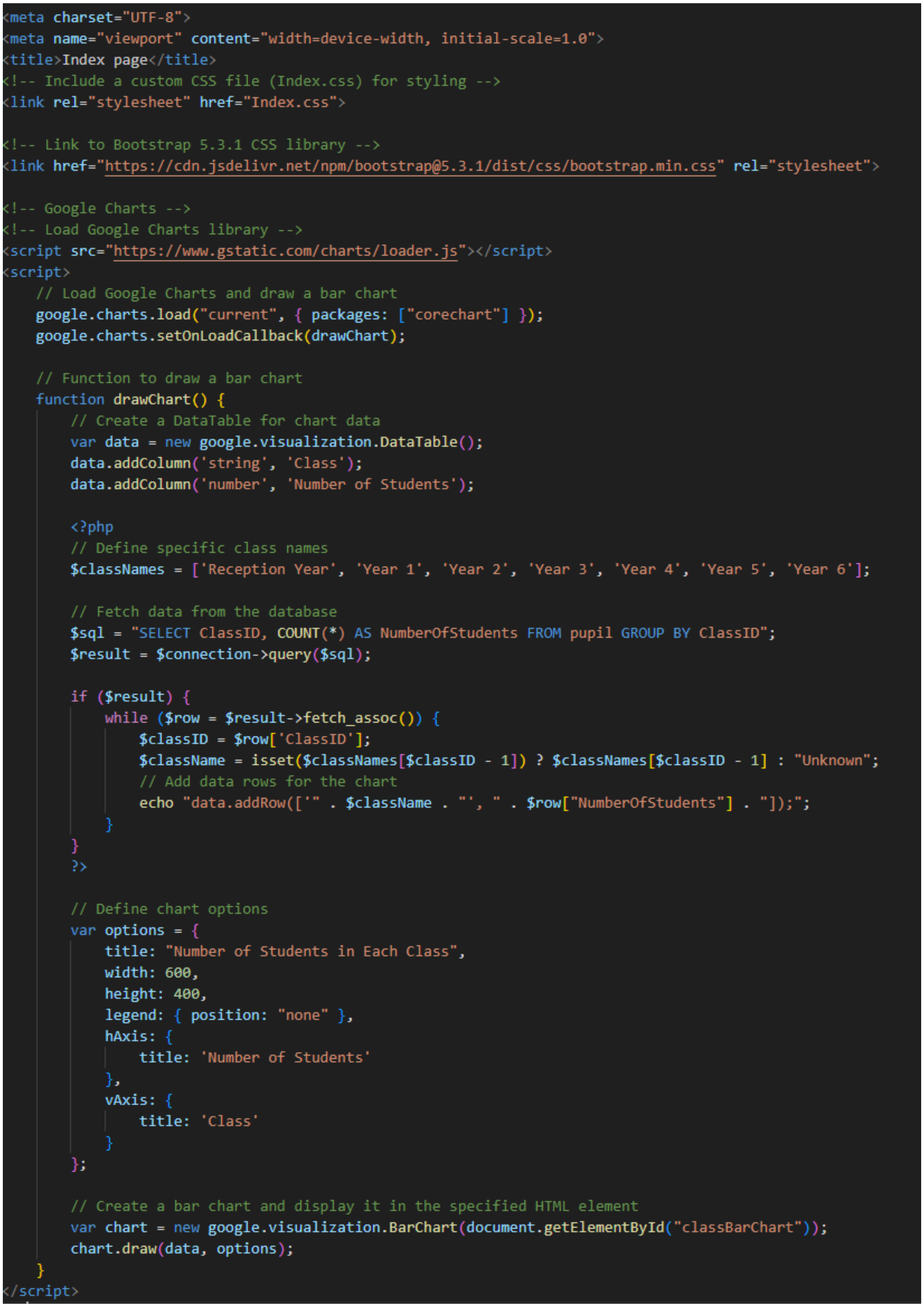
```
// Function to generate a random number of a specified length
function random_num($length) {
    // Ensure the minimum length is 8 characters
    $length = max($length, 8);
    $text = '';
    $usedDigits = [];

    // Generate a random number by appending unique digits
    while (strlen($text) < $length) {
        $digit = mt_rand(0, 9);
        if (!in_array($digit, $usedDigits)) {
            $usedDigits[] = $digit;
            $text .= $digit;
        }
    }

    return $text;
}
?>
```

Figure 32

(*Signup and Login with PHP and MySQL*. [online])The function page consists of various functions that are implemented on each page. It has a permission level function that restricts certain users from entering and using some of the pages according to their user type.

It also has a function that creates random numbers or the UserId that are not the same. The login features were inspired by video that will be linked later in the references.

Index Page:

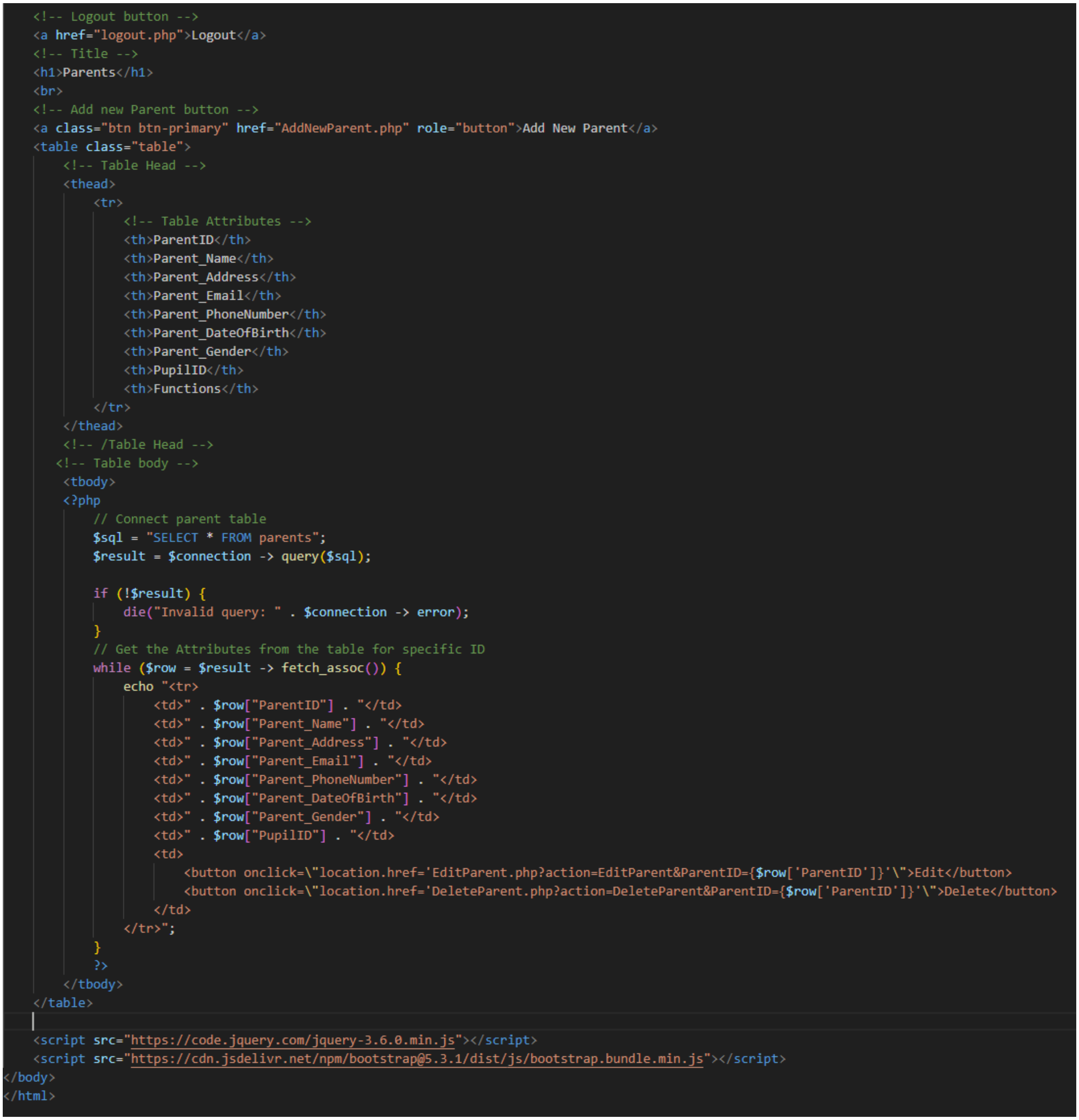


```
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Index page</title>
<!-- Include a custom CSS file (Index.css) for styling -->
<link rel="stylesheet" href="Index.css">

<!-- Link to Bootstrap 5.3.1 CSS library -->
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.1/dist/css/bootstrap.min.css" rel="stylesheet">

<!-- Google Charts -->
<!-- Load Google Charts library -->
<script src="https://www.gstatic.com/charts/loader.js"></script>
<script>
    // Load Google Charts and draw a bar chart
    google.charts.load("current", { packages: ["corechart"] });
    google.charts.setOnLoadCallback(drawChart);

    // Function to draw a bar chart
    function drawChart() {
        // Create a DataTable for chart data
        var data = new google.visualization.DataTable();
        data.addColumn('string', 'Class');
        data.addColumn('number', 'Number of Students');

        <?php
        // Define specific class names
        $classNames = ['Reception Year', 'Year 1', 'Year 2', 'Year 3', 'Year 4', 'Year 5', 'Year 6'];

        // Fetch data from the database
        $sql = "SELECT ClassID, COUNT(*) AS NumberOfStudents FROM pupil GROUP BY ClassID";
        $result = $connection->query($sql);

        if ($result) {
            while ($row = $result->fetch_assoc()) {
                $classID = $row['ClassID'];
                $className = isset($classNames[$classID - 1]) ? $classNames[$classID - 1] : "Unknown";
                // Add data rows for the chart
                echo "data.addRow(['" . $className . "', " . $row["NumberOfStudents"] . "]);";
            }
        }
        ?>

        // Define chart options
        var options = {
            title: "Number of Students in Each Class",
            width: 600,
            height: 400,
            legend: { position: "none" },
            hAxis: {
                title: 'Number of Students'
            },
            vAxis: {
                title: 'Class'
            }
        };

        // Create a bar chart and display it in the specified HTML element
        var chart = new google.visualization.BarChart(document.getElementById("classBarChart"));
        chart.draw(data, options);
    }
</script>
```

Figure 33

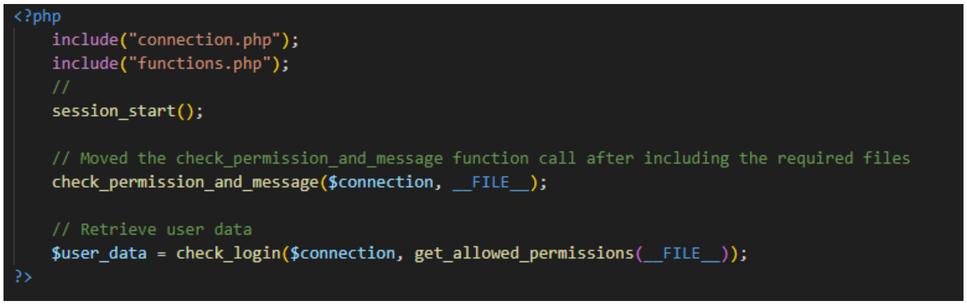
(Google Developers. (n.d.). *Charts*. [online])The index page consists of navigation bar from bootstrap with the links of the other pages and this chart from google that was changed to adapt this assignment and it shows how many students ae in a class.

