**ESKIŞEHIR TECHNICAL UNIVERSITY**

**FACULTY OF ENGINEERING**

**DEPARTMENT OF COMPUTER ENGINEERING**

**2022-23 SPRING SEMESTER**

**BIM216**

**DATABASE MANAGEMENT SYSTEMS**

**STADIUM SEATS BOOKING AND SALES DATABASE SYSTEM**

**Group Members(G.29):**

**Ayşenur YAYLACI -18850734860**

**Berru KAYA– 57964172236**

***Requirement Analysis and Classification of User Types***

We wanted to create a Stadium Seats Booking and Sale Database System. We first gathered the requriments we need to satisfy to get a properly working application. We considered our system’s purpose as being “a computerized system designed to manage the booking and sale of seats in a staidum”.

Our system needs to store information about each seat in the stadium. The seats can be avaliable, booked or purchased(not avaliable). Once the customer books the seat, he/she needs to decide if he wants to go with purchase or not. If he does, he does his payment. After paying, the user is given a ticket with the information written. if the person does not want to take the booked seat, in this case that seat will be added to the booked seats list and will not appear among the available seats.The system also needs to store information about the events that are going to be hold, the host stadiums, their date and times and their price.

Two kind of user type is present in our database. First is “admin”. Admin has a username and a password to access events. Admin can view the events and can do CRUD operations to them. Which means, admin can add an event, update an event or delete an event. Also, admin can view the general system information.

Second user type is “customer”. Customer is the user that our system service. Customer needs to enter his name,surname,age,city and country to view the available events. Customer can book or purchase an available seat for his desired event. Also, costumer is given a ticket.

***Functionalities(updated)***

1.The system stores the user's person id, name, surname, age, location which consists of city and country.

2.The system stores information about seats with different id in different class, located in each stadium that has stadium id, stadium name, stadium location(city and country)and capacity of a stadium.

3.The system keeps track of the sports events, which have event id and name, that will be played at the stadium and their dates(time, date) that will take place.

4.The system provides the user with booking and buying features.

5.The system eventually provides the user with a ticket containing user information, seat, stadium ,event information and price.

By the data in the system, following queries can be derived:

* Which age group is the most commonly seen between the customers?
* What is the total number of tickets that has been sold monthly?
* Does customer buy ticket for the events at their stored location or somewhere else?
* What is the most preferred stadium among the customers?
* What is the most preferred layer of seat to watch the events?
* For which events, all the tickets were sold out?

Overall, the stadium seat booking and sales database system is a valuable tool for customers. It streamlines the process of allocating and purchasing seats, provides valuable data for decision-making, and enhances the overall customer experience.

***E-R Diagram Explanation:***

**Entity Sets**

1.admin

2.Customer

3.Seats

4.Stadium

5.Event

6.Sports

7. Payment

8. Books

**Attributes & Primary Keys**

1.Customer: Person\_ID, c\_name,surname,age,City,Country

2.Seats: Seats\_ID,Seat\_Number,class

3.Stadium: Stadium\_ID, City, Country,Capacity,st\_Name

4.Event: Event\_ID,Event\_Name,e\_date,e\_time

5.Sports: S ports\_ID,Sports\_Name

6.admin:aid,a\_name,password

7.Payment:payment\_id,total\_price,

8.books:b\_id,price,occupied

**Relationships**

***handles***: admin-Events (one-to-many)

* Admin can handle many events. He can create, delete or update them.

***Purchase***: Payment-Seats-Event (many-to-many-to-many)

* There are many payment for many seats at many events.

***Ticket\_info***: Customer-Event-Seats-Payment(many-to-many-to-many-to-many)

* Many customer can do many payments for many seats at many events.

***booking***: Seats-Books-Event (many-to-many-to-many)

* There can be many seats in one layer.

***TakePlace***: Stadium-Event (one-to-many)

* A stadium can hold many events.

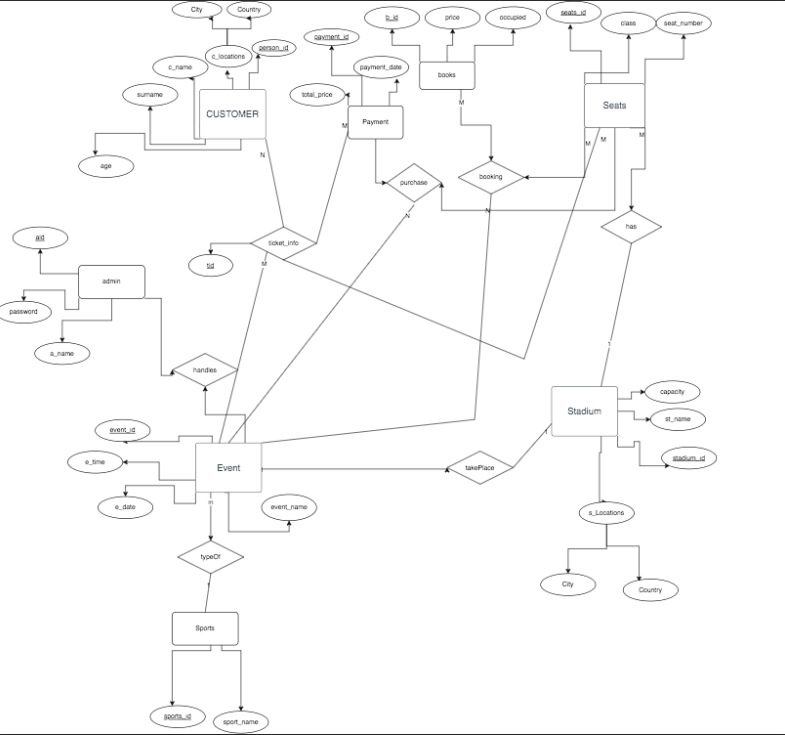
***TypeOf***: Event-Sports (one-to-many)

* An event can be a type of different sports.

***has***: Stadium-Seats (one-to-many)

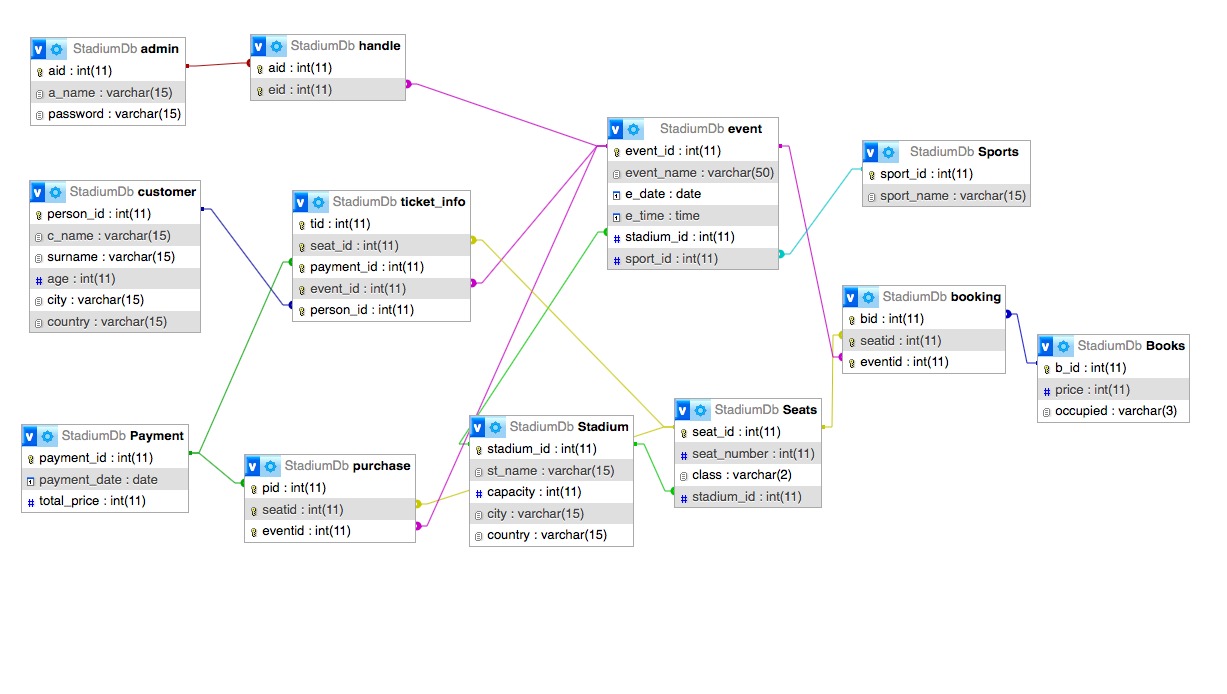
* A stadium has many seats.

***E-R DIAGRAM***



***E-R Diagram Digital Version*** https://drive.google.com/file/d/1HK2PKDtdcS4S7I7Cfiu8yFhVQ8P2JdW5/view?usp=sharing

***Relational Database Schema***



**Relational Tables of Our DB**

**Customer**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Person\_id | C\_name | surname | age | city | country | role |

**Admin**

|  |  |  |
| --- | --- | --- |
| aid | A\_name | password |

**Seats**

|  |  |  |  |
| --- | --- | --- | --- |
| Seat\_id | **Stadium\_id** | Seat\_number | class |

**Stadium**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Stadium\_id | St\_name | capacity | city | country |

**Event**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Event\_id | **Stadium\_id** | **Sports\_id** | Event\_name | E\_time | E\_date |

**Sports**

|  |  |
| --- | --- |
| Sports\_id | Sport\_name |

**Books**

|  |  |  |
| --- | --- | --- |
| B\_id | price | occupied |

**Payment**

|  |  |  |
| --- | --- | --- |
| Payment\_id | Payment\_date | Total\_price |

**Ticket\_Info**

Pk(tid, seat\_id, payment\_id,person\_id,event\_id)

Fk(seat\_id, payment\_id,person\_id,event\_id))

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| tid | **Seat\_id** | **Payment\_id** | **Person\_id** | **Event\_id** |

**Booking**

Pk(b\_id, seat\_id, event\_id)

Fk(b\_id, seat\_id, event\_id)

|  |  |  |
| --- | --- | --- |
| **B\_id** | **Seat\_id** | **Event\_id** |

**Purchase**

Pk(payment\_id, seat\_id, event\_id)

Fk(payment\_id, seat\_id, event\_id)

|  |  |  |
| --- | --- | --- |
| **Payment\_id** | **Seat\_id** | **Event\_id** |

