Movie Booking System

Team ASMV

We have designed solution for small movie theatres with efficient system. As long as a user has access to a phone or computer, they can get easy access to the services available. We are making services easily accessible to the customers means that fewer will be coming to purchase from the ticket window. This might also cut costs for the business because they will not need so many at the ticket windows. Employees will be able to direct their energies elsewhere. Our platform don't need a break or the night off. They can run constantly, giving every customer the benefit of accessing information at any time of the day.

We have made a movie ticket booking system using Python(Tkinter) and Sql. our database is on an online cloud so that any one can download our setup file of our program and avail our booking system directly connected to our online database.

Our project has 2 parts namely:- 1)Customer Interface

2)Admin Interface

1)Customer Interface:-

The very first page of our customer interface has an option to login or sign-up for the

customer. For the login page customer has to enter his email and password with which he has signed-up before. To check the authenticity of the credentials we run a sql query :-

Which returns whether the following credentials exist in the database as 0 or 1.

mycursor.execute('''select case when (%s in (select distinct email from customer) and %s in (select distinct password from customer)) then 1 else 0 end as val''',(email,password))

For the sign-up page when the customer enters his details and clicks the sign-up button the following update query(a) is executed to add the user credentials to the database. another query(b) also ensures that the email is a fresh email which is not already in use.

(a):-

mycursor.execute("insert into customer(name,email,password) values(%s,%s,%s)", (name, email, password))

(b):-

mycursor.execute('''select case when exists(select email from customer where email=%s)then 0 else 1 end as val''',(email,))

After this we reach the movies page which shows us which all movies are available. Names

of the movies will be fetched from our database form the Movie table using the following query:-

The names of the movies will be displayed and the customer can select one movie at a time from the radio-button. We then reach to the theatre and hall page where all the shows of the selected movie which are available and in what all theatre,halls will be displayed on the screen. This data would be fetched from the Shows table using the following query:-

mycursor.execute('select ID,name from movie')

query = '''select s.ID,t.name,start\_time,show\_date,hall\_ID from shows s,theatre t

where s.movie\_ID=%s and s.theatre\_ID=t.ID order by show\_date,start\_time'''

After this when a person selects a movie and a show for that we move to the hall page where all the available seats for the specific show would be present and the person can manually select the open seats. The page uses the Booking table to check which all seats are booked.

Which are then delivered to the interface. Also, the page uses the seatinline table to check which seats are on the process of being booked and blocks those seats that have been accessed within 10 minutes of someone else booking it. The query used is:-

mycursor.execute('''select seat\_ID from books where show\_ID=%s''',(show\_ID,))

mycursor.execute('''SELECT seat\_ID from seatinline where show\_ID=%s and book\_date=curdate()and (cast(curtime() as time)-cast(book\_time as time))<=1000''',(show\_ID,))

mycursor.execute('''insert into seatinline(seat\_ID,show\_ID,book\_time,book\_date) values (%s,%s,curtime(),curdate())''',(i,show\_ID))

The above query updates the.

Then we move onto the payment page and execute the following queries

mycursor.execute('''insert into payment(amt,pay\_time,pay\_date)

values(%s,curtime(),curdate())''',(fl\_amt,))

mycursor.execute('''insert into books

values(%s,%s,%s,%s)''',(custId,i,show\_ID,pay\_ID))

mycursor.execute('''delete from seatinline where seat\_ID=%s and show\_ID=%s''',(i,show\_ID))

The first query inserts payment details in payment table The second query inserts the booking seats in books table

The third query removes the booked seats from the seatinline page as they are no longer in a transition phase but are already booked.

2)Admin Interface

mycursor.execute('''insert into movie(name,language,genre,length) values(%s,%s,%s,%s)''',(name,language,genre,length)) mycursor.execute('''insert into theatre(name,road,city,pincode) values(%s,%s,%s,%s)''',(name,road,city,pincode)) mycursor.execute('''insert into shows(movie\_ID,hall\_ID,theatre\_ID,start\_time,end\_time,show\_date,price) values (%s,%s,%s,%s,%s,%s,%s)''',(mov\_ID,h\_ID,t\_ID,val1,en\_time,val2,val3))

The above queries are used to update the movie, theatre, hall and shows tables.