Table Pages:



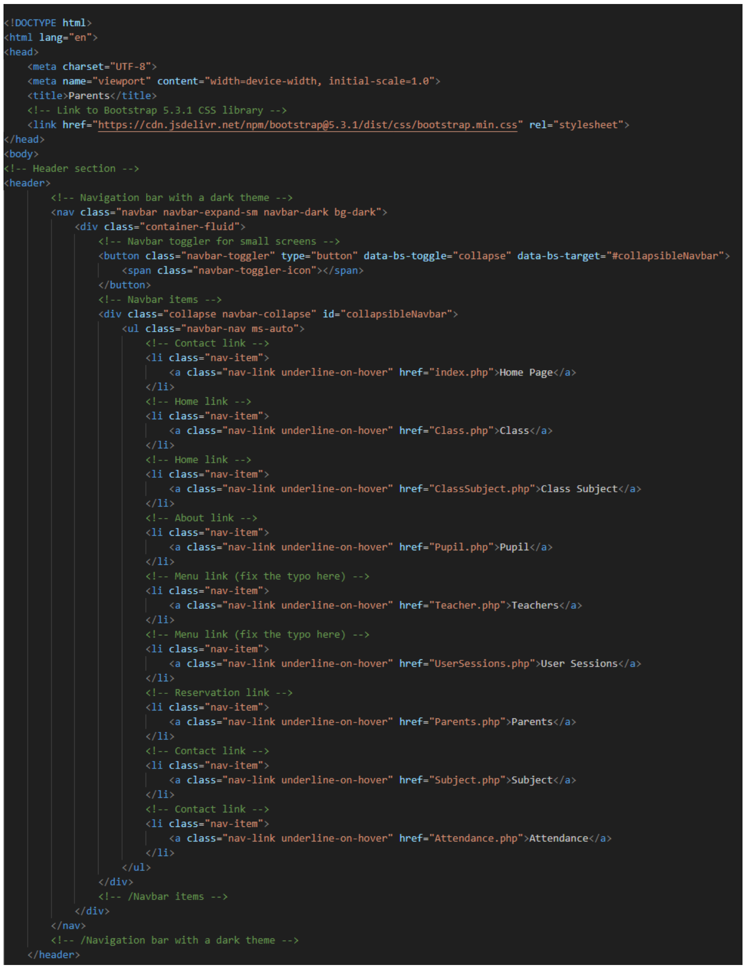
```
    <!-- Logout button -->
    <a href="logout.php">Logout</a>
    <!-- Title -->
    <h1>Parents</h1>
    <br>
    <!-- Add new Parent button -->
    <a class="btn btn-primary" href="AddNewParent.php" role="button">Add New Parent</a>
    <table class="table">
        <!-- Table Head -->
        <thead>
            <tr>
                <!-- Table Attributes -->
                <th>ParentID</th>
                <th>Parent_Name</th>
                <th>Parent_Address</th>
                <th>Parent_Email</th>
                <th>Parent_PhoneNumber</th>
                <th>Parent_DateOfBirth</th>
                <th>Parent_Gender</th>
                <th>PupilID</th>
                <th>Functions</th>
            </tr>
        </thead>
        <!-- /Table Head -->
       <!-- Table body -->
        <tbody>
        <?php
            // Connect parent table
            $sql = "SELECT * FROM parents";
            $result = $connection -> query($sql);

            if (!$result) {
                die("Invalid query: " . $connection -> error);
            }
            // Get the Attributes from the table for specific ID
            while ($row = $result -> fetch_assoc()) {
                echo "<tr>
                    <td>" . $row["ParentID"] . "</td>
                    <td>" . $row["Parent_Name"] . "</td>
                    <td>" . $row["Parent_Address"] . "</td>
                    <td>" . $row["Parent_Email"] . "</td>
                    <td>" . $row["Parent_PhoneNumber"] . "</td>
                    <td>" . $row["Parent_DateOfBirth"] . "</td>
                    <td>" . $row["Parent_Gender"] . "</td>
                    <td>" . $row["PupilID"] . "</td>
                    <td>
                        <button onclick=\"location.href='EditParent.php?action=EditParent&ParentID={$row['ParentID']}'\">Edit</button>
                        <button onclick=\"location.href='DeleteParent.php?action=DeleteParent&ParentID={$row['ParentID']}'\">Delete</button>
                    </td>
                </tr>";
            }
            ?>
        </tbody>
    </table>

    <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.1/dist/js/bootstrap.bundle.min.js"></script>
</body>
</html>
```

Figure 34



```php
<?php
    include("connection.php");
    include("functions.php");
    //
    session_start();

    // Moved the check_permission_and_message function call after including the required files
    check_permission_and_message($connection, __FILE__);

    // Retrieve user data
    $user_data = check_login($connection, get_allowed_permissions(__FILE__));
?>
```

Figure 35

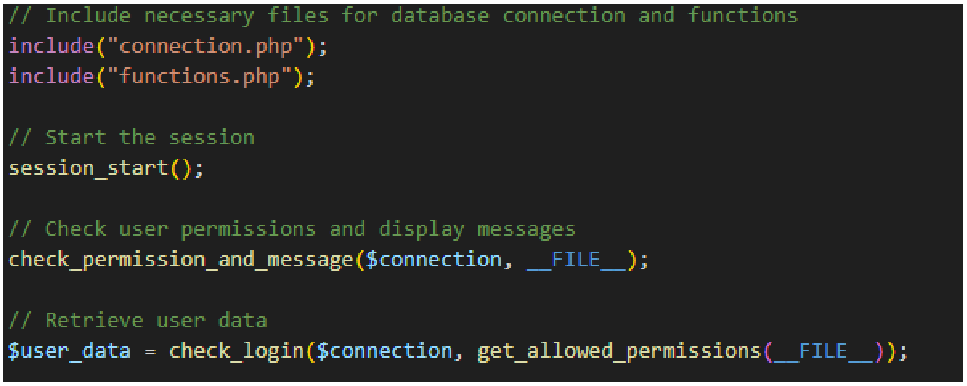


```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Parents</title>
    <!-- Link to Bootstrap 5.3.1 CSS library -->
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.1/dist/css/bootstrap.min.css" rel="stylesheet">
</head>
<body>
<!-- Header section -->
<header>
        <!-- Navigation bar with a dark theme -->
        <nav class="navbar navbar-expand-sm navbar-dark bg-dark">
            <div class="container-fluid">
                <!-- Navbar toggler for small screens -->
                <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#collapsibleNavbar">
                    <span class="navbar-toggler-icon"></span>
                </button>
                <!-- Navbar items -->
                <div class="collapse navbar-collapse" id="collapsibleNavbar">
                    <ul class="navbar-nav ms-auto">
                        <!-- Contact link -->
                        <li class="nav-item">
                            <a class="nav-link underline-on-hover" href="index.php">Home Page</a>
                        </li>
                        <!-- Home link -->
                        <li class="nav-item">
                            <a class="nav-link underline-on-hover" href="Class.php">Class</a>
                        </li>
                        <!-- Home link -->
                        <li class="nav-item">
                            <a class="nav-link underline-on-hover" href="ClassSubject.php">Class Subject</a>
                        </li>
                        <!-- About link -->
                        <li class="nav-item">
                            <a class="nav-link underline-on-hover" href="Pupil.php">Pupil</a>
                        </li>
                        <!-- Menu link (fix the typo here) -->
                        <li class="nav-item">
                            <a class="nav-link underline-on-hover" href="Teacher.php">Teachers</a>
                        </li>
                        <!-- Menu link (fix the typo here) -->
                        <li class="nav-item">
                            <a class="nav-link underline-on-hover" href="UserSessions.php">User Sessions</a>
                        </li>
                        <!-- Reservation link -->
                        <li class="nav-item">
                            <a class="nav-link underline-on-hover" href="Parents.php">Parents</a>
                        </li>
                        <!-- Contact link -->
                        <li class="nav-item">
                            <a class="nav-link underline-on-hover" href="Subject.php">Subject</a>
                        </li>
                        <!-- Contact link -->
                        <li class="nav-item">
                            <a class="nav-link underline-on-hover" href="Attendance.php">Attendance</a>
                        </li>
                    </ul>
                </div>
                <!-- /Navbar items -->
            </div>
        </nav>
        <!-- /Navigation bar with a dark theme -->
    </header>
```

