|  |  |
| --- | --- |
| **Name** | Manish Shashikant Jadhav |
| **UID** | 2023301005 |
| **Subject** | Software Engineering |
| **Experiment No.** | 9 |
| **Aim** | Testing |
| **Implementation** | Unit testing was performed using the **PHPUnit** framework to validate the functionalities of adding, updating, and deleting room details in the `hostelmsphp` database. The `setUp()` method established a connection to the database and ensured the required `rooms` table was created with constraints like `UNIQUE` for `room\_no`. It also inserted default test data to simulate real-world scenarios. Test cases were implemented to cover various outcomes, such as successful operations, handling invalid data, and managing edge cases like duplicate entries or non-existent records. Assertions were used to verify expected outcomes, such as ensuring data consistency and detecting failures caused by invalid inputs or constraints. The `tearDown()` method closed the database connection, ensuring that each test ran in isolation. This systematic approach ensured that the database interactions performed as expected under different conditions. |

1. **Units to Test:**

**Unit 1: addUser Function**

This unit tests the functionality to add a user to the database.

**Unit 2: updateUser Function**

This unit tests the functionality to update a user's details in the database.

**Unit 3: deleteUser Function**

This unit tests the functionality to delete a user from the database.

1. **Exhaustive Test Case for Unit 1:**

|  |  |  |
| --- | --- | --- |
| **Test Case ID** | **Description** | **Expected Output** |
| TC1 | Add a valid room with correct data (valid seater, room\_no, and fees). | Room added successfully |
| TC2 | Add a room with a duplicate room\_no (i.e., already exists in the database). | Error message or failure: "Room number already exists." |
| TC3 | Add a room with invalid data (e.g., negative seater, zero fees, or zero room\_no). | Error message or failure: "Invalid room data." |
| TC4 | Add a room with missing required fields (e.g., missing seater, room\_no, or fees). | Error message or failure: "All fields are required." |

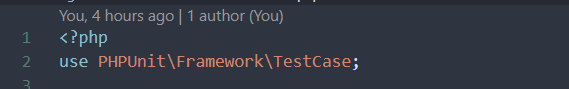
1. **Exhaustive Test Case for Unit 2:**

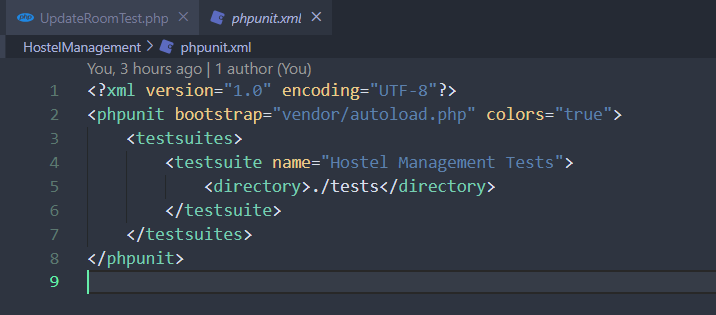
|  |  |  |
| --- | --- | --- |
| **Test Case ID** | **Description** | **Expected Output** |
| **TC1** | Update an existing room's data with valid information (e.g., new fees). | Room updated successfully |
| **TC2** | Attempt to update a non-existent room (invalid room\_no). | Error message or failure: "Room not found." |
| **TC3** | Attempt to update a room's data with invalid information (e.g., negative fees). | Error message or failure: "Invalid room data." |

1. **Exhaustive Test Case for Unit 3:**

|  |  |  |
| --- | --- | --- |
| **Test Case ID** | **Description** | **Expected Output** |
| **TC1** | Delete an existing room by valid room\_no. | Room deleted successfully. |
| **TC2** | Attempt to delete a non-existent room (invalid room\_no). | Error message or failure: "Room not found." |
| **TC3** | Attempt to delete a room with invalid data (e.g., non-integer room\_no). | Error or exception: "Invalid room number format." |