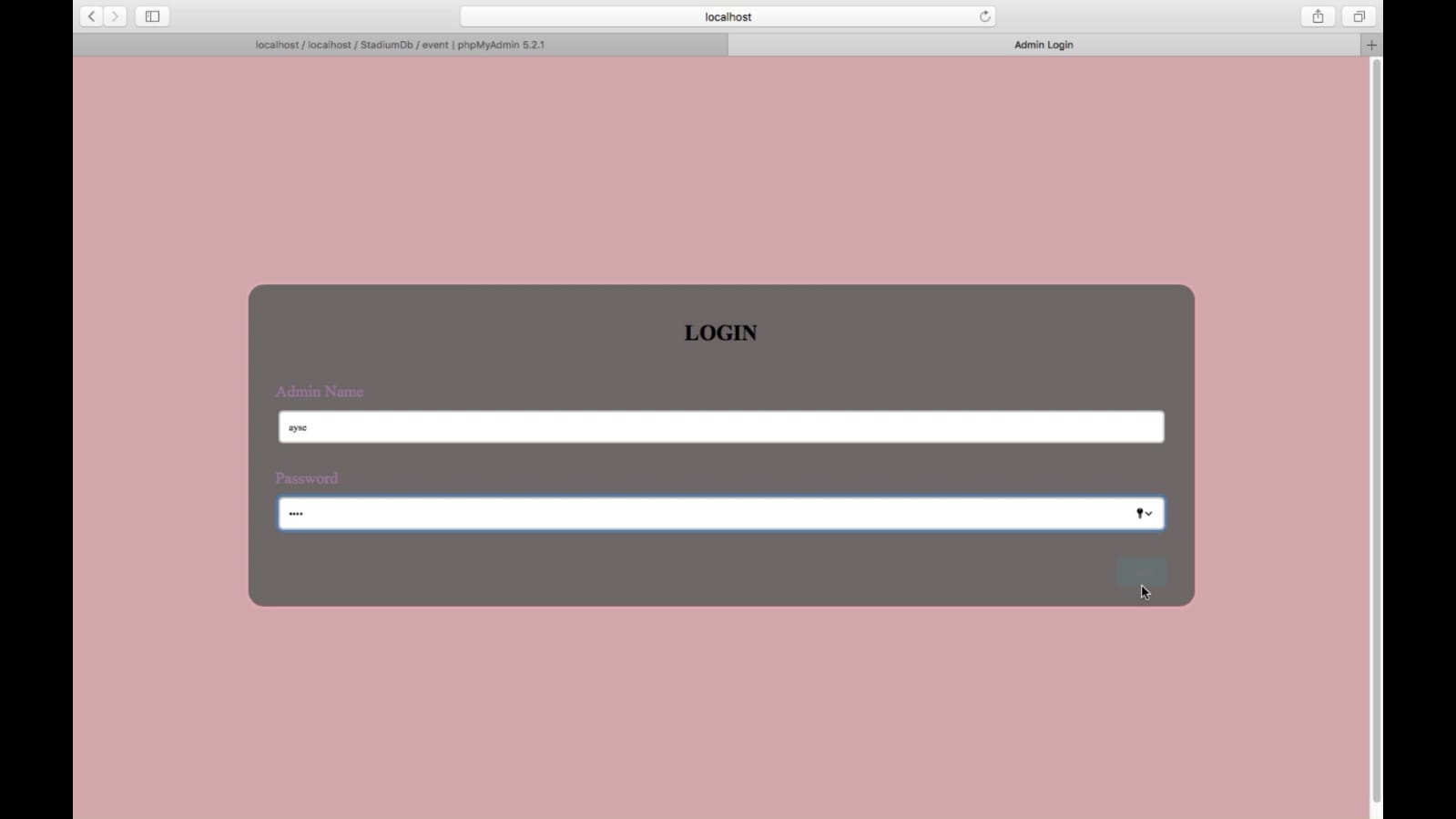
**Handles**

Pk(aid, event\_id)

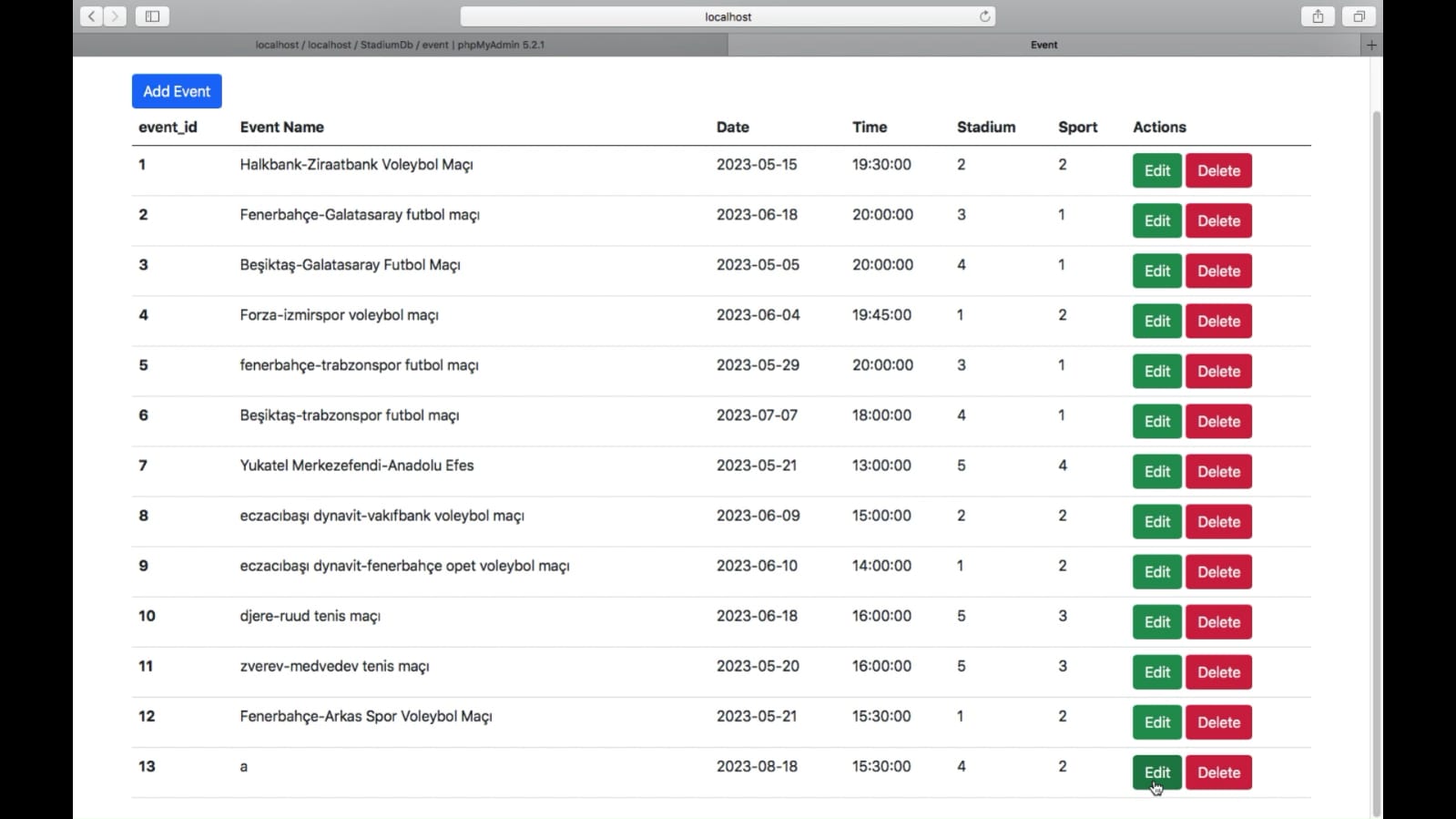
Fk(aid, event\_id)

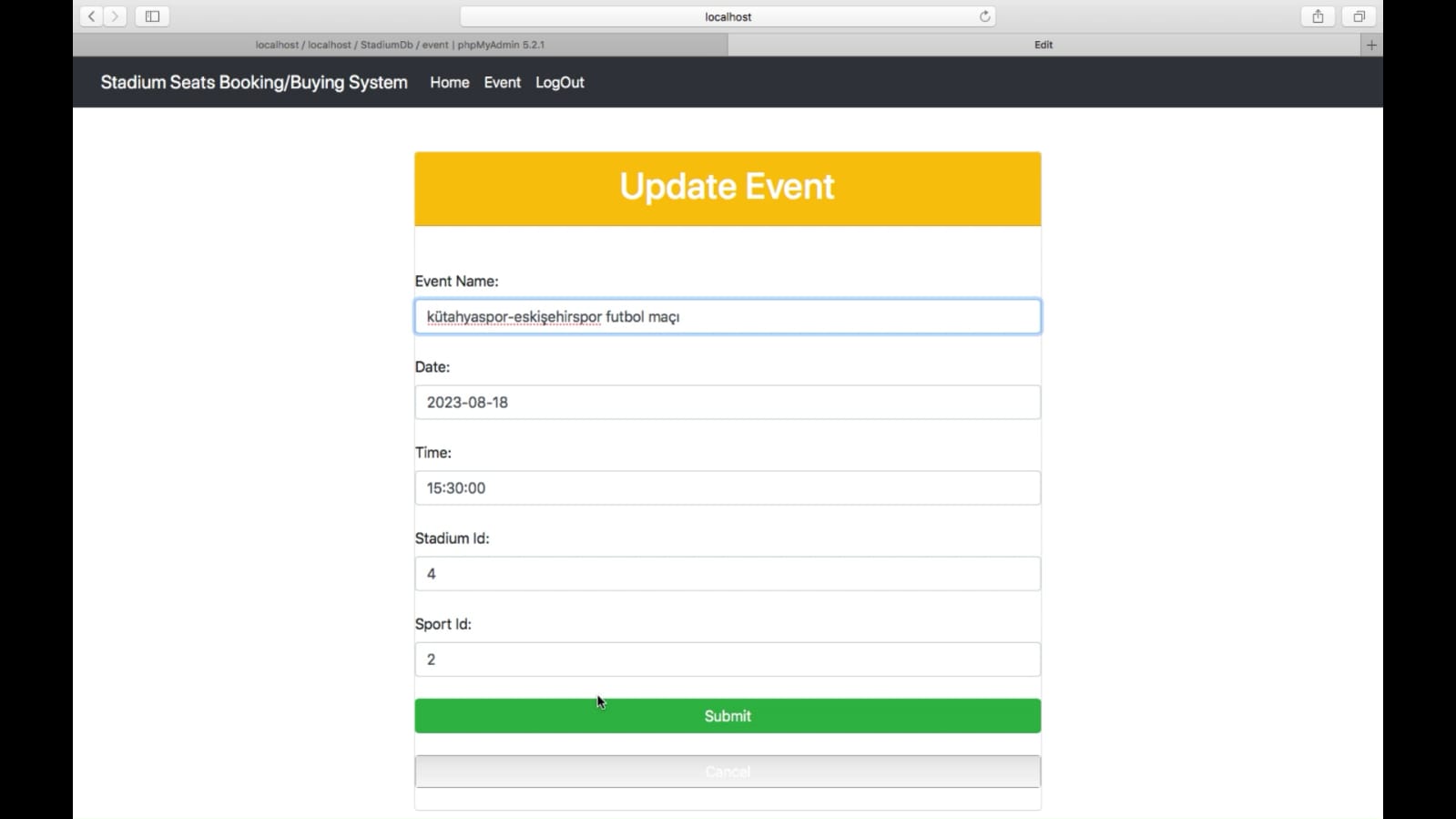
|  |  |
| --- | --- |
| **aid** | **Event\_id** |