Figure 36

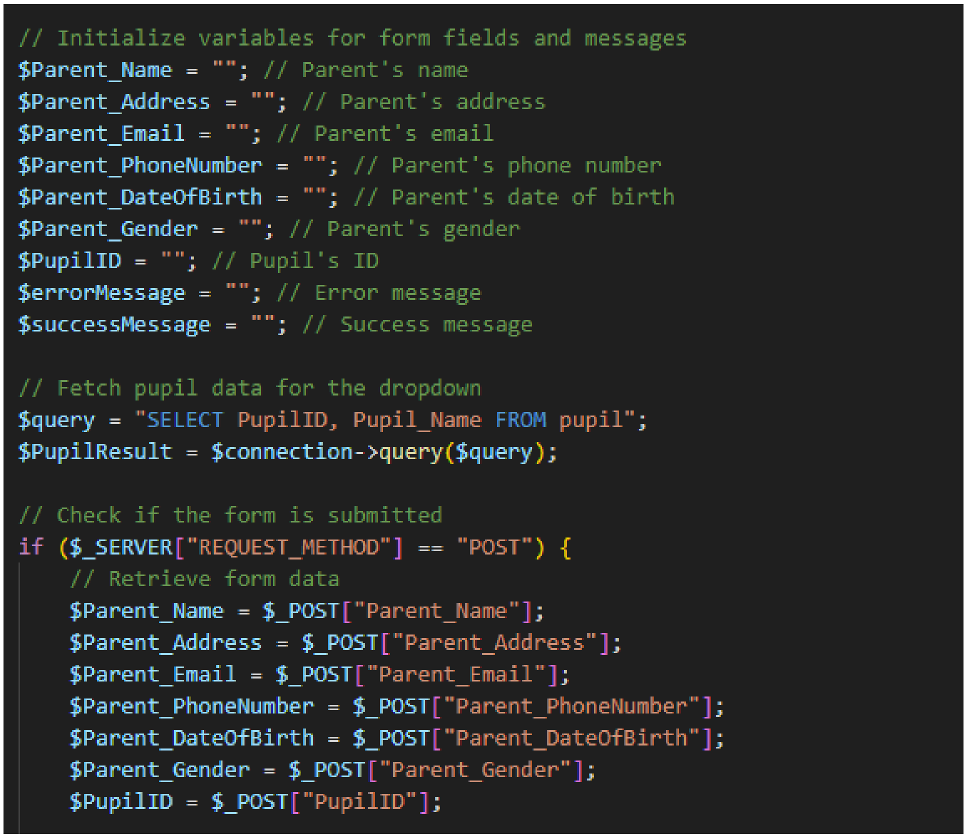
((BoostMyTool (2022). *PHP and MySQL with CRUD Operations: Create, Read, Update, Delete*. *YouTube*.)There is only 1 table shown here, but all the other tables are similar, they just have different var names. It consists of a session that checks what type of user you are, and it will let those that are allowed. It also has a nav bar to navigate to other pages. Finally, it has a table containing the head which has the titles of the attributes and a body which pulls the data from the database and inserts it into the body of the table.

Add new record:



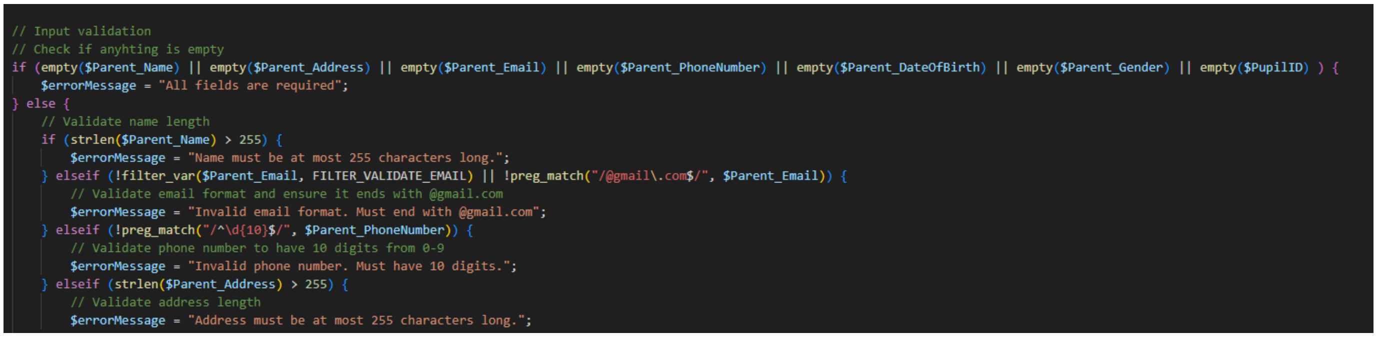
```php
// Include necessary files for database connection and functions
include("connection.php");
include("functions.php");

// Start the session
session_start();

// Check user permissions and display messages
check_permission_and_message($connection, __FILE__);

// Retrieve user data
$user_data = check_login($connection, get_allowed_permissions(__FILE__));
```

Figure 37



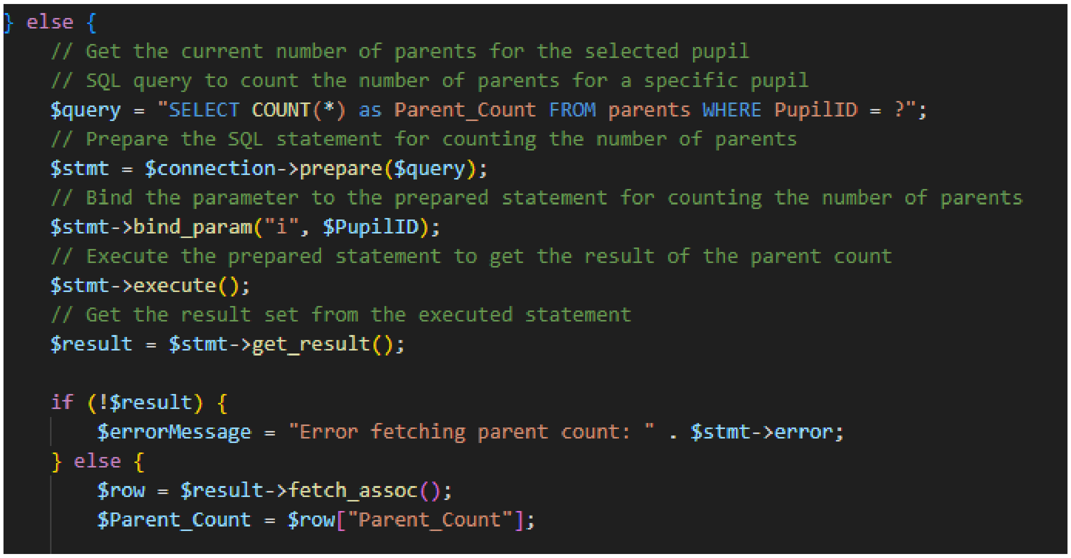
```php
// Initialize variables for form fields and messages
$Parent_Name = ""; // Parent's name
$Parent_Address = ""; // Parent's address
$Parent_Email = ""; // Parent's email
$Parent_PhoneNumber = ""; // Parent's phone number
$Parent_DateOfBirth = ""; // Parent's date of birth
$Parent_Gender = ""; // Parent's gender
$PupilID = ""; // Pupil's ID
$errorMessage = ""; // Error message
$successMessage = ""; // Success message

// Fetch pupil data for the dropdown
$query = "SELECT PupilID, Pupil_Name FROM pupil";
$PupilResult = $connection->query($query);

// Check if the form is submitted
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    // Retrieve form data
    $Parent_Name = $_POST["Parent_Name"];
    $Parent_Address = $_POST["Parent_Address"];
    $Parent_Email = $_POST["Parent_Email"];
    $Parent_PhoneNumber = $_POST["Parent_PhoneNumber"];
    $Parent_DateOfBirth = $_POST["Parent_DateOfBirth"];
    $Parent_Gender = $_POST["Parent_Gender"];
    $PupilID = $_POST["PupilID"];
```

Figure 38



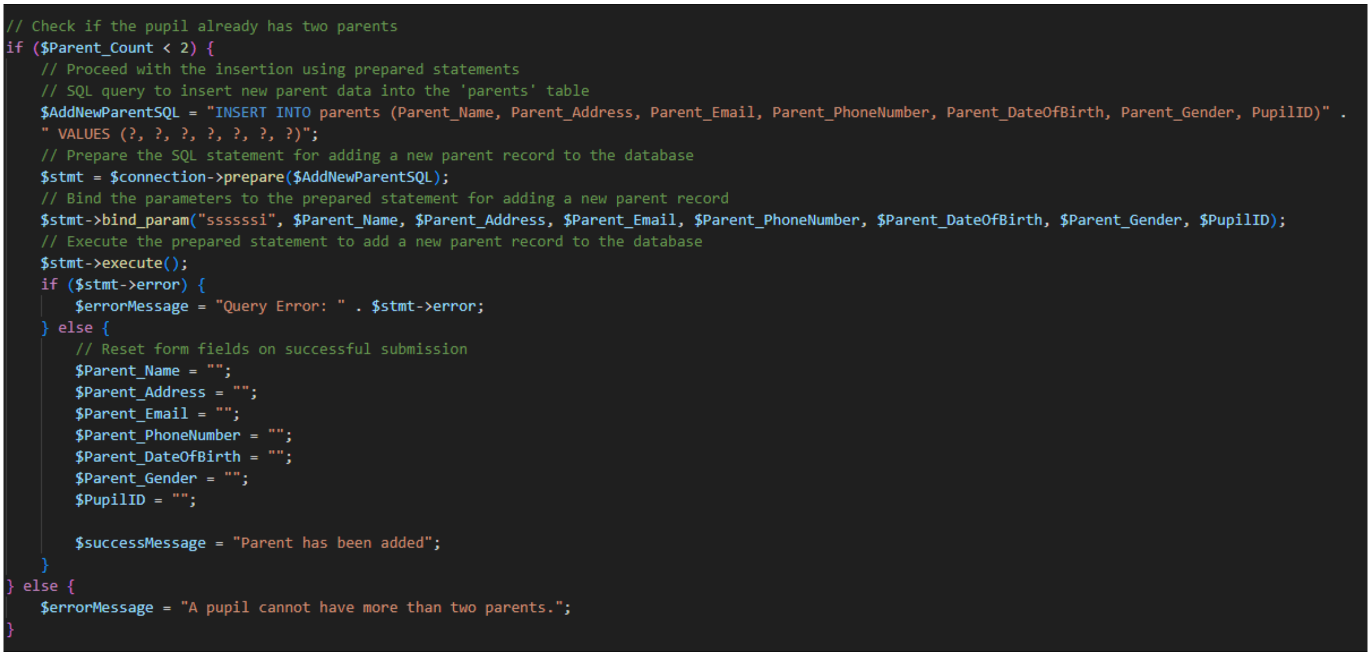
```php
// Input validation
// Check if anyhting is empty
if (empty($Parent_Name) || empty($Parent_Address) || empty($Parent_Email) || empty($Parent_PhoneNumber) || empty($Parent_DateOfBirth) || empty($Parent_Gender) || empty($PupilID) ) {
    $errorMessage = "All fields are required";
} else {
    // Validate name length
    if (strlen($Parent_Name) > 255) {
        $errorMessage = "Name must be at most 255 characters long.";
    } elseif (!filter_var($Parent_Email, FILTER_VALIDATE_EMAIL) || !preg_match("/@gmail\.com$/", $Parent_Email)) {
        // Validate email format and ensure it ends with @gmail.com
        $errorMessage = "Invalid email format. Must end with @gmail.com";
    } elseif (!preg_match("/^\d{10}$/", $Parent_PhoneNumber)) {
        // Validate phone number to have 10 digits from 0-9
        $errorMessage = "Invalid phone number. Must have 10 digits.";
    } elseif (strlen($Parent_Address) > 255) {
        // Validate address length
        $errorMessage = "Address must be at most 255 characters long.";
```

