**Server-Side Development   
Working Manual for the Final Project**

**Event Company System**

**T00258226 Onishchenko Vladislav**

Name of the database: EventCompanySYS.

The sql script for creating the database: eventcompany.sql

**Tables which included in EventCompanySYS:**

A screen shot of a computer

AI-generated content may be incorrect.

**Table «Bookings»:**

A screen shot of a computer

AI-generated content may be incorrect.

**Table «Clients»:**

A black screen with white text

AI-generated content may be incorrect.

**Table «Events»:**

A screenshot of a computer program

AI-generated content may be incorrect.

**Table «EventTypes»:**

A black screen with white text

AI-generated content may be incorrect.

**Table «Users»:**

A black screen with white text

AI-generated content may be incorrect.

**Table «Venues»:**

A screenshot of a computer program

AI-generated content may be incorrect.

**User Mode**

**Index.php:**

This page is designed for a ticket booking, where users can browse upcoming events, filter them based on certain criteria (like event name, date, or location), and book tickets. Once users are logged in, they can also view their booking history. Below is an explanation of each part of the program and how they interact with each other.

A screenshot of a computer

AI-generated content may be incorrect.

A screen shot of a computer

AI-generated content may be incorrect.

**Login:**Standard login form with hidden password.

A screenshot of a login form

AI-generated content may be incorrect.

A screenshot of a login screen

AI-generated content may be incorrect.

**MyBookings.php:**

To show bookings the program use this sql query:

*SELECT Events.name, Events.eventType, Events.description, Events.eventDate, Events.startTime, Events.duration, Events.filepath, Bookings.\**

*FROM Bookings*

*JOIN Events ON Bookings.eventid = Events.eventID*

*WHERE Bookings.email = :bEmail*

A screenshot of a computer

AI-generated content may be incorrect.

**BookTicket.php:**

On booking page user have to provide his email address and number of tickets for booking. If user already logged in the email address will be insert automatically. If user makes mistakes the alert system will notify what error the user made.

A screen shot of a computer screen

AI-generated content may be incorrect.

A screenshot of a computer screen

AI-generated content may be incorrect.

User can logout from account at any time

A blue and white sign

AI-generated content may be incorrect.

**Admin and Staff mode**

After logging under Admin account the administrator can access one of four management pages: Clients, Events, Venues, Bookings. All of them contain option: View all or Add new.

A screenshot of a computer

AI-generated content may be incorrect.

A blue and white rectangular object with black text

AI-generated content may be incorrect.

**Add Form:**

Simple Form to add new client, event, venue or booking.

For add element to database system use this sql query:

*INSERT INTO Clients (orgName, contactPerson, telephone, email, building, street, city, county, eircode, status)*

*VALUES (:cOrgName, :cContactPerson, :cTelephone, :cEmail, :cBuilding, :cStreet, :cCity, :cCountry, :cEircode, 'A')*

A screenshot of a web page

AI-generated content may be incorrect.

For more advanced insert like add booking to the file, system use two sql queries.

Insert booking to the database:

*INSERT INTO Bookings (email, eventid, numberOfTickets, price, status)*

*VALUES (:bEmail, :bEventID, :bNumberOfTickets, :bPrice, 'A')*

Update Available tickets in event file:

*Update Events*

*SET availableTickets = :bavaliableTickets*

*WHERE eventID = :bEventID*

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

Advanced add event form to choose records of client and venue from another tables. By clicking the selected row the client information store in the client input.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

**ListOf.php**

Table with all existing records in the system with update and delete buttons. On click of delete record the system will check is any table contain this information, if yes the status will be changed, if not the row will be deleted from the file. On click on update button the system will transfer you to the update form.

List of use this sql query to show all element in the table:  
*SELECT \* FROM Clients*

For search:

*SELECT \* FROM Clients WHERE orgName LIKE :cOrgName*

A screenshot of a computer

AI-generated content may be incorrect.

**Update.php**

Update form contain all information about selected entity. The working staff can change any information except the ID. The new entered information will be validate and added to the file.

For update system use update sql query:  
*UPDATE clients*

*SET orgName = :cOrgName, contactPerson = :cContactPerson, telephone = :cTelephone, email = :cEmail,*

*building = :cBuilding, street = :cStreet, city = :cCity, county = :cCounty, eircode = :cEircode,*

*status = :cStatus*

*WHERE clientID = :cClientID*

**A screenshot of a computer

AI-generated content may be incorrect.**

**A green and white sign with black text

AI-generated content may be incorrect.**