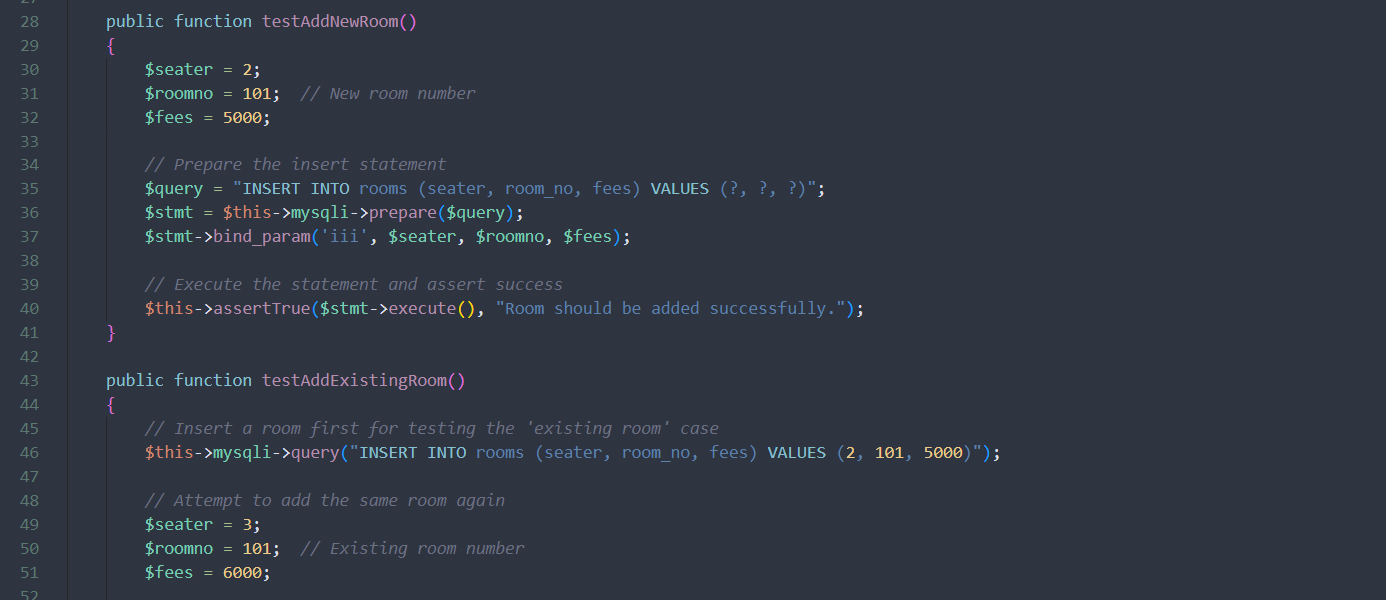
**Every File contains a header of PHPUnit:**