**(admin logs in)**

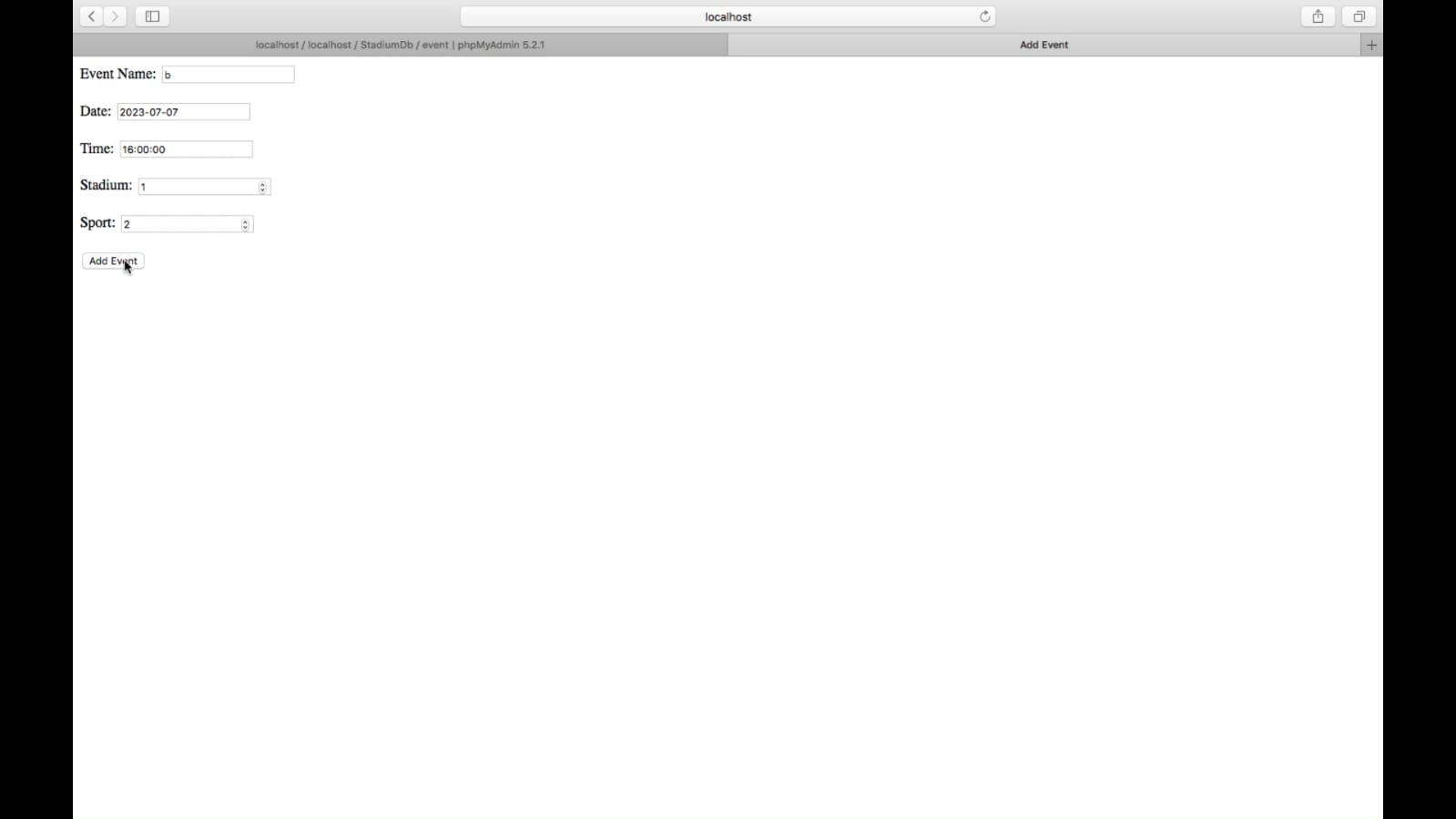


**(admin can edit events by clicking on “event”**

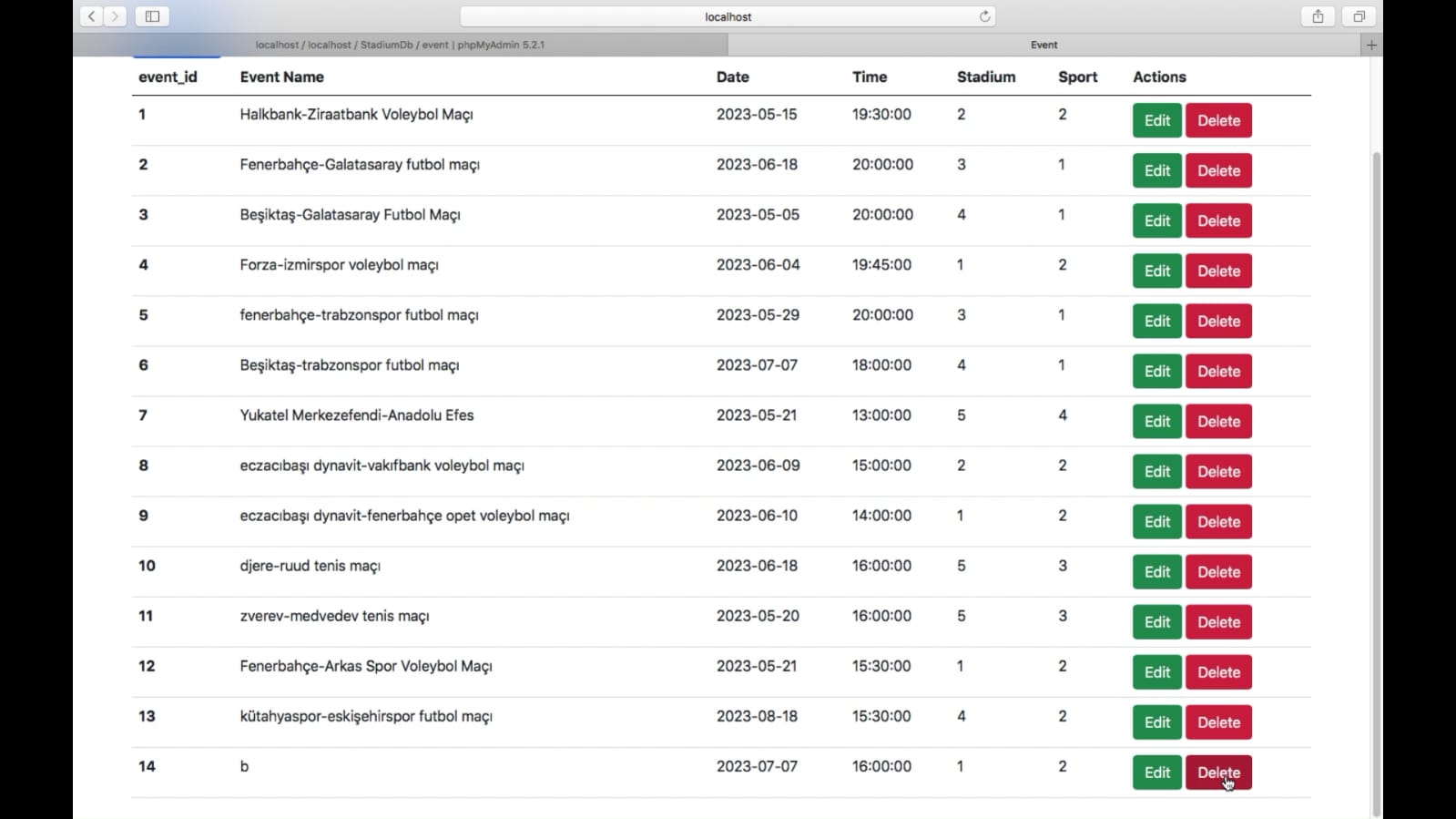


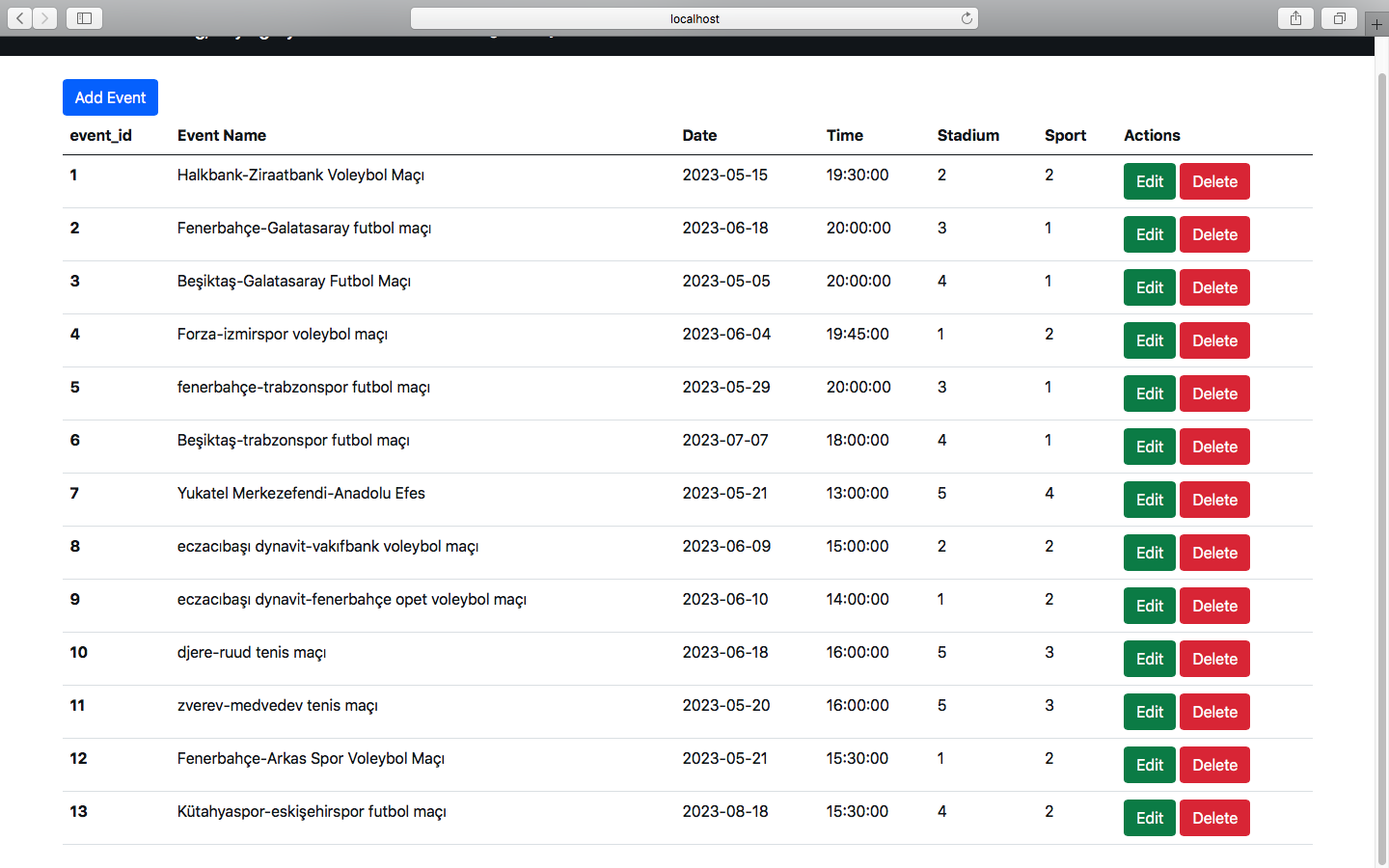