Figure 39



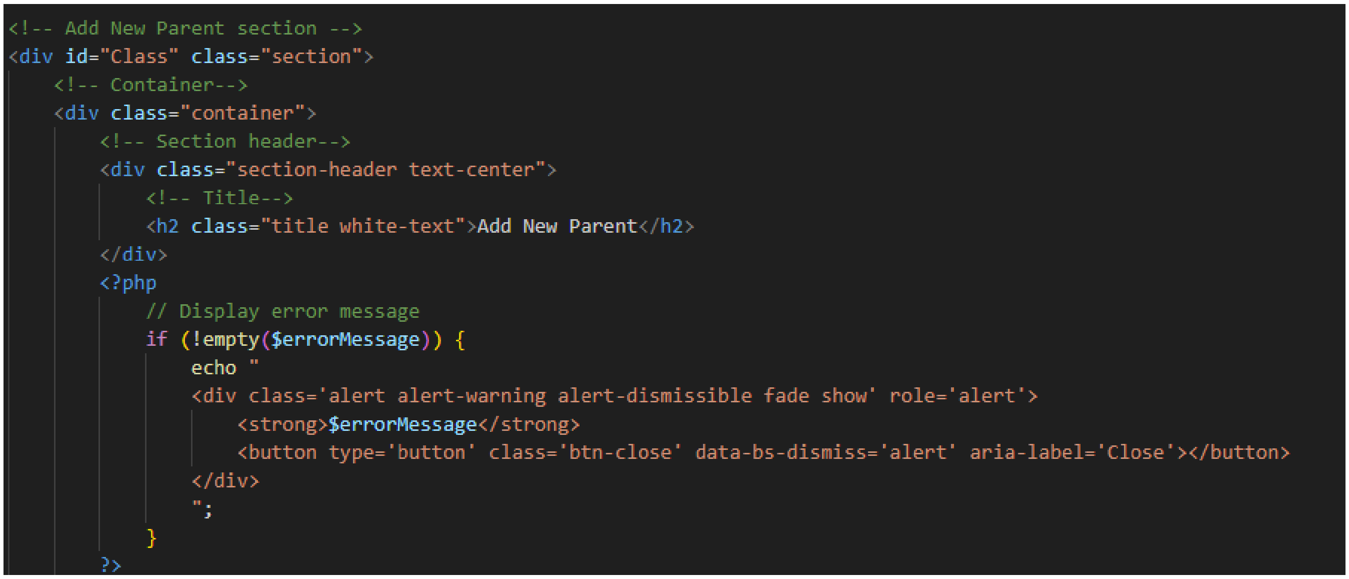
```php
} else {
    // Get the current number of parents for the selected pupil
    // SQL query to count the number of parents for a specific pupil
    $query = "SELECT COUNT(*) as Parent_Count FROM parents WHERE PupilID = ?";
    // Prepare the SQL statement for counting the number of parents
    $stmt = $connection->prepare($query);
    // Bind the parameter to the prepared statement for counting the number of parents
    $stmt->bind_param("i", $PupilID);
    // Execute the prepared statement to get the result of the parent count
    $stmt->execute();
    // Get the result set from the executed statement
    $result = $stmt->get_result();

    if (!$result) {
        $errorMessage = "Error fetching parent count: " . $stmt->error;
    } else {
        $row = $result->fetch_assoc();
        $Parent_Count = $row["Parent_Count"];
```

Figure 40



```php
// Check if the pupil already has two parents
if ($Parent_Count < 2) {
    // Proceed with the insertion using prepared statements
    // SQL query to insert new parent data into the 'parents' table
    $AddNewParentSQL = "INSERT INTO parents (Parent_Name, Parent_Address, Parent_Email, Parent_PhoneNumber, Parent_DateOfBirth, Parent_Gender, PupilID)" .
    " VALUES (?, ?, ?, ?, ?, ?, ?)";
    // Prepare the SQL statement for adding a new parent record to the database
    $stmt = $connection->prepare($AddNewParentSQL);
    // Bind the parameters to the prepared statement for adding a new parent record
    $stmt->bind_param("ssssssi", $Parent_Name, $Parent_Address, $Parent_Email, $Parent_PhoneNumber, $Parent_DateOfBirth, $Parent_Gender, $PupilID);
    // Execute the prepared statement to add a new parent record to the database
    $stmt->execute();
    if ($stmt->error) {
        $errorMessage = "Query Error: " . $stmt->error;
    } else {
        // Reset form fields on successful submission
        $Parent_Name = "";
        $Parent_Address = "";
        $Parent_Email = "";
        $Parent_PhoneNumber = "";
        $Parent_DateOfBirth = "";
        $Parent_Gender = "";
        $PupilID = "";

        $successMessage = "Parent has been added";
    }
} else {
    $errorMessage = "A pupil cannot have more than two parents.";
}
```

Figure 41

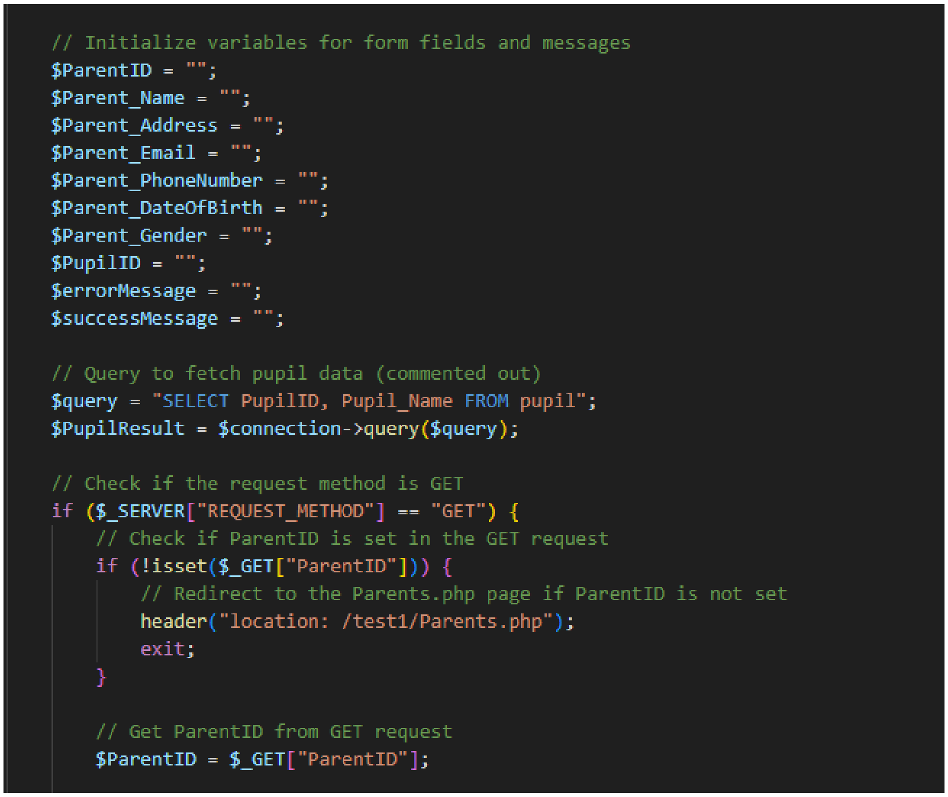


```
<!-- Add New Parent section -->
<div id="Class" class="section">
    <!-- Container-->
    <div class="container">
        <!-- Section header-->
        <div class="section-header text-center">
            <!-- Title-->
            <h2 class="title white-text">Add New Parent</h2>
        </div>
        <?php
            // Display error message
            if (!empty($errorMessage)) {
                echo "
                <div class='alert alert-warning alert-dismissible fade show' role='alert'>
                    <strong>$errorMessage</strong>
                    <button type='button' class='btn-close' data-bs-dismiss='alert' aria-label='Close'></button>
                </div>
                ";
            }
        ?>
```

Figure 42

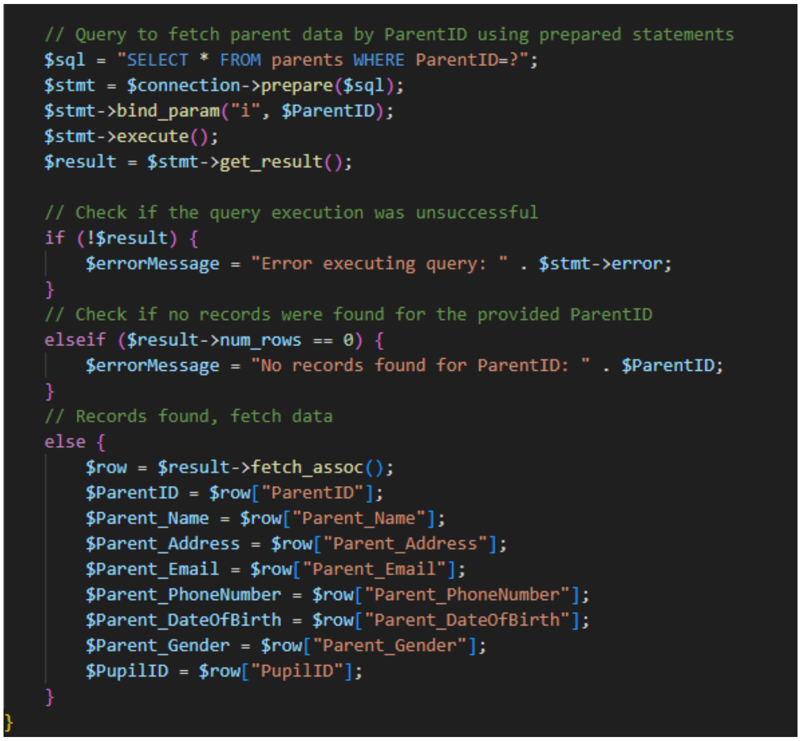
(BoostMyTool (2022). *PHP and MySQL with CRUD Operations: Create, Read, Update, Delete*. *YouTube*.)The add function consists of a session start because only admins can add, after empty variables will be populated later with the form. Then it will check if the form is submitted and if it is it will check a couple of conditions to check if the data is clean with a validation and it will also secure the submission, so SQL injection does not happen. After this, it will insert it into the database and refresh the variables. After the PHP code, there is also a form code that I will show later and is where the data is entered.