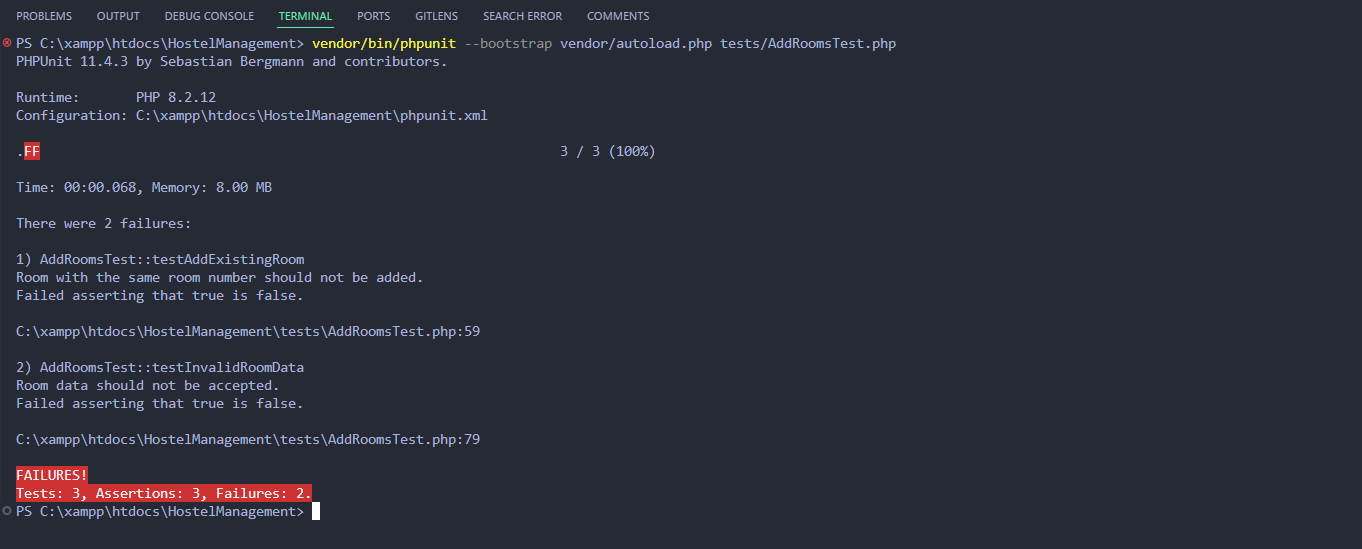
****

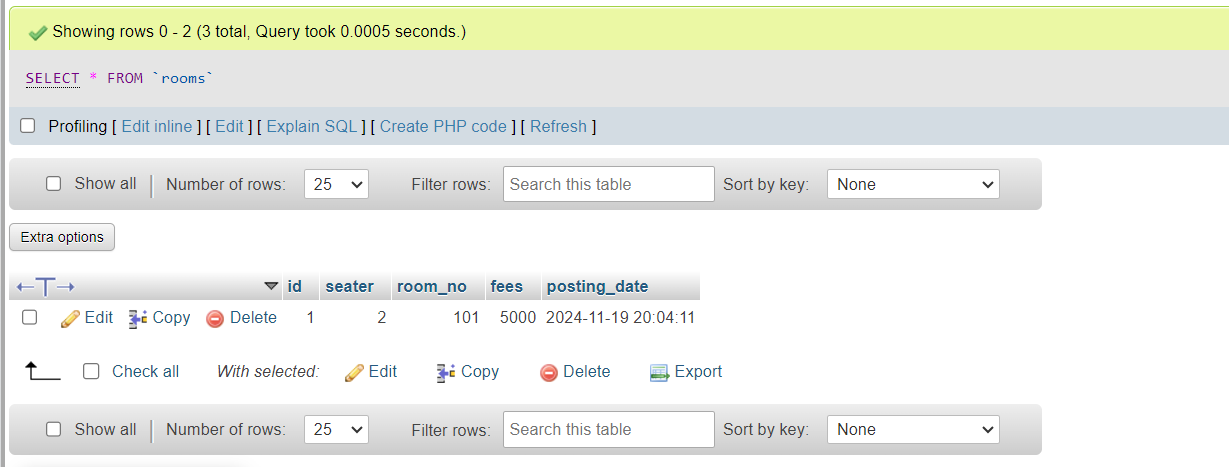




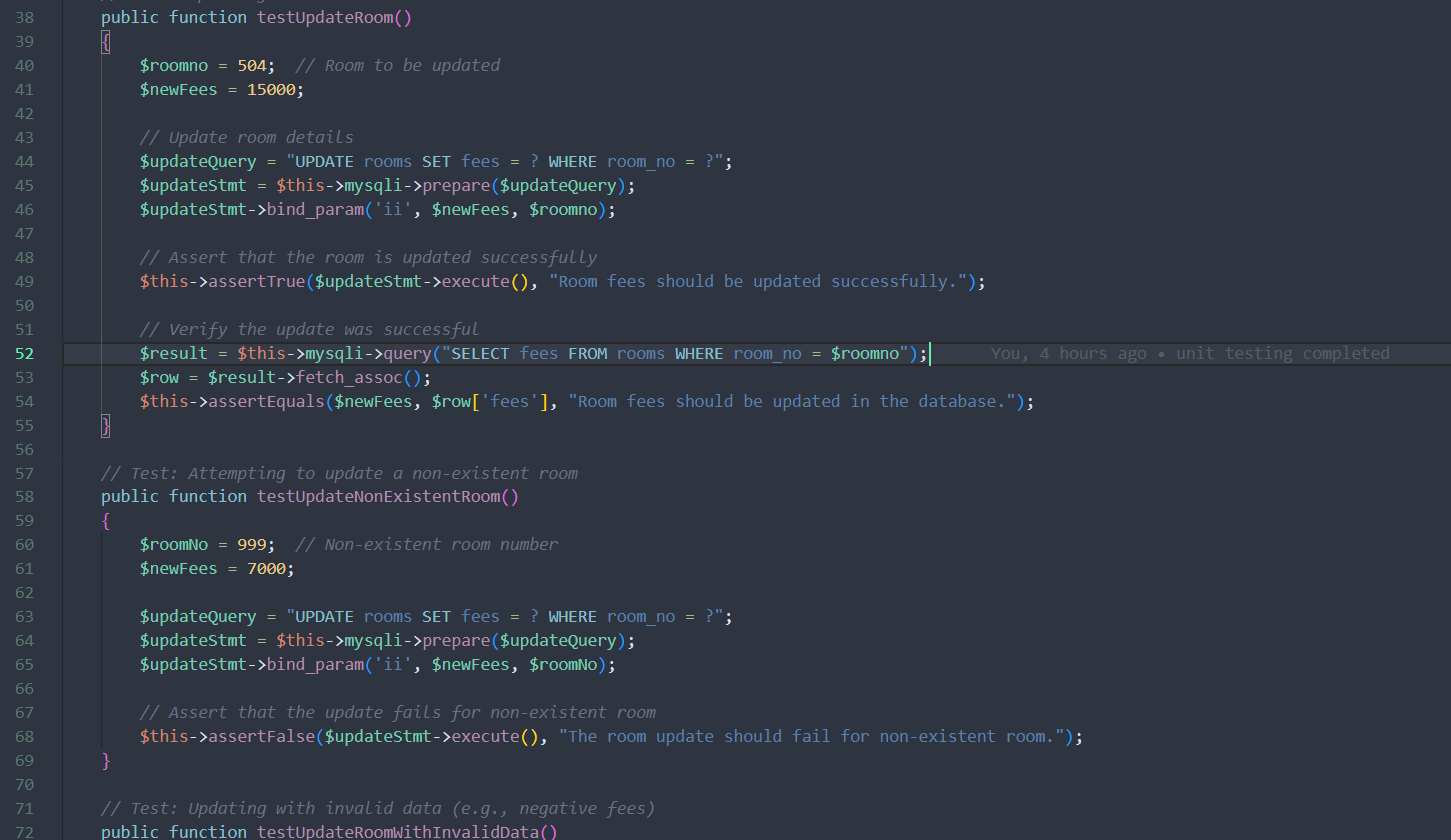
* **Unit Test for Add room:**