**(admin can add new event by clicking “add event”)**

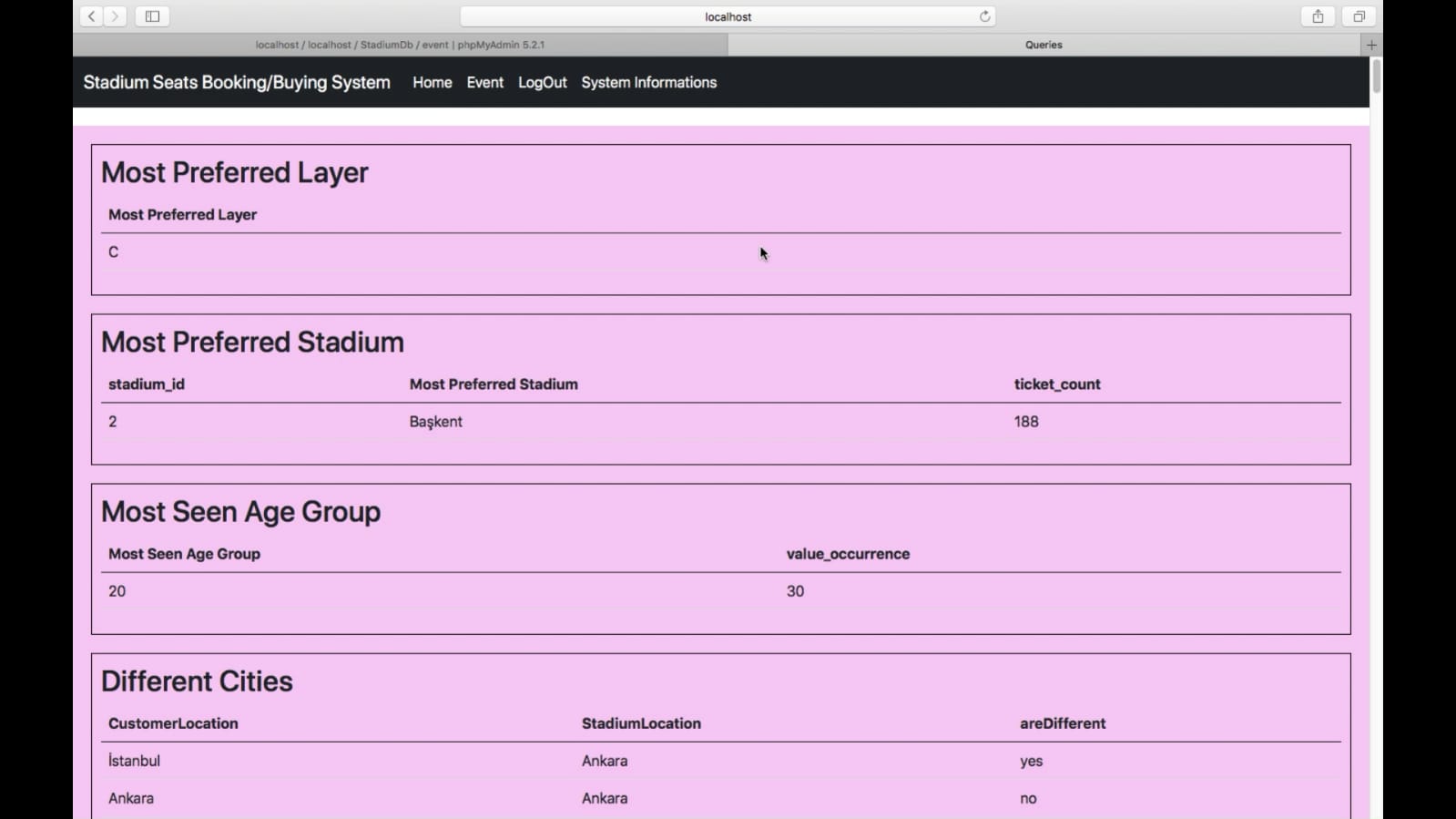
****

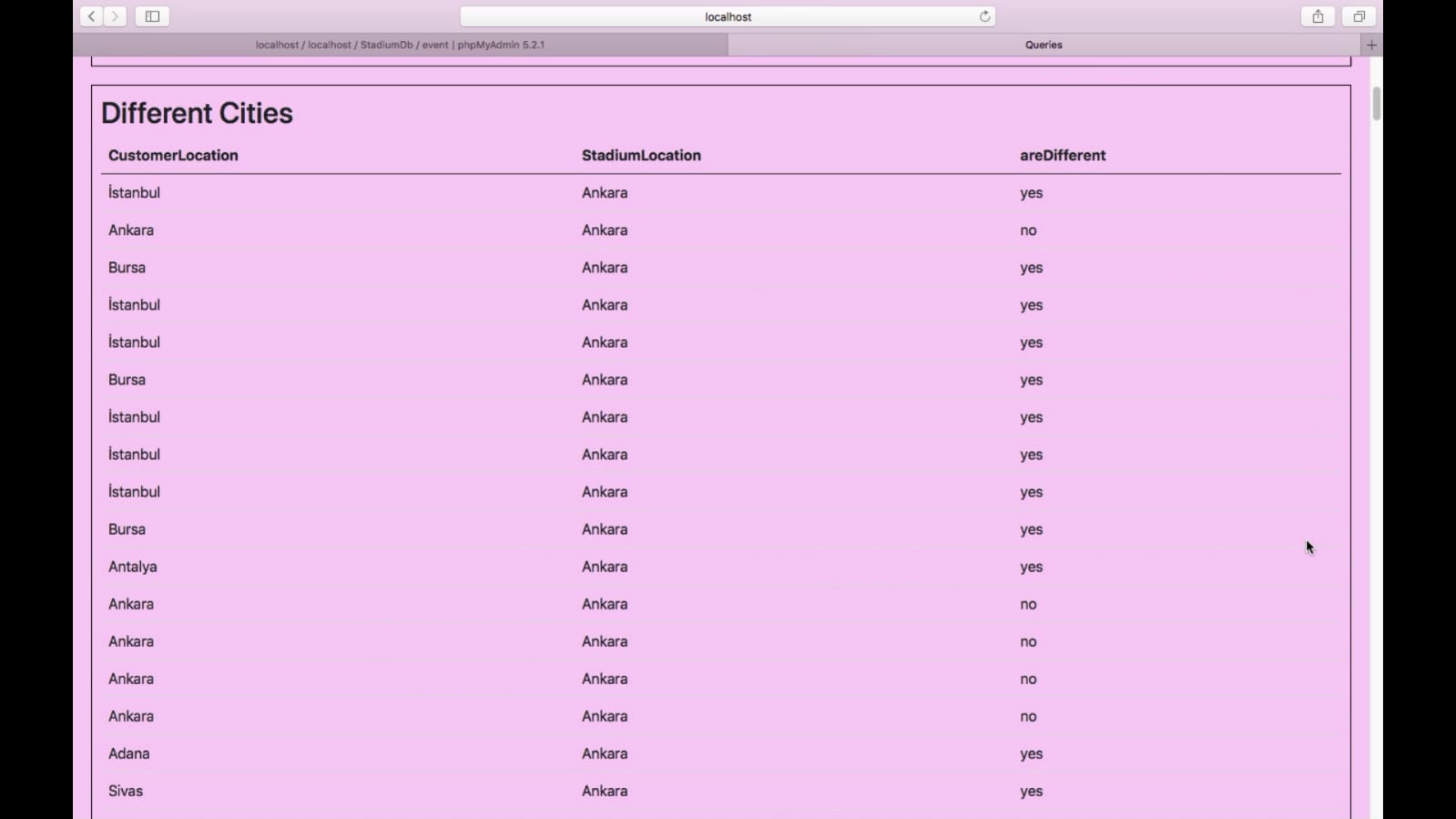
**(admin can delete events by clicking “delete”)**