Edit record:



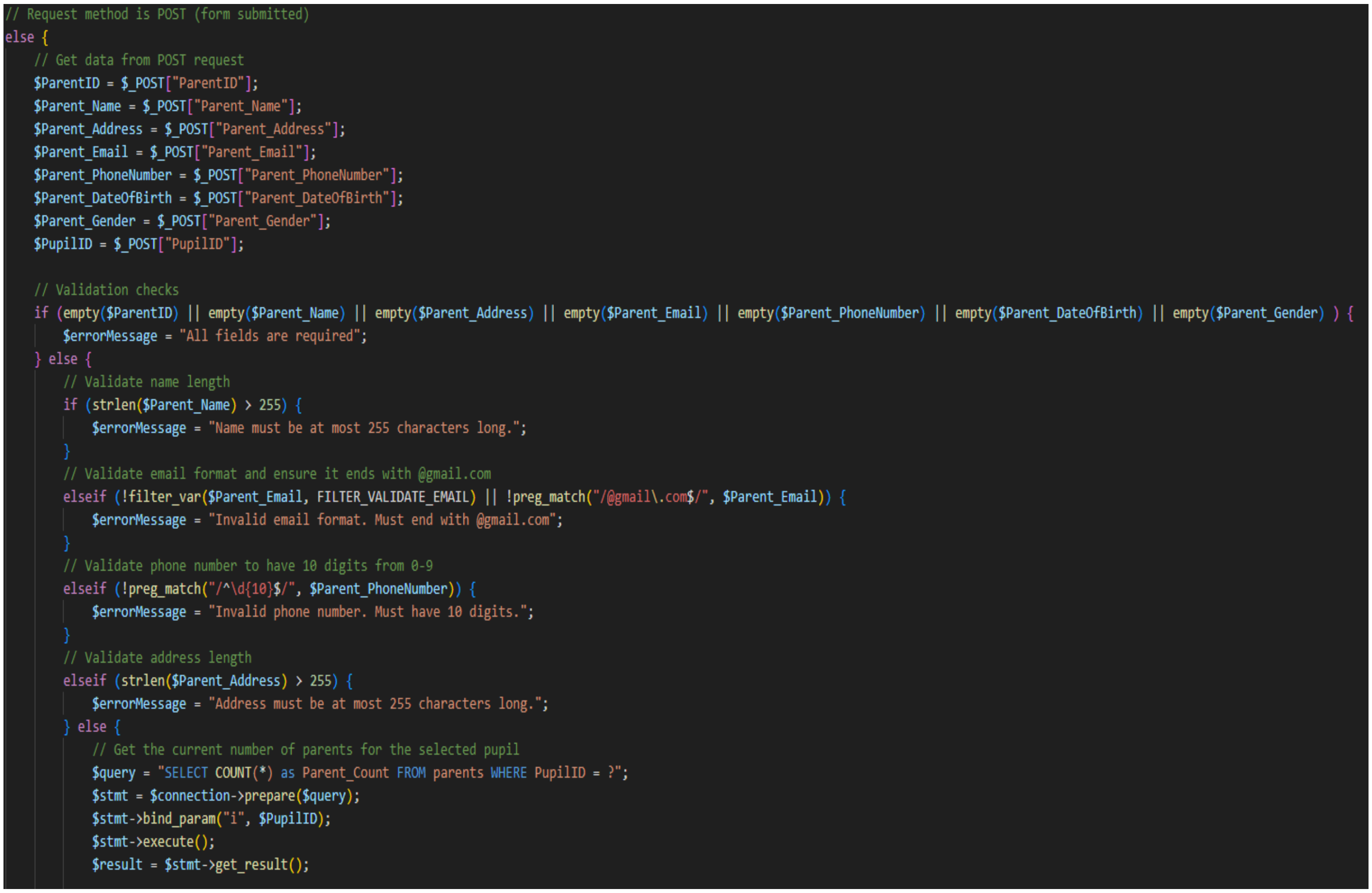
```
// Initialize variables for form fields and messages
$ParentID = "";
$Parent_Name = "";
$Parent_Address = "";
$Parent_Email = "";
$Parent_PhoneNumber = "";
$Parent_DateOfBirth = "";
$Parent_Gender = "";
$PupilID = "";
$errorMessage = "";
$successMessage = "";

// Query to fetch pupil data (commented out)
$query = "SELECT PupilID, Pupil_Name FROM pupil";
$PupilResult = $connection->query($query);

// Check if the request method is GET
if ($_SERVER["REQUEST_METHOD"] == "GET") {
    // Check if ParentID is set in the GET request
    if (!isset($_GET["ParentID"])) {
        // Redirect to the Parents.php page if ParentID is not set
        header("location: /test1/Parents.php");
        exit;
    }

    // Get ParentID from GET request
    $ParentID = $_GET["ParentID"];
```

Figure 43



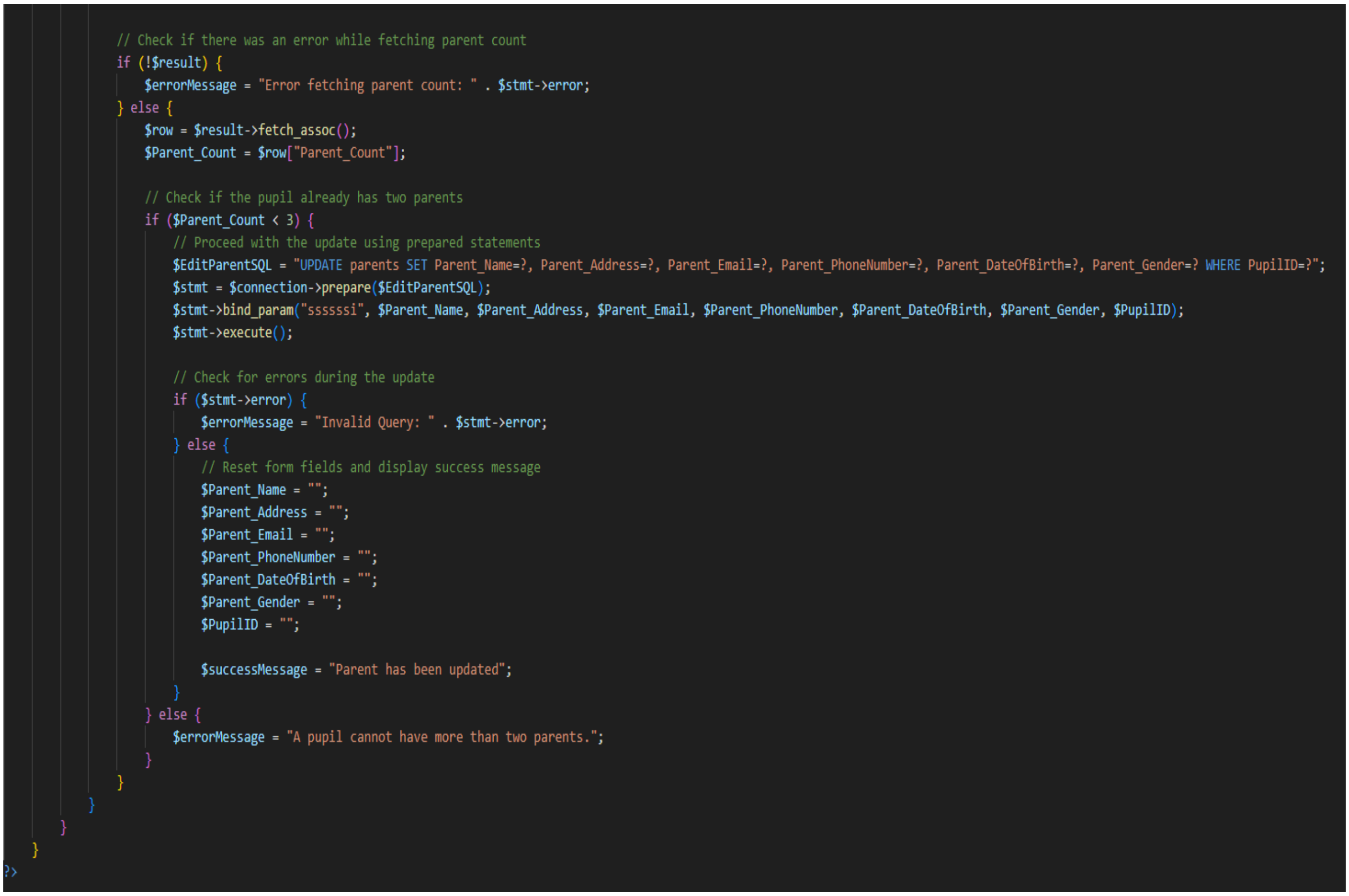
```
    // Query to fetch parent data by ParentID using prepared statements
    $sql = "SELECT * FROM parents WHERE ParentID=?";
    $stmt = $connection->prepare($sql);
    $stmt->bind_param("i", $ParentID);
    $stmt->execute();
    $result = $stmt->get_result();

    // Check if the query execution was unsuccessful
    if (!$result) {
        $errorMessage = "Error executing query: " . $stmt->error;
    }
    // Check if no records were found for the provided ParentID
    elseif ($result->num_rows == 0) {
        $errorMessage = "No records found for ParentID: " . $ParentID;
    }
    // Records found, fetch data
    else {
        $row = $result->fetch_assoc();
        $ParentID = $row["ParentID"];
        $Parent_Name = $row["Parent_Name"];
        $Parent_Address = $row["Parent_Address"];
        $Parent_Email = $row["Parent_Email"];
        $Parent_PhoneNumber = $row["Parent_PhoneNumber"];
        $Parent_DateOfBirth = $row["Parent_DateOfBirth"];
        $Parent_Gender = $row["Parent_Gender"];
        $PupilID = $row["PupilID"];
    }
}
```