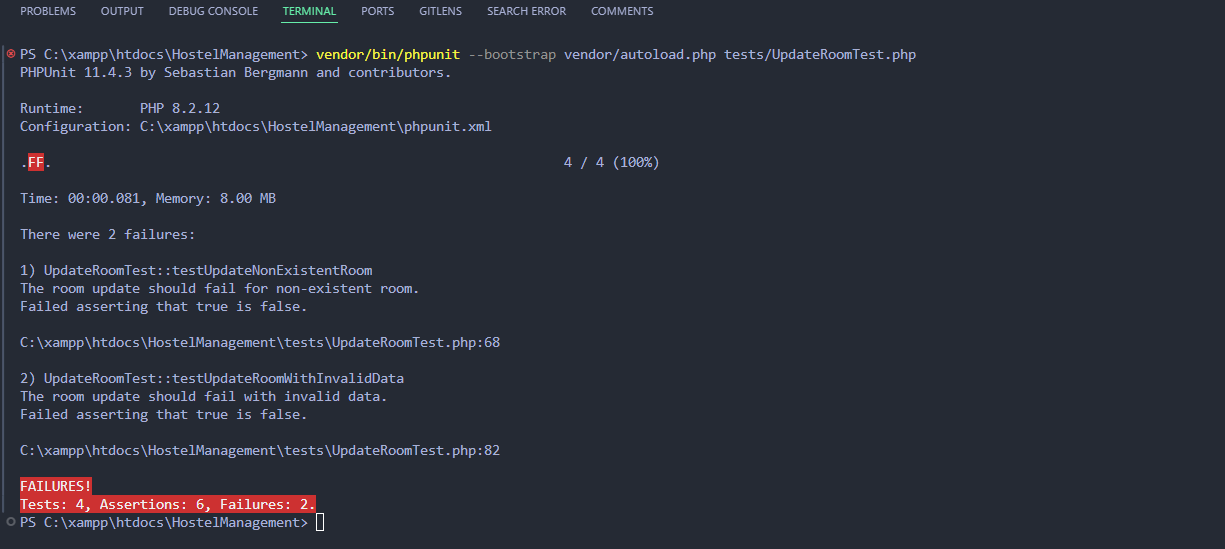


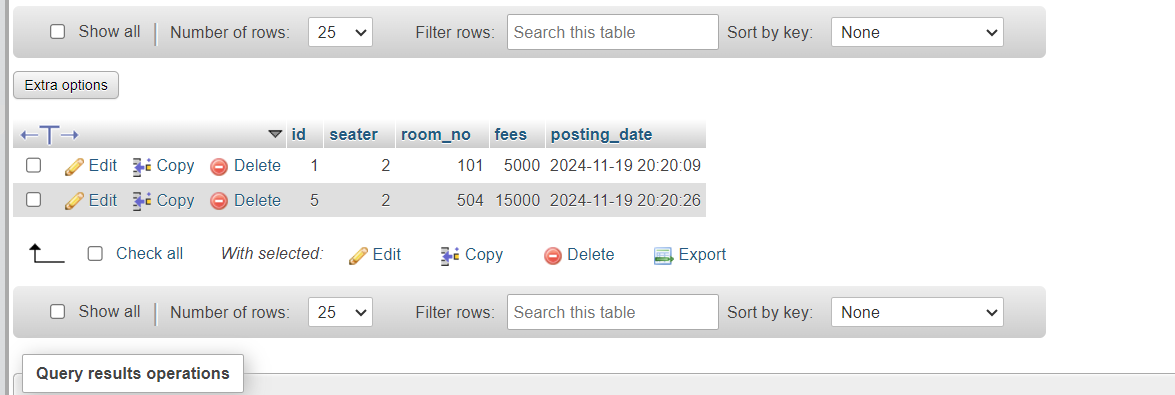




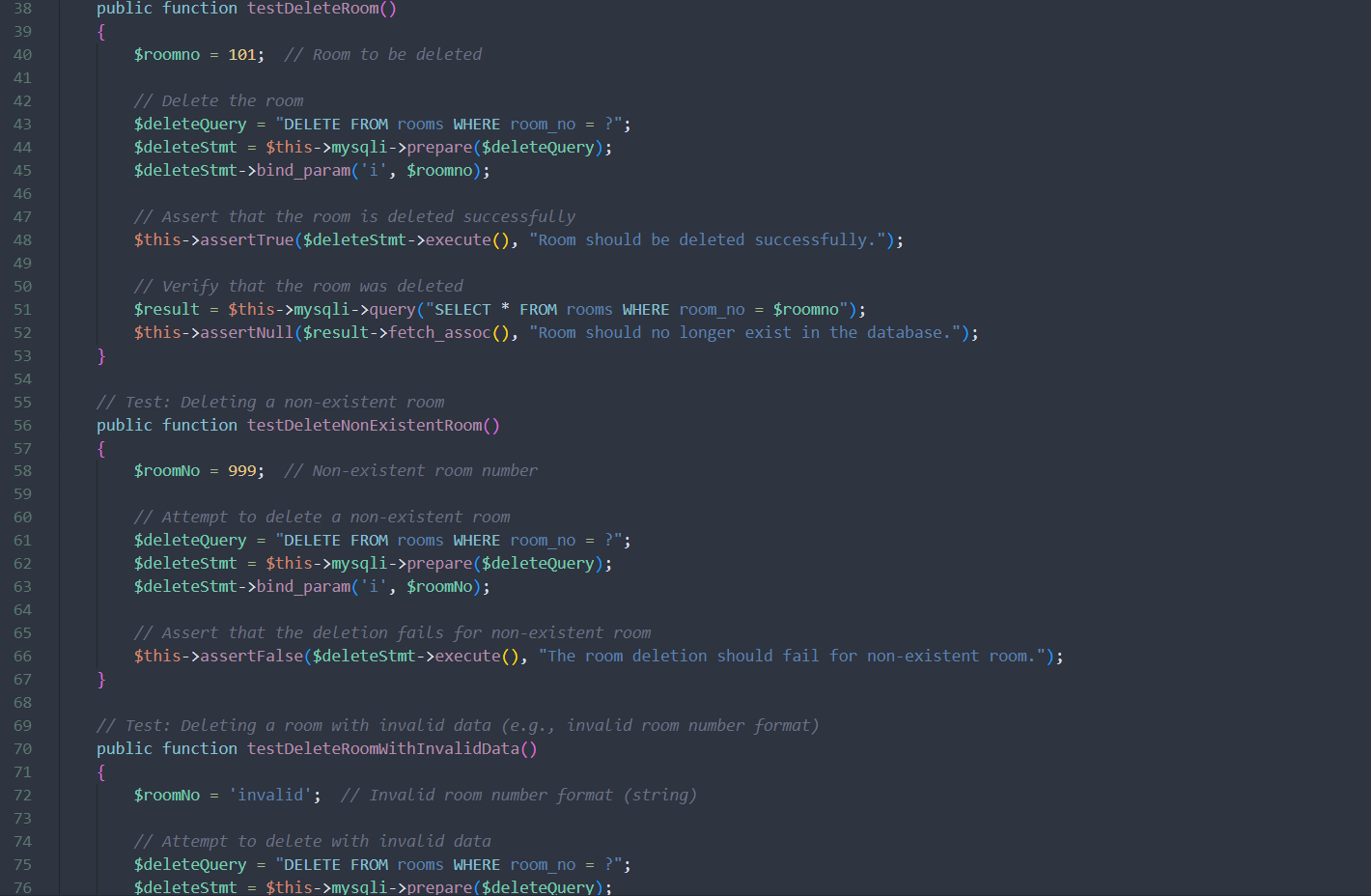
* **Unit Test for update room:**