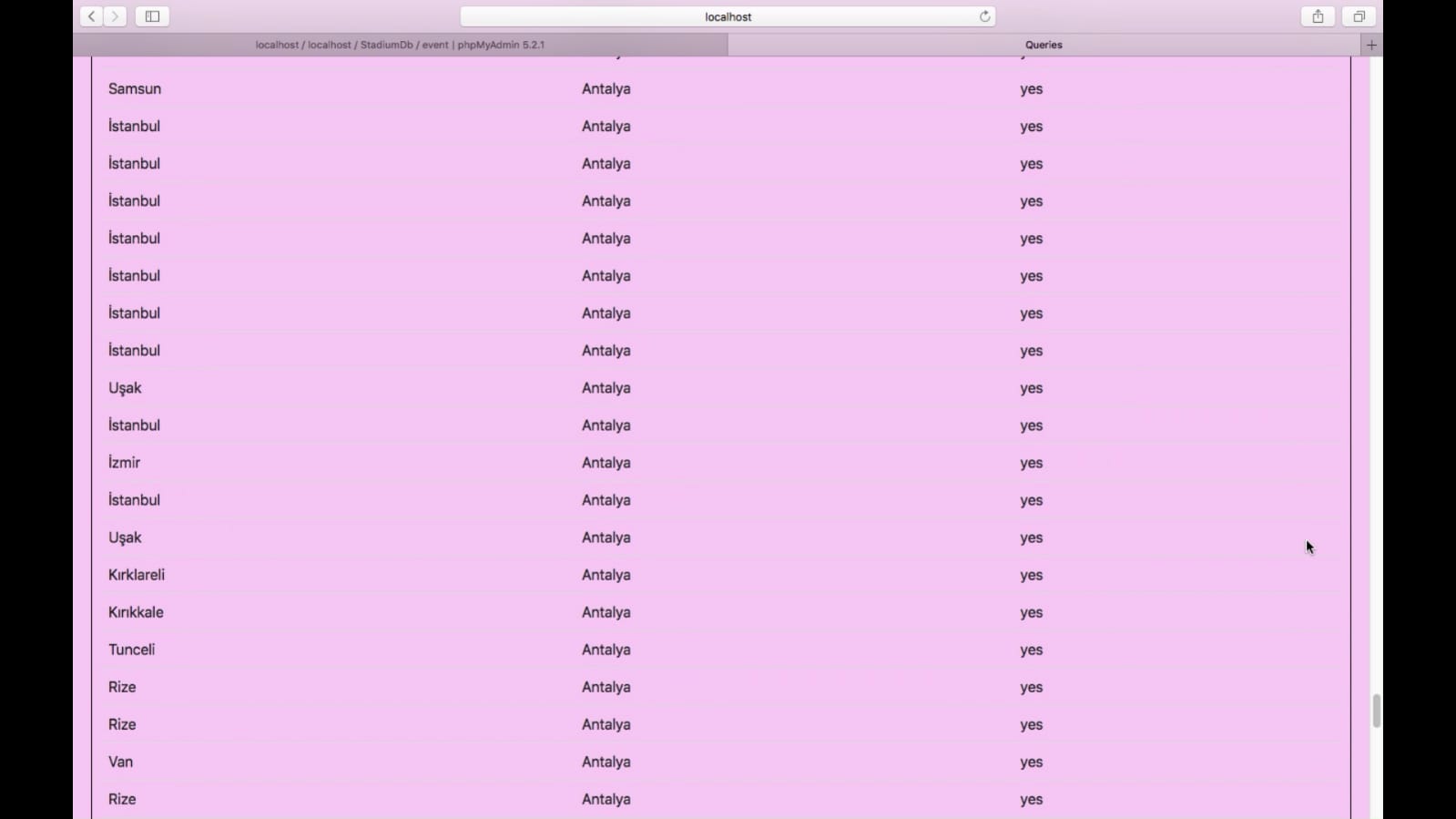
****

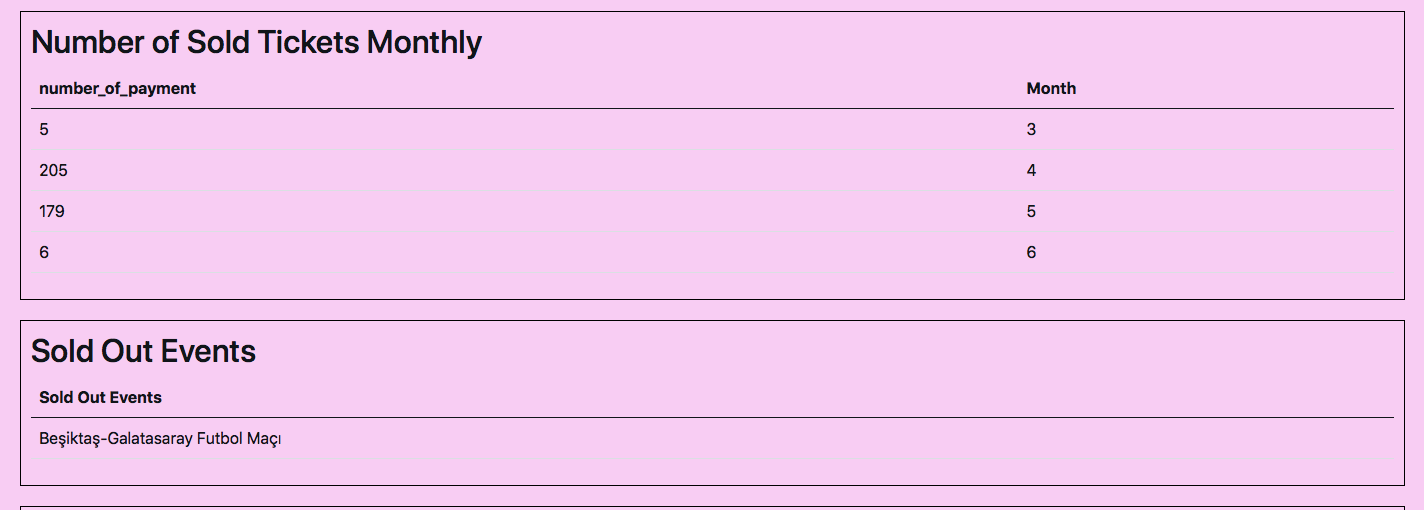
**After delete operation.**

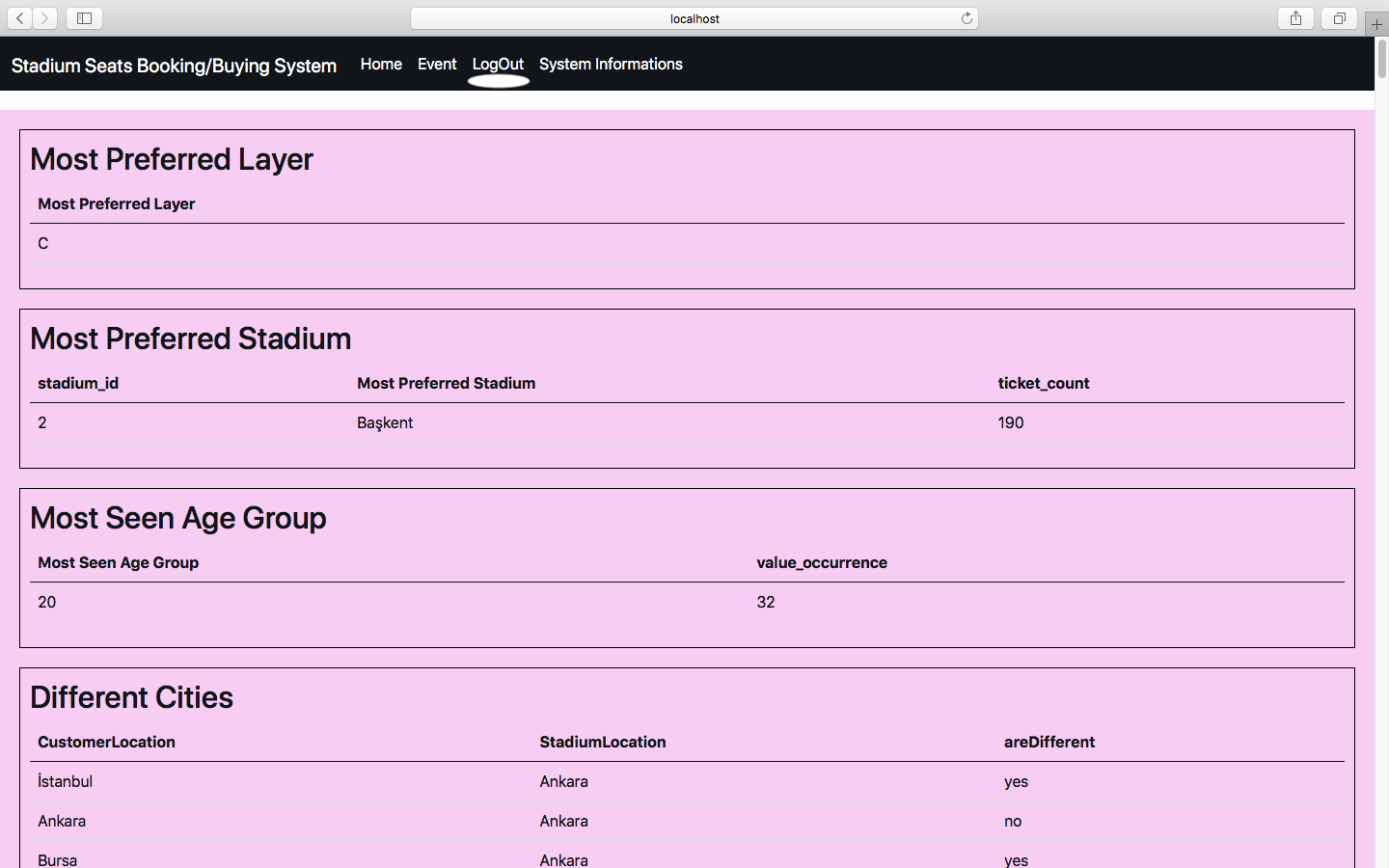
**(admin can view data from the system ) ----QUERY RESULTS**