Figure 44



```
// Request method is POST (form submitted)
else {
    // Get data from POST request
    $ParentID = $_POST["ParentID"];
    $Parent_Name = $_POST["Parent_Name"];
    $Parent_Address = $_POST["Parent_Address"];
    $Parent_Email = $_POST["Parent_Email"];
    $Parent_PhoneNumber = $_POST["Parent_PhoneNumber"];
    $Parent_DateOfBirth = $_POST["Parent_DateOfBirth"];
    $Parent_Gender = $_POST["Parent_Gender"];
    $PupilID = $_POST["PupilID"];

    // Validation checks
    if (empty($ParentID) || empty($Parent_Name) || empty($Parent_Address) || empty($Parent_Email) || empty($Parent_PhoneNumber) || empty($Parent_DateOfBirth) || empty($Parent_Gender) ) {
        $errorMessage = "All fields are required";
    } else {
        // Validate name length
        if (strlen($Parent_Name) > 255) {
            $errorMessage = "Name must be at most 255 characters long.";
        }
        // Validate email format and ensure it ends with @gmail.com
        elseif (!filter_var($Parent_Email, FILTER_VALIDATE_EMAIL) || !preg_match("/@gmail\.com$/", $Parent_Email)) {
            $errorMessage = "Invalid email format. Must end with @gmail.com";
        }
        // Validate phone number to have 10 digits from 0-9
        elseif (!preg_match("/^\d{10}$/", $Parent_PhoneNumber)) {
            $errorMessage = "Invalid phone number. Must have 10 digits.";
        }
        // Validate address length
        elseif (strlen($Parent_Address) > 255) {
            $errorMessage = "Address must be at most 255 characters long.";
        } else {
            // Get the current number of parents for the selected pupil
            $query = "SELECT COUNT(*) as Parent_Count FROM parents WHERE PupilID = ?";
            $stmt = $connection->prepare($query);
            $stmt->bind_param("i", $PupilID);
            $stmt->execute();
            $result = $stmt->get_result();
```

Figure 45

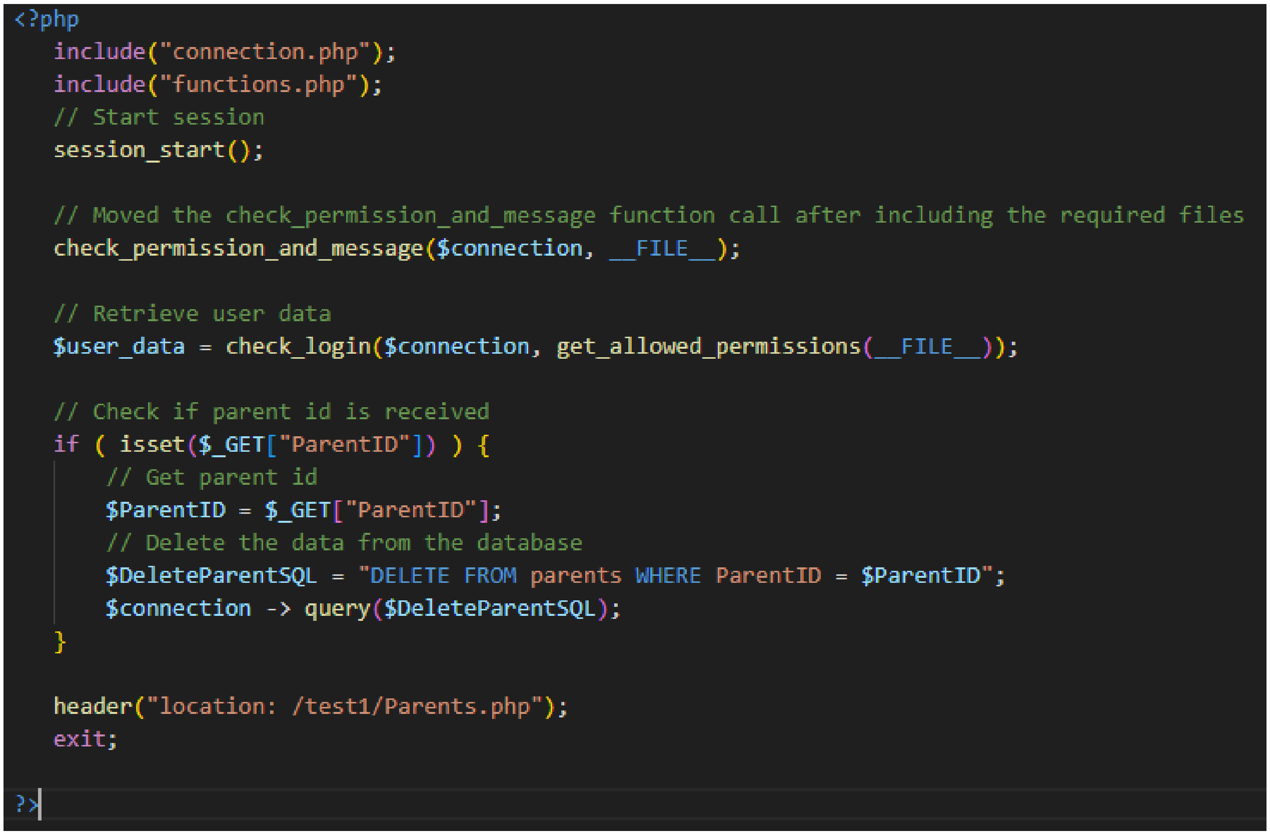


```
                // Check if there was an error while fetching parent count
                if (!$result) {
                    $errorMessage = "Error fetching parent count: " . $stmt->error;
                } else {
                    $row = $result->fetch_assoc();
                    $Parent_Count = $row["Parent_Count"];

                    // Check if the pupil already has two parents
                    if ($Parent_Count < 3) {
                        // Proceed with the update using prepared statements
                        $EditParentSQL = "UPDATE parents SET Parent_Name=?, Parent_Address=?, Parent_Email=?, Parent_PhoneNumber=?, Parent_DateOfBirth=?, Parent_Gender=? WHERE PupilID=?";
                        $stmt = $connection->prepare($EditParentSQL);
                        $stmt->bind_param("ssssssi", $Parent_Name, $Parent_Address, $Parent_Email, $Parent_PhoneNumber, $Parent_DateOfBirth, $Parent_Gender, $PupilID);
                        $stmt->execute();

                        // Check for errors during the update
                        if ($stmt->error) {
                            $errorMessage = "Invalid Query: " . $stmt->error;
                        } else {
                            // Reset form fields and display success message
                            $Parent_Name = "";
                            $Parent_Address = "";
                            $Parent_Email = "";
                            $Parent_PhoneNumber = "";
                            $Parent_DateOfBirth = "";
                            $Parent_Gender = "";
                            $PupilID = "";

                            $successMessage = "Parent has been updated";
                        }
                    } else {
                        $errorMessage = "A pupil cannot have more than two parents.";
                    }
                }
            }
        }
    }
?>
```

Figure 46

(BoostMyTool (2022). *PHP and MySQL with CRUD Operations: Create, Read, Update, Delete*. *YouTube*.)Editing the record is like the add function but this time it will fetch the information stored in the database and display it to the end user. And another difference is that instead of inserting the data, it will update it into the database.

Delete record:

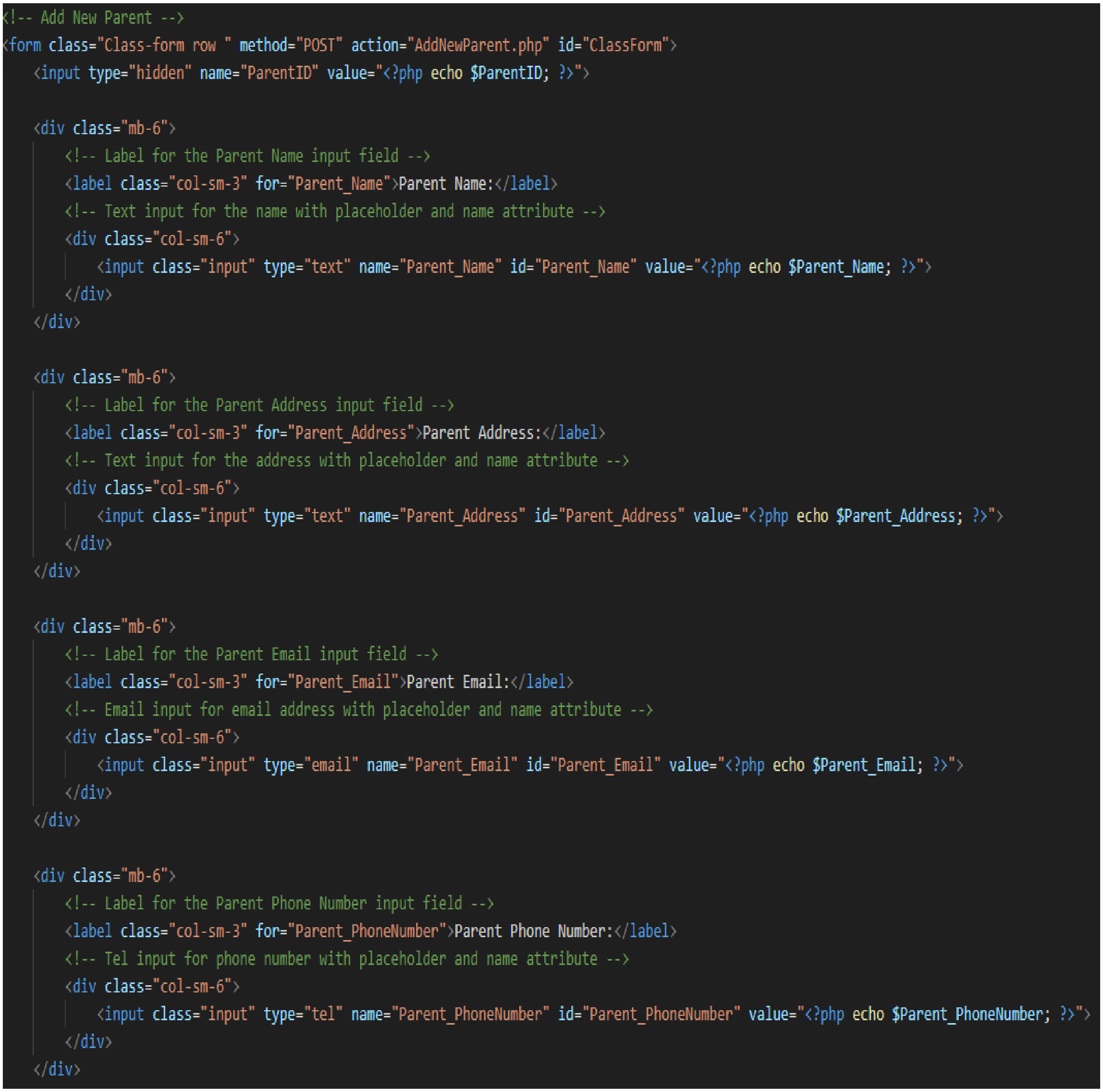


```php
<?php
    include("connection.php");
    include("functions.php");
    // Start session
    session_start();

    // Moved the check_permission_and_message function call after including the required files
    check_permission_and_message($connection, __FILE__);

    // Retrieve user data
    $user_data = check_login($connection, get_allowed_permissions(__FILE__));

    // Check if parent id is received
    if ( isset($_GET["ParentID"]) ) {
        // Get parent id
        $ParentID = $_GET["ParentID"];
        // Delete the data from the database
        $DeleteParentSQL = "DELETE FROM parents WHERE ParentID = $ParentID";
        $connection -> query($DeleteParentSQL);
    }

    header("location: /test1/Parents.php");
    exit;

?>
```