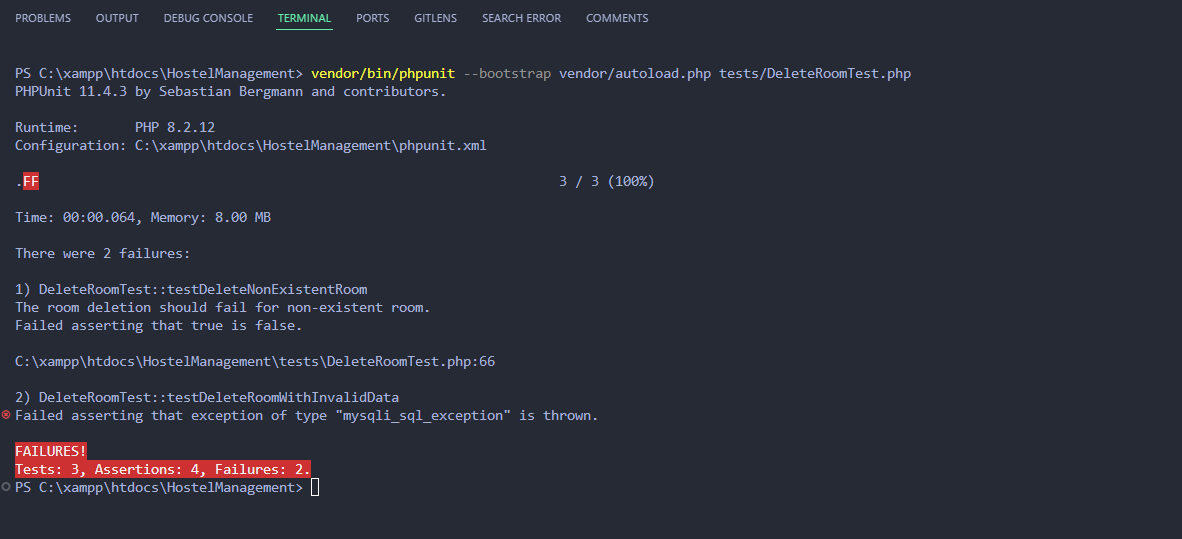


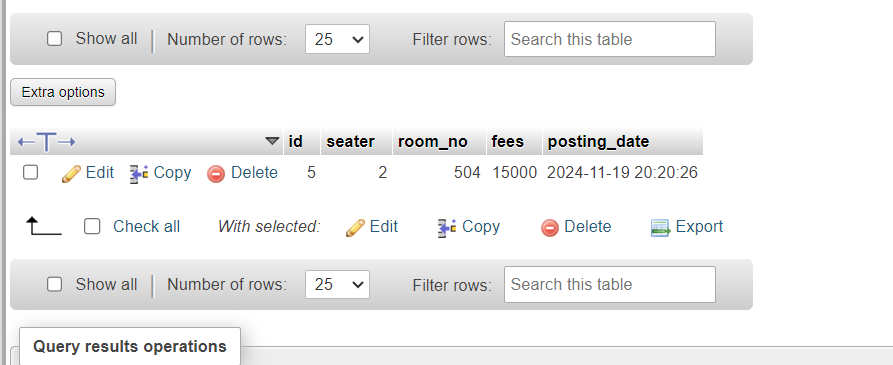




* **Unit Test for Delete Room:**







|  |  |
| --- | --- |
| **Conclusion** | Hence by completing this experiment I got to know how to perform unit testing on web application in PHP and MySQL using framework called PHPUnit. |