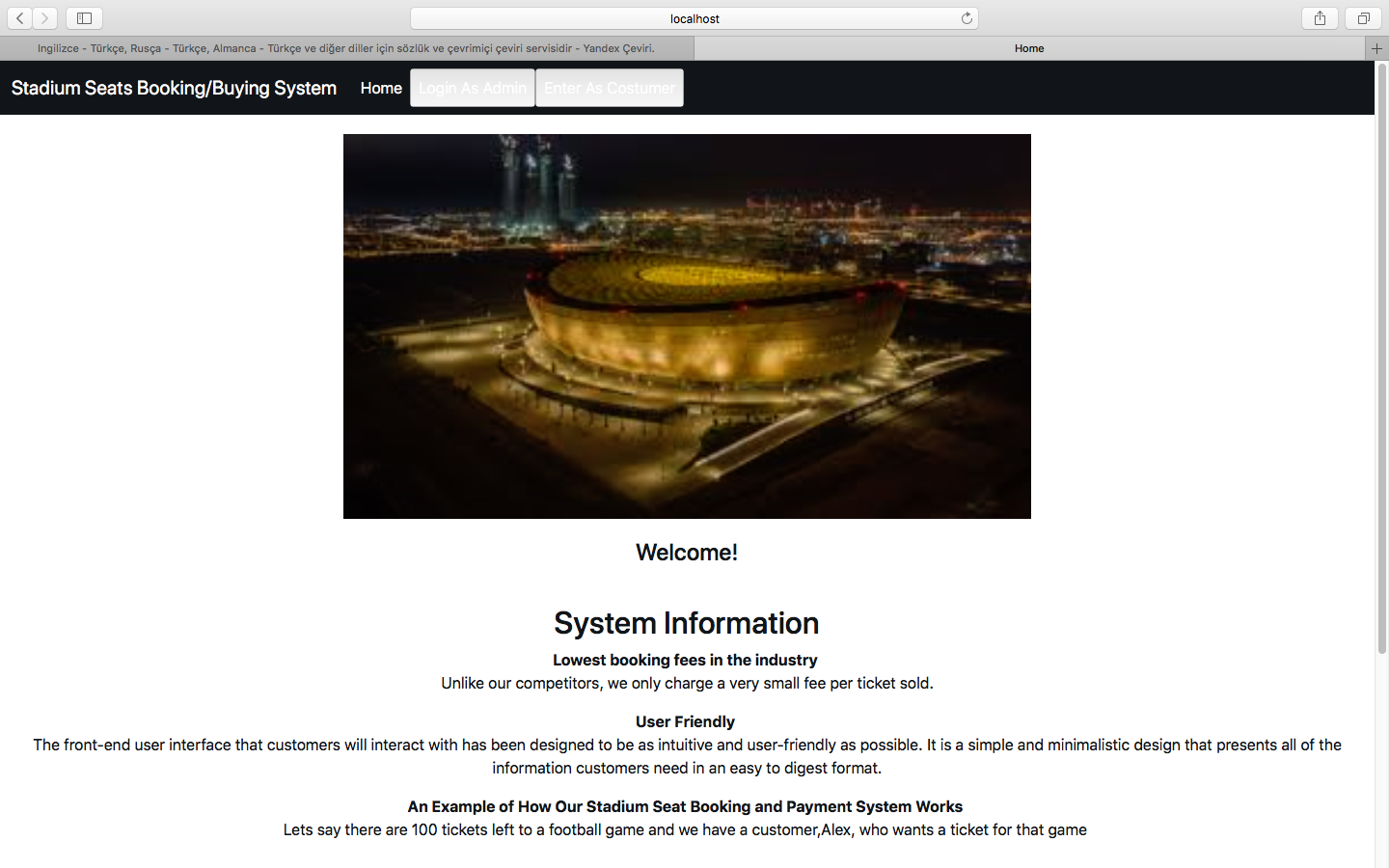
****

****

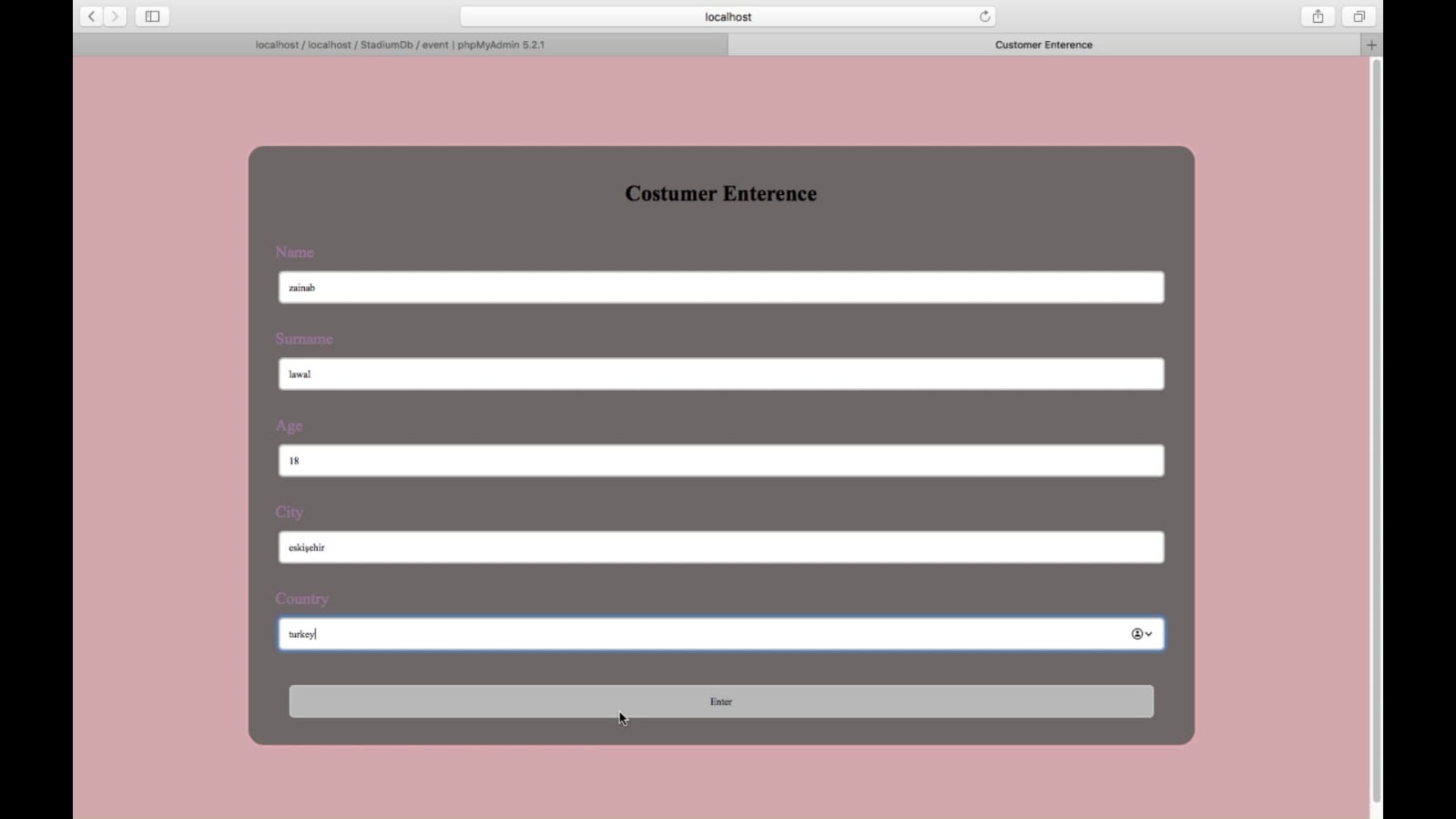
****

****

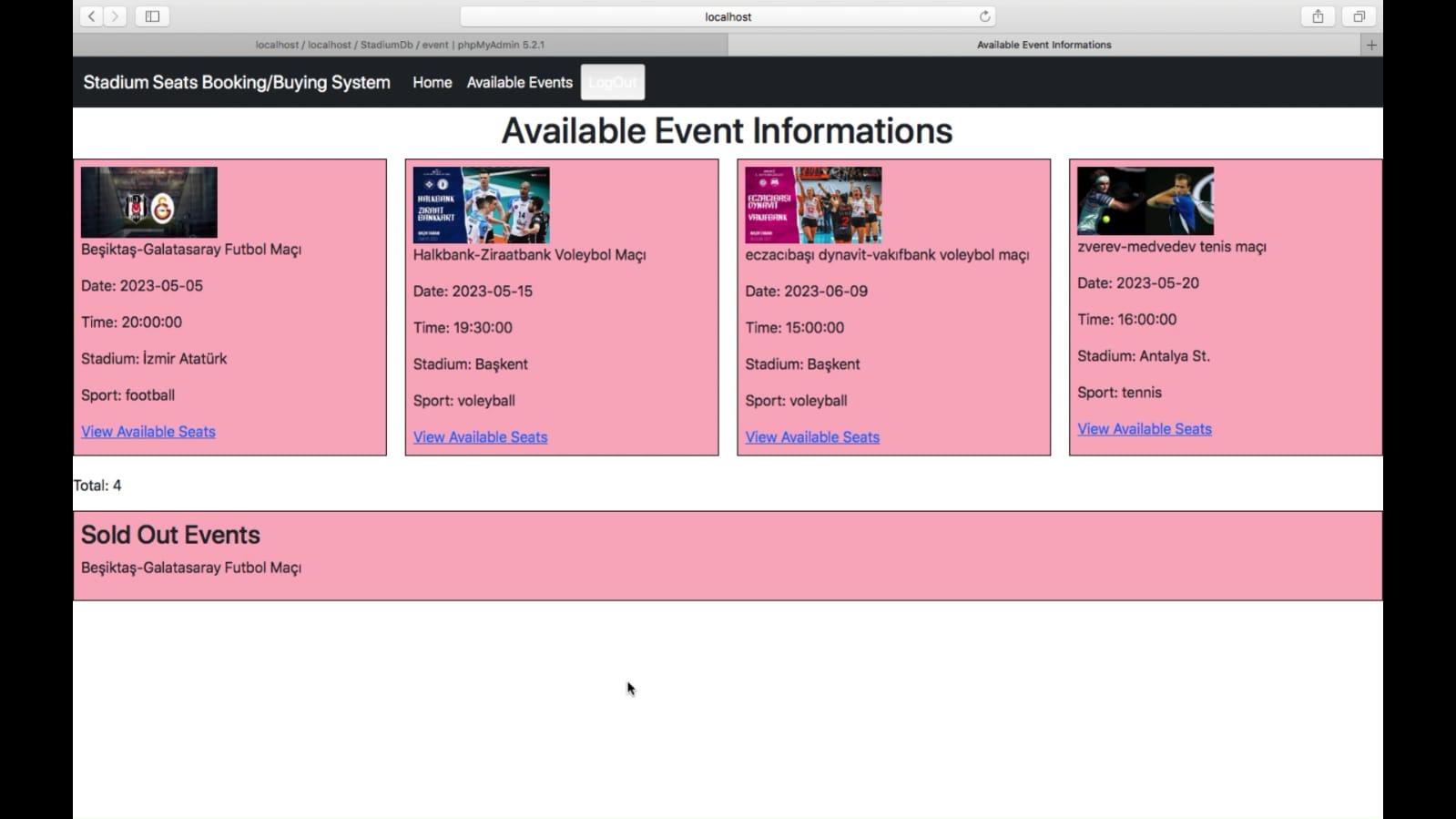
**Admin can logout by pressing the logout button.**