Figure 47

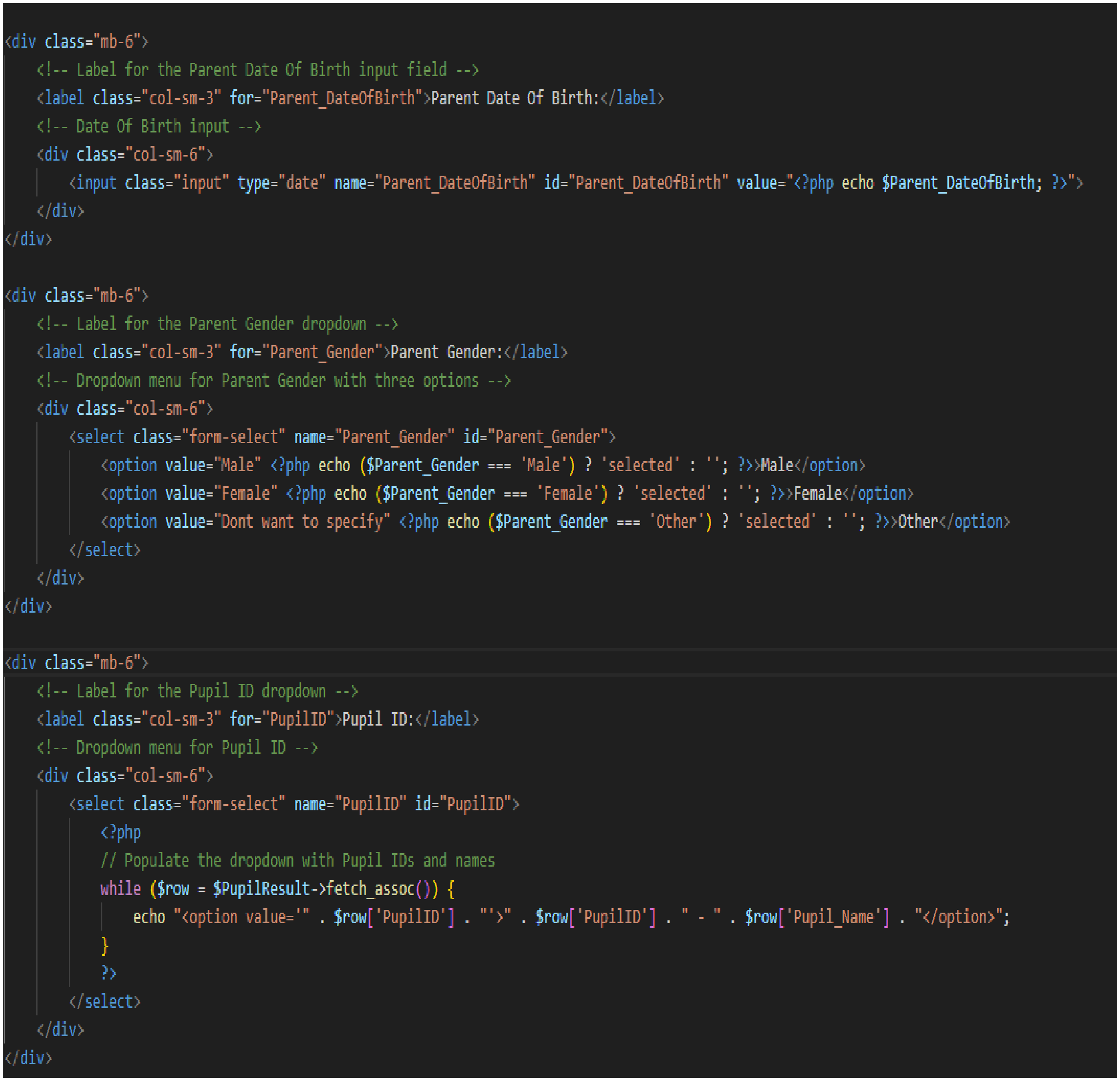
(BoostMyTool (2022). *PHP and MySQL with CRUD Operations: Create, Read, Update, Delete*. *YouTube*.)Deleting the record consists of starting the session to check that only allowed users are in and it will use the Delete command of SQL to delete the specific record that the user picked. Deleting entries is not recommended because the data might also impact other tables that have foreign keys, so companies store that data in another table, but for this assignment it was used to prove that it can be done.

Form:



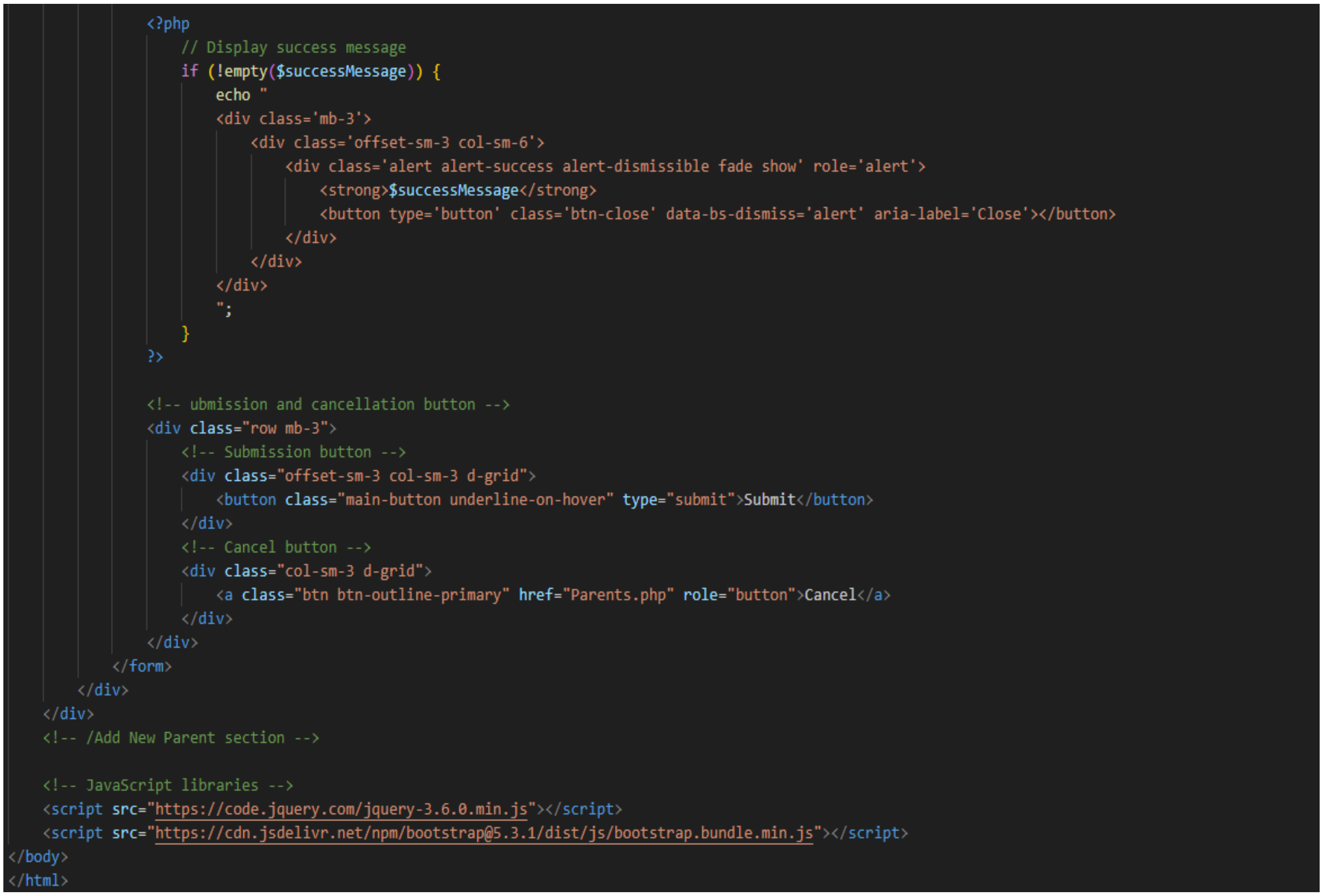
```
<!-- Add New Parent -->
<form class="Class-form row " method="POST" action="AddNewParent.php" id="ClassForm">
    <input type="hidden" name="ParentID" value="<?php echo $ParentID; ?>">

    <div class="mb-6">
        <!-- Label for the Parent Name input field -->
        <label class="col-sm-3" for="Parent_Name">Parent Name:</label>
        <!-- Text input for the name with placeholder and name attribute -->
        <div class="col-sm-6">
            <input class="input" type="text" name="Parent_Name" id="Parent_Name" value="<?php echo $Parent_Name; ?>">
        </div>
    </div>

    <div class="mb-6">
        <!-- Label for the Parent Address input field -->
        <label class="col-sm-3" for="Parent_Address">Parent Address:</label>
        <!-- Text input for the address with placeholder and name attribute -->
        <div class="col-sm-6">
            <input class="input" type="text" name="Parent_Address" id="Parent_Address" value="<?php echo $Parent_Address; ?>">
        </div>
    </div>

    <div class="mb-6">
        <!-- Label for the Parent Email input field -->
        <label class="col-sm-3" for="Parent_Email">Parent Email:</label>
        <!-- Email input for email address with placeholder and name attribute -->
        <div class="col-sm-6">
            <input class="input" type="email" name="Parent_Email" id="Parent_Email" value="<?php echo $Parent_Email; ?>">
        </div>
    </div>

    <div class="mb-6">
        <!-- Label for the Parent Phone Number input field -->
        <label class="col-sm-3" for="Parent_PhoneNumber">Parent Phone Number:</label>
        <!-- Tel input for phone number with placeholder and name attribute -->
        <div class="col-sm-6">
            <input class="input" type="tel" name="Parent_PhoneNumber" id="Parent_PhoneNumber" value="<?php echo $Parent_PhoneNumber; ?>">
        </div>
    </div>
```

