

There is database schema file name “event_booking_data_rexx_systems”

Below are the details for database connection.

```
// Database connection parameters
```

```
$host_name = "localhost";
```

```
$database_name = "event_booking_data_rexx_systems";
```

```
$user_name = "root";
```

```
$password = ""; //empty
```

Index.php

My PHP script begins by including class definitions and setting up database connection parameters. I initialize variables for data storage and filtering. An instance of the Database class is created for database operations. Data is retrieved from the database and stored in \$responseForLoopData.

If the user uploads a JSON file, the script processes and adds its data to the database.

If the user submits a search, the script filters data based on user input.

The HTML part of the script sets up a web page with a navigation bar for filtering and uploading. A table displays database records, and the total participation fee is shown.

In summary, this index.php script serves as a web application for interacting with a database and processing JSON files. Users can filter and display database records and upload JSON files to add new data. The displayed data is presented in an organized table format on the web page.

JsonFileReader.php

The JsonFileReader class in my code reads data from a JSON file. It starts with an empty array called \$data. When I create an instance of this class by providing a file path, it automatically reads the JSON file at that path. It checks if the file exists and can be read. If not, it shows an error message. If the file exists and can be read, it reads the content of the file. Then, it tries to convert the file's content into a PHP array. If it successfully converts the content, it stores that array in the \$data property. If any errors occur during this process, it catches and displays the error message.

we can later get the data from the JSON file using the getData() method. In short, this class helps in read data from a JSON file and handles possible errors along the way.

Database.php

This is my PHP class called "Database" that helps connect to a database and perform various actions with it. It connects to a database when create an instance of this class. It needs details like the host, database name, username, and password.

public function saveData(\$data) It can save data into different tables in the database. It checks if the data already exists and updates it if necessary. The data includes information about employees, events, and bookings.

public function getData() It can retrieve data from the database, such as employee names, emails, event names, participation IDs, participation fees, event dates, and versions. It also calculates the total participation fees.

public function filterData(\$filter, \$filterValue) It can filter data based on a specific filter, like searching by employee name or event name that match a given filter value.

public function getExistingEmployeeId(\$employeeMail) It has a function to check if an employee with a specific email already exists in the database to avoid duplication.

Overall, this class helps to interact with a database by providing methods to save, retrieve, filter, data.

Database schema

I created three tables in my Database schema

Employees Table

The Employees table is used to store information about employees. It has a primary key named `employee_id`, which uniquely identifies each employee in the table. There are also other information about employees, such as their names (`employee_name`) and email addresses (`employee_mail`), is stored in this table.

Database schema

The Events table is used to store information about various events. It has a primary key named `event_id`, which uniquely identifies each event in the table. The table also includes the name of the event (`event_name`).

Database schema

The Bookings table is used to track which employees participate in which events and store additional information related to these bookings. It has its own primary key named `participation_id`, which uniquely identifies each booking in the table.

It includes the `employee_id` column, which is a foreign key referencing the `employee_id` column in the Employees table. This relationship indicates that each booking is associated with a specific

employee. Similarly, it includes the event_id column, which is a foreign key referencing the event_id column in the Events table. This relationship indicates that each booking is associated with a specific event. The participation_fee column stores the cost or fee associated with the participation in the event. The event_date column stores the date and time of the event.

Overall, the Bookings table acts as a bridge between the Employees and Events tables. It allows you to record which employees are participating in which events, along with additional details like participation fees and event dates. The foreign keys in the Bookings table (i.e., employee_id and event_id) establish the relationships between employees, events, and their respective bookings, ensuring data consistency and integrity.

90 mins are very less for it.