**When the admin log out , s/he directed to home page**

**(customer can log into the system as following)**

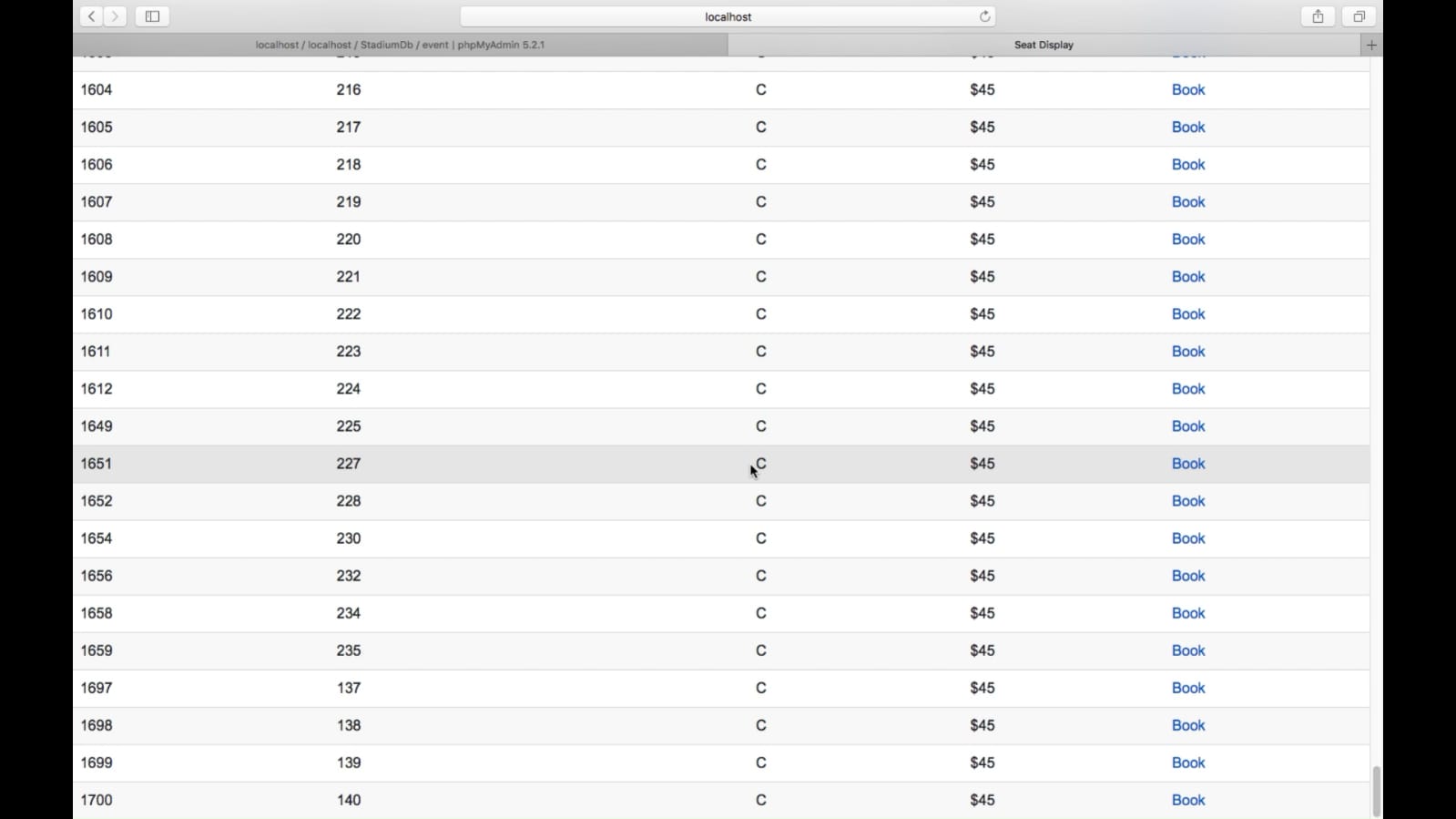
****

**(customer can see all the events avaliable)**

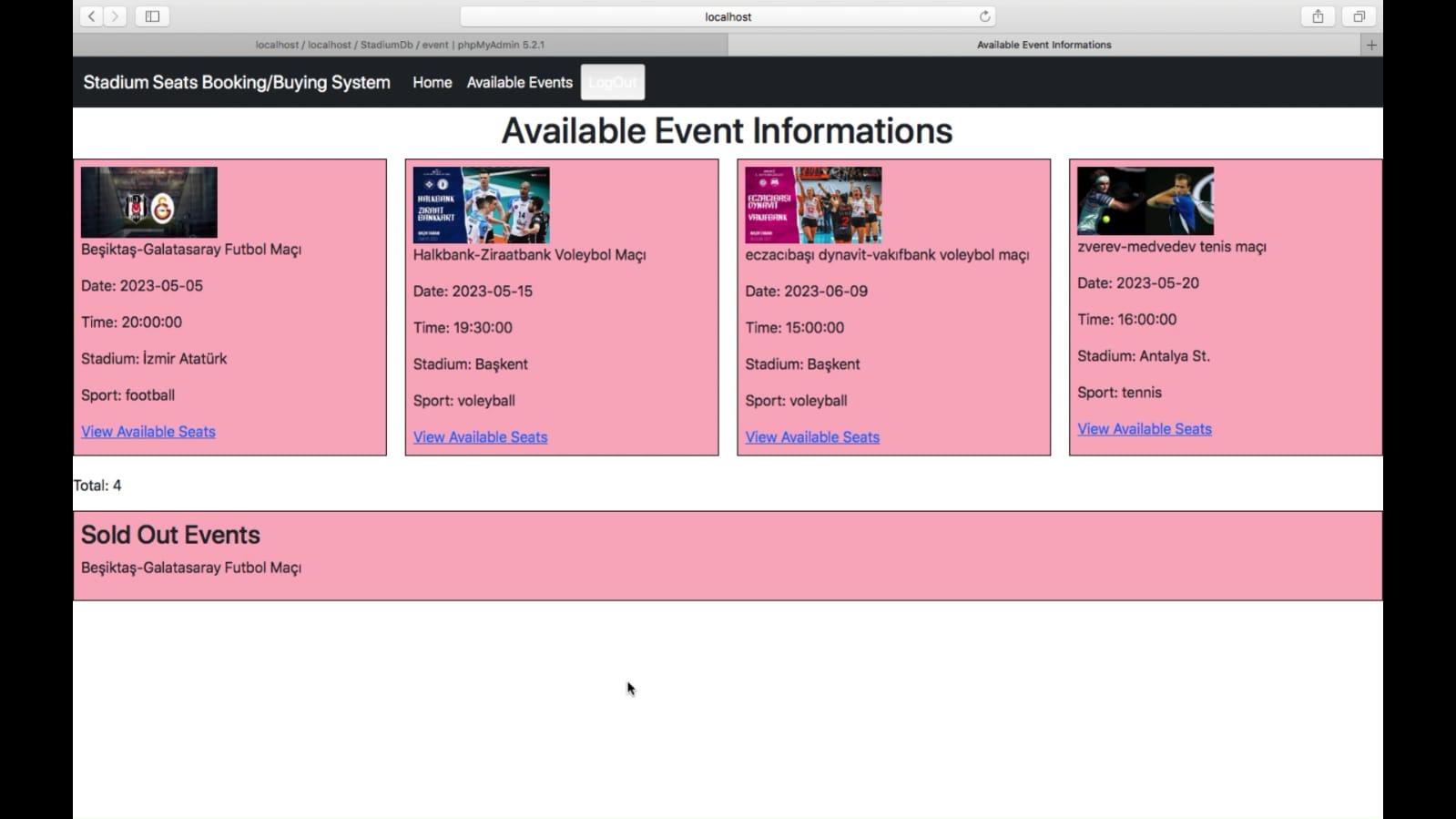
****

**(by clicking on “view avaliable seats”, the system redirects the customer, so he can pick his seat)**

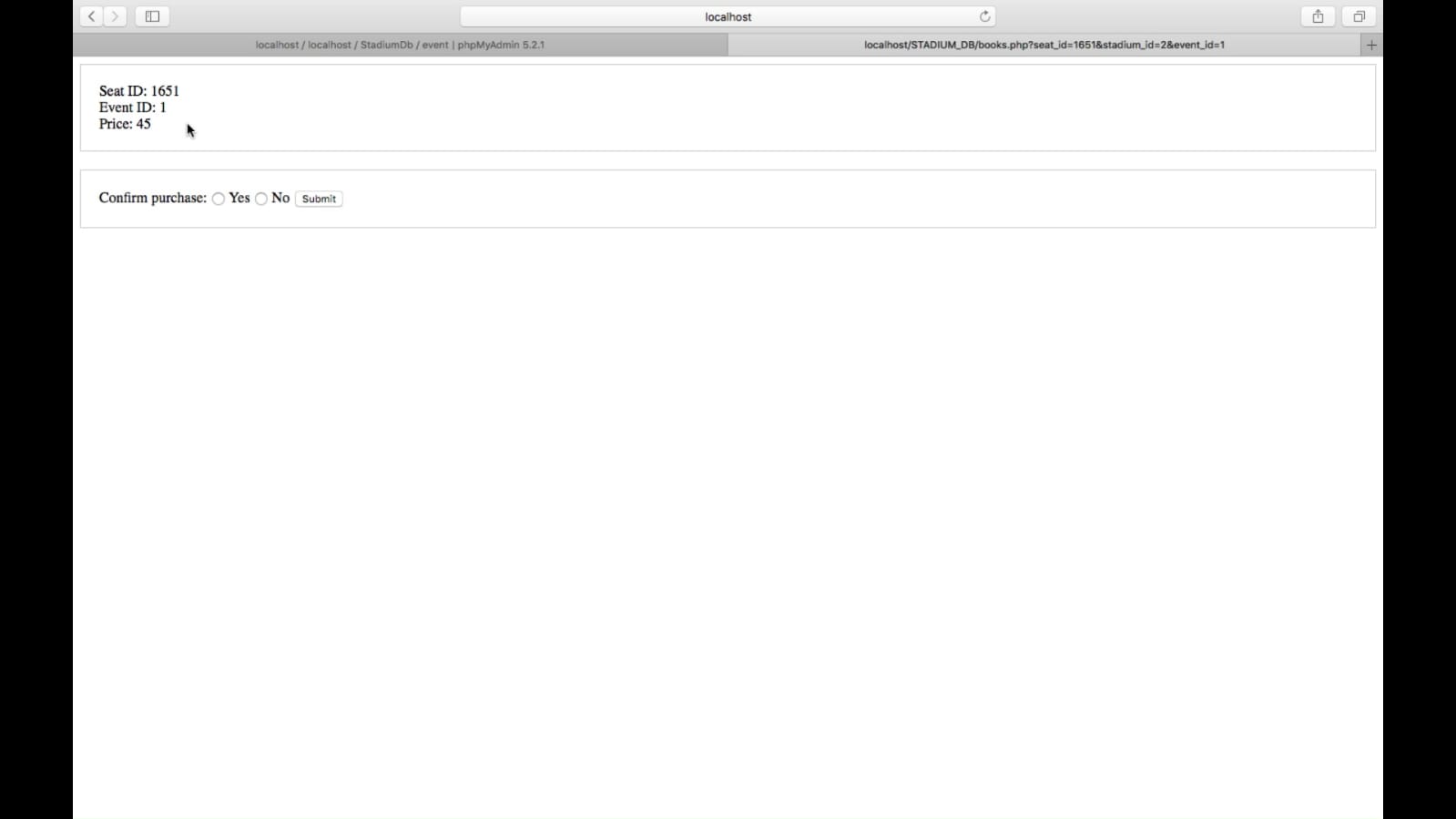
**(by clicking on “book” , the customer books the chosen seat)**