Figure 48



```
<div class="mb-6">
    <!-- Label for the Parent Date Of Birth input field -->
    <label class="col-sm-3" for="Parent_DateOfBirth">Parent Date Of Birth:</label>
    <!-- Date Of Birth input -->
    <div class="col-sm-6">
        <input class="input" type="date" name="Parent_DateOfBirth" id="Parent_DateOfBirth" value="<?php echo $Parent_DateOfBirth; ?>">
    </div>
</div>

<div class="mb-6">
    <!-- Label for the Parent Gender dropdown -->
    <label class="col-sm-3" for="Parent_Gender">Parent Gender:</label>
    <!-- Dropdown menu for Parent Gender with three options -->
    <div class="col-sm-6">
        <select class="form-select" name="Parent_Gender" id="Parent_Gender">
            <option value="Male" <?php echo ($Parent_Gender === 'Male') ? 'selected' : ''; ?>>Male</option>
            <option value="Female" <?php echo ($Parent_Gender === 'Female') ? 'selected' : ''; ?>>Female</option>
            <option value="Dont want to specify" <?php echo ($Parent_Gender === 'Other') ? 'selected' : ''; ?>>Other</option>
        </select>
    </div>
</div>

<div class="mb-6">
    <!-- Label for the Pupil ID dropdown -->
    <label class="col-sm-3" for="PupilID">Pupil ID:</label>
    <!-- Dropdown menu for Pupil ID -->
    <div class="col-sm-6">
        <select class="form-select" name="PupilID" id="PupilID">
            <?php
            // Populate the dropdown with Pupil IDs and names
            while ($row = $PupilResult->fetch_assoc()) {
                echo "<option value='" . $row['PupilID'] . "'>" . $row['PupilID'] . " - " . $row['Pupil_Name'] . "</option>";
            }
            ?>
        </select>
    </div>
</div>
```

Figure 49



```
                <?php
                    // Display success message
                    if (!empty($successMessage)) {
                        echo "
                        <div class='mb-3'>
                            <div class='offset-sm-3 col-sm-6'>
                                <div class='alert alert-success alert-dismissible fade show' role='alert'>
                                    <strong>$successMessage</strong>
                                    <button type='button' class='btn-close' data-bs-dismiss='alert' aria-label='Close'></button>
                                </div>
                            </div>
                        </div>
                        ";
                    }
                ?>

                <!-- ubmission and cancellation button -->
                <div class="row mb-3">
                    <!-- Submission button -->
                    <div class="offset-sm-3 col-sm-3 d-grid">
                        <button class="main-button underline-on-hover" type="submit">Submit</button>
                    </div>
                    <!-- Cancel button -->
                    <div class="col-sm-3 d-grid">
                        <a class="btn btn-outline-primary" href="Parents.php" role="button">Cancel</a>
                    </div>
                </div>
            </form>
        </div>
    </div>
    <!-- /Add New Parent section -->

    <!-- JavaScript libraries -->
    <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.1/dist/js/bootstrap.bundle.min.js"></script>
</body>
</html>
```

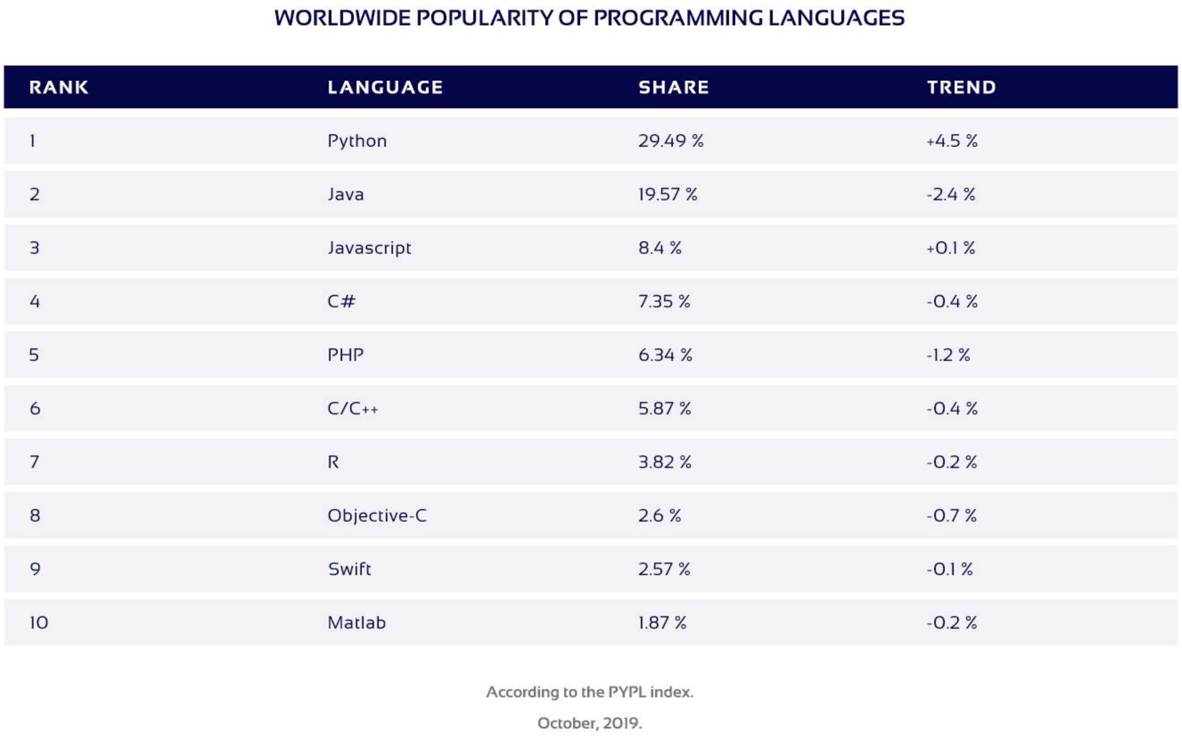
Figure 50

(BoostMyTool (2022). *PHP and MySQL with CRUD Operations: Create, Read, Update, Delete*. *YouTube*.)The form consists of getting the id and inputting the data into the appropriate box and submitting it. If the data is wrong it will send an error message, if the data is correct, it will display the success message.

Reflecting on the technologies used:

MySQL is a widely used technology to use CRUD operations as it is quite simple yet complete software that has many advantages. One of the biggest advantages is that it is free software, so it is widely available to everyone. This is helpful for students because they get a lot of experience about databases, and it is remotely accessible, allowing it to be dynamic and an accessible learning environment. Another advantage is that it is simple software that is easy to learn, which makes it user-friendly. It is also one of the software's most used in databases "MySQL claims that their software products now represent over 20% of all RDBMS installations".

Another software used is PHP. PHP is a back-end language that is used alongside MySQL to interactively make a web server. PHP has an in-built framework that helps it make coding easier as it already has all the code made so there is no need to make new code every time and the code is reusable. It is also a language that has been documented a lot so there are a lot of resources out there on the internet. It is also the 5th most used language in the world as shown in Figure 51.



WORLDWIDE POPULARITY OF PROGRAMMING LANGUAGES

RANK	LANGUAGE	SHARE	TREND
1	Python	29.49 %	+4.5 %
2	Java	19.57 %	-2.4 %
3	Javascript	8.4 %	+0.1 %
4	C#	7.35 %	-0.4 %
5	PHP	6.34 %	-1.2 %
6	C/C++	5.87 %	-0.4 %
7	R	3.82 %	-0.2 %
8	Objective-C	2.6 %	-0.7 %
9	Swift	2.57 %	-0.1 %
10	Matlab	1.87 %	-0.2 %

According to the PYPL index.
October, 2019.

Figure 51

Both MySQL and PHP are open-source projects. This means that it does not need licenses. It also makes it easier to identify bugs or errors and it also helps when testing or debugging.

References:

Denton, J. and Peace, A. (2003). Selection and Use of MySQL in a Database Management Course. *Journal of Information Systems Education*, [online] 14(4). Available at: https://jise.org/volume14/n4/JISEv14n4p401.pdf [Accessed 14 Dec. 2023].

W3.org. (2013). *The W3C Markup Validation Service*. [online] Available at: https://validator.w3.org/#validate_by_upload.

W3Schools (2019). *PHP Tutorial*. [online] W3schools.com. Available at: https://www.w3schools.com/php/default.asp.

Roznovsky, A. (n.d.). *Why Use PHP in 2020? Main Advantages and Disadvantages*. [online] light-it.net. Available at: https://light-it.net/blog/why-use-php-main-advantages-and-disadvantages/.

Google Books. (2023). *Learning PHP & MySQL*. [online] Available at: https://books.google.co.uk/books?hl=en&lr=&id=IpuT0GxHvAkC&oi=fnd&pg=PR5&dq=advantages+of+using+php&ots=8ar_Ax79yE&sig=t11XK3S8-YbJzbR3nUvG8pQzjyE&redir_esc=y#v=onepage&q=advantages%20of%20using%20php&f=false [Accessed 14 Dec. 2023].

www.youtube.com. (n.d.). *Part (01)- Dropdown value insert, edit & update in PHP MySQL Database with Source Code*. [online] Available at: https://www.youtube.com/watch?v=2DCDzAMIFsU [Accessed 15 Dec. 2023].

www.youtube.com. (n.d.). *Signup and Login with PHP and MySQL*. [online] Available at: https://www.youtube.com/watch?v=5L9UhOnuos0.

BoostMyTool (2022). *PHP and MySQL with CRUD Operations: Create, Read, Update, Delete*. *YouTube*. Available at: https://www.youtube.com/watch?v=NqP0-UkIQS4.

Google Developers. (n.d.). *Charts*. [online] Available at: https://developers.google.com/chart.