****

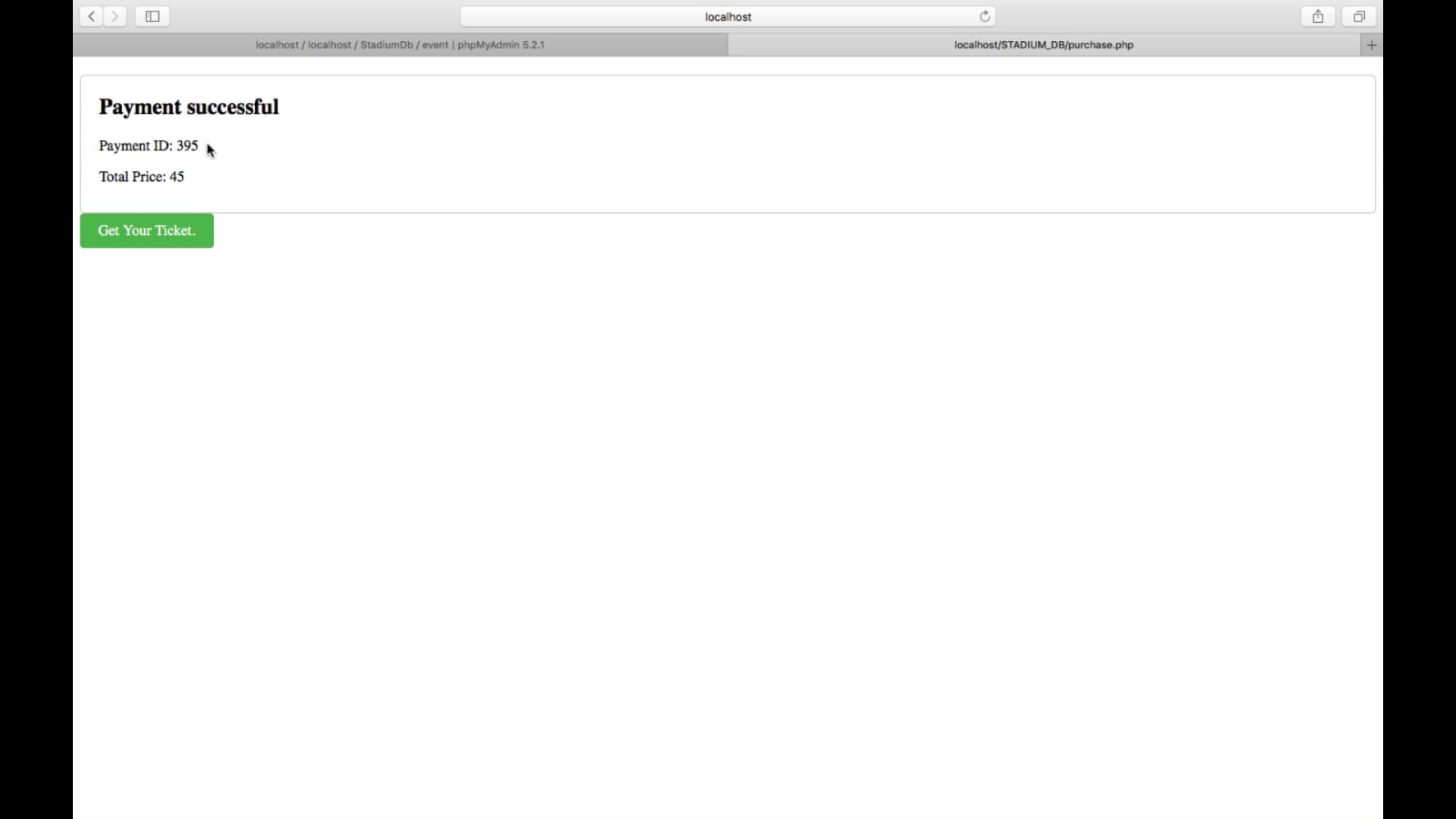
**(when customer books the seat, following box comes. If customer pick “no”, he redirected to the events page.**

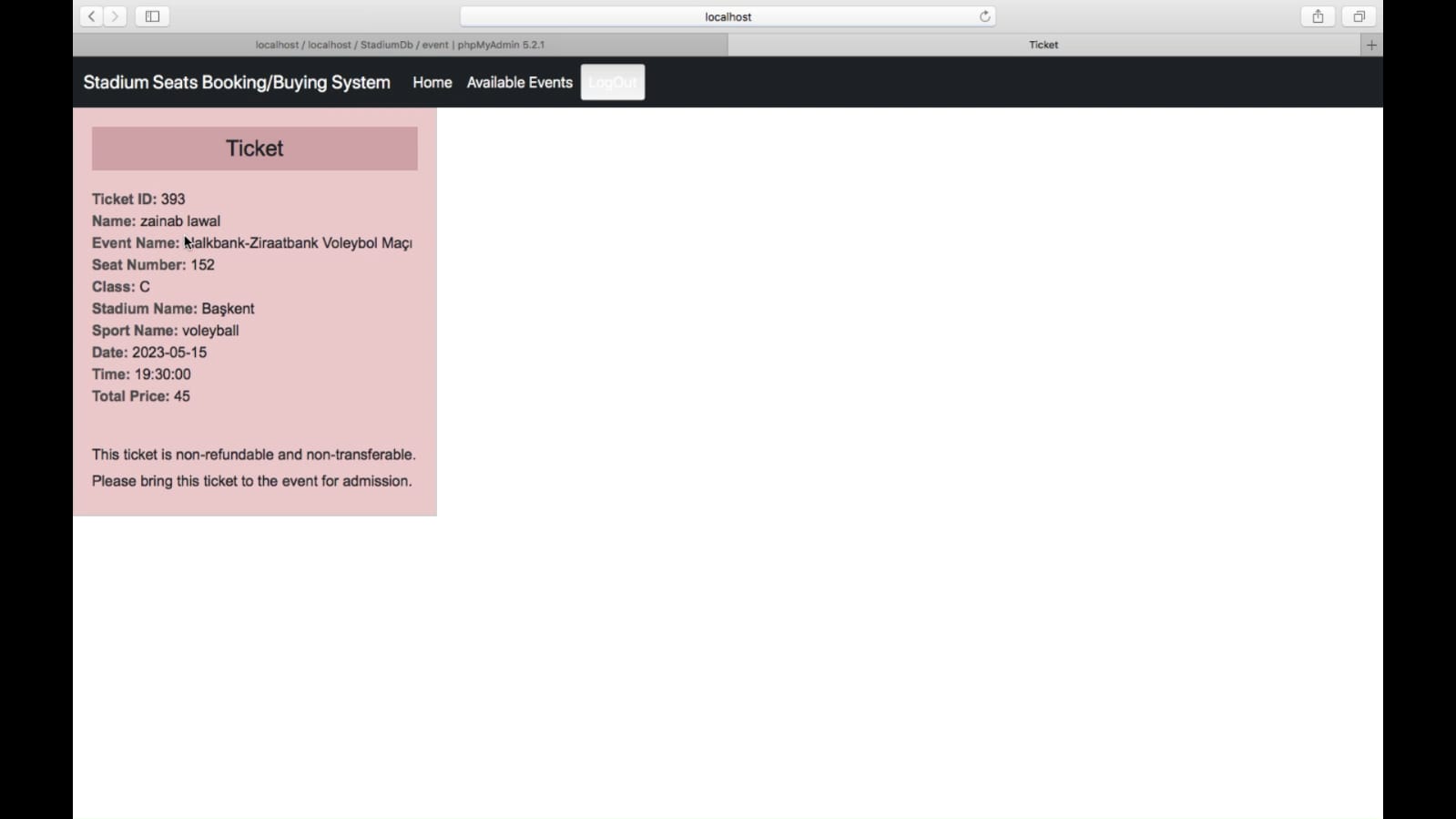
****

**If he pick “yes”, he redirected to payment page, where he can see his payment is successfull.**

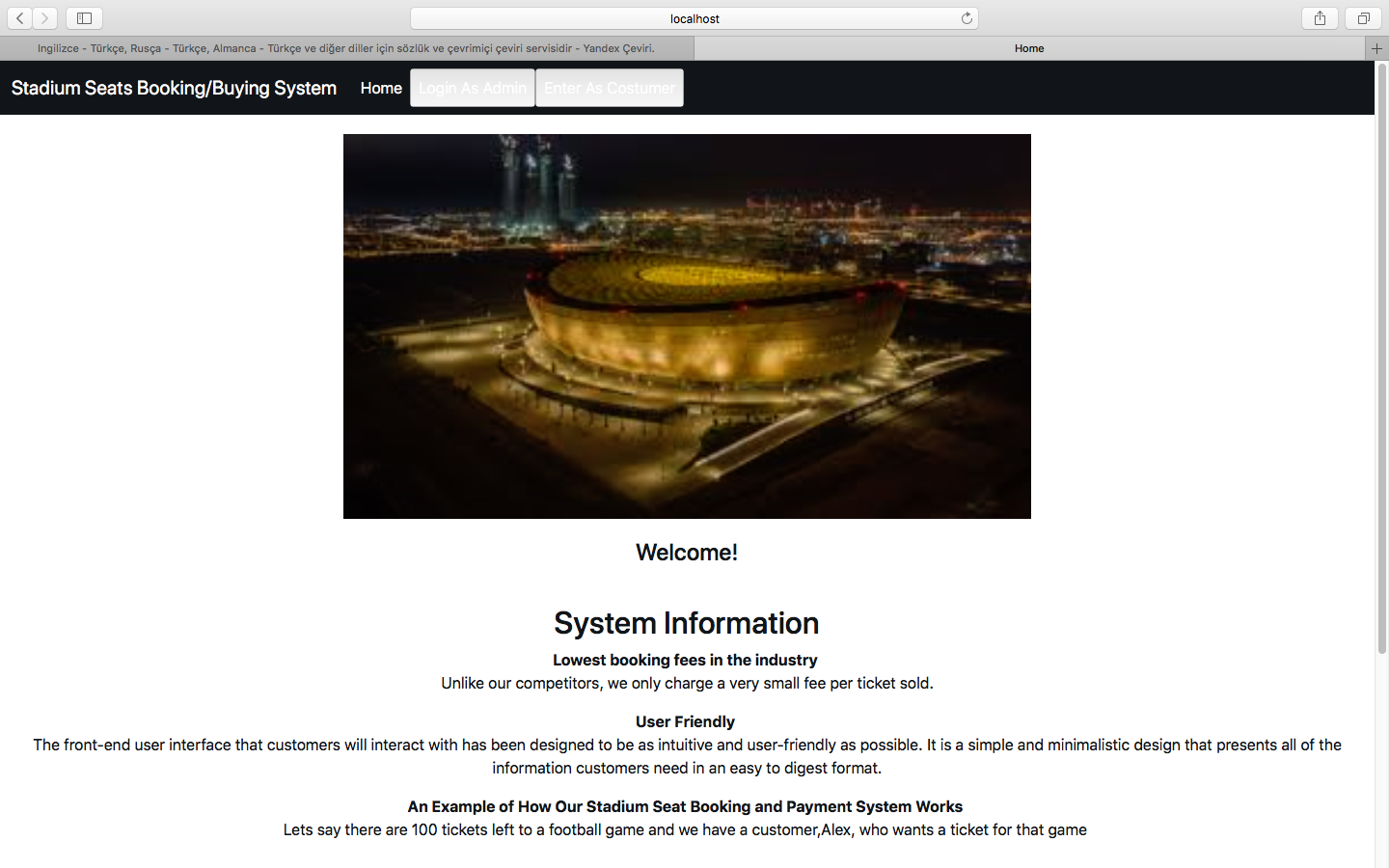
****

**(if yes is picked)**

****

****

**When the costumer click logout button , s/he is out of the system and redirected to the home page.**

****