**//incomplete additional parts written by ariyan of full report**

**Content**

Acknowledgment ………………………………………………………………. 4

Abstract ………………………………………..……………………………….. 6

Introduction ……………………………………………………………………. 7

Project Description ……………………………………………………...…....... 8

List of Entities & Attributes ……………………………………………………. 10

ER Diagram(Conceptual Model) ......................................................................... 12

Create & Insert SQL Queries …………………………………………………... 16

SQL Queries related to Report Generation …………………………………..... 19

Project Based Screenshot ……………………………………………………... 22

Conclusion ……………………………………………………………………... 24

**Abstract**

Our project is a movie theatre management system of the theatre name being “Dhaka Cinemax” Users of the movie ticket booking system can inquire about the movies that are currently playing, the ability to reserve and cancel movie tickets, and inquire about the status of the reserved tickets, the theater type and class type, etc. This project's goal is to create a website that provides a simple platform for people to learn about popular movies and purchase cinema tickets in the Making it as simple as possible for everyone to purchase tickets at any time

anywhere, too. This project includes an introduction to the mechanism for purchasing cinema tickets. It is an automated system for making reservations for movie seats in advance. This virtual ticket for a movie the use of a booking system can make purchasing movie tickets much simpler than ever before. Additionally, this project includes an introduction to the relational model and an example of basic SQL queries that can be used to obtain data from the system's database based on an entity relationship model diagram based on a system for booking movie tickets.

**Introduction**

A structured collection of data is a database. In most cases, the data is set up to enable information-intensive activities by modeling. End users can add, read, update, and remove data in a database thanks to a DBMS. Data is continuously organized and kept accessible thanks to the DBMS, which effectively acts as an interface between the database and users or application applications. The database management system (DBMS) controls three crucial elements: the data, the database engine that enables data access, locking, and modification, and the database schema that establishes the logical structure of the database. Concurrency, security, data integrity, and consistent administrative practices are all supported by these three fundamental components. The DBMS can provide independent access to logical and physical data. This means that it can shield users and programs from having to know where data is kept or from being concerned about modifications to the data's physical structure.

The main goal of maintaining a database for a movie ticket booking system is to minimize manual errors that occur during ticket booking and cancellation, as well as to make it easier for customers and service providers to keep track of information about their clients and the seats that are available to them. Many of the gaps that exist in manual record maintenance can be closed by automation. Data gathering and processing will happen quickly. The proposed system can be web enabled for future growth so that customers can inquire about booking movie tickets in a variety of ways. As a result, they occasionally run into a lot of issues and have several conflicts with clients. We create a database that includes customer information, movie seat availability, etc. to address the aforementioned issue.

Project Description

The goal of this project is to create a web application for ordering movie tickets. The method for purchasing movie tickets enables the user to conduct research, learn about popular films, and determine whether seats are still available for certain films based on theater type, class type, etc. This website will offer consumers a simple method for ordering and canceling movie tickets, finding out the status of ordered tickets, offering system comments, etc. Additionally, this ticket booking website accepts a bkash transaction receipt page where unique user id is generated including the mobile number required to send money to the theatre and users need to show this upon entry of the theatre. Additionally, in the contact page has contact page where users may provide us their inquiry with emails also the map location of the cinema hall is also given.

Additionally, there is an admin dashboard that allows system providers to monitor the operation of the system. Through this admin dashboard system, service providers can view all reservations for movies, add new entries for movies in accordance with newly released titles, amend data, and in some situations, delete reservations. Additionally, service providers monitor the number of reservations made through this admin dashboard system. Additionally, through this admin dashboard system, providers keep tabs on fresh movie releases and convey the same information to consumers. They also keep tabs on recent user bookings. Users receive an electronic ticket with information on the movie time, theater type, and class.

**List of entries and attributes:**

|  |  |
| --- | --- |
| Entities | Attributes |
| bookingtable | **bookingID**  movieID  bookingTheatre  bookingType  bookingDate  bookingTime  bookingFName  bookingLName  bookingPNumber  bookingEmail  amount  ORDERID  DATE-TIME |
| movietable | **movieID**  movieImg  movieTitle  movieGenre  movieDuration  movieRelDate  movieDirector  movieActors  mainhall  viphall  privatehall |
| users | **id**  username  name  password |

Create and Insert into SQL queries

Create Table Queries:

1. bookingtable

Query:

CREATE TABLE `bookingtable` (

`bookingID` int(11) NOT NULL,

`movieID` int(11) DEFAULT NULL,

`bookingTheatre` varchar(100) NOT NULL,

`bookingType` varchar(100) DEFAULT NULL,

`bookingDate` varchar(50) NOT NULL,

`bookingTime` varchar(50) NOT NULL,

`bookingFName` varchar(100) NOT NULL,

`bookingLName` varchar(100) DEFAULT NULL,

`bookingPNumber` varchar(12) NOT NULL,

`bookingEmail` varchar(255) NOT NULL,

`amount` varchar(255) NOT NULL,

`ORDERID` varchar(255) NOT NULL,

`DATE-TIME` datetime NOT NULL DEFAULT current\_timestamp()

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

1. movietable

Query:

CREATE TABLE `movietable` (

`movieID` int(11) NOT NULL,

`movieImg` varchar(150) NOT NULL,

`movieTitle` varchar(100) NOT NULL,

`movieGenre` varchar(50) NOT NULL,

`movieDuration` int(11) NOT NULL,

`movieRelDate` date NOT NULL,

`movieDirector` varchar(50) NOT NULL,

`movieActors` varchar(150) NOT NULL,

`mainhall` int(11) NOT NULL,

`viphall` int(11) NOT NULL,

`privatehall` int(11) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

3)users

Query:

CREATE TABLE `users` (

`id` int(11) NOT NULL,

`username` varchar(80) NOT NULL,

`name` varchar(80) NOT NULL,

`password` varchar(80) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

Insert Queries:

1. booking table

INSERT INTO `bookingtable` (`bookingID`, `movieID`, `bookingTheatre`,`bookingType`, `bookingDate`, `bookingTime`, `bookingFName`, `bookingLName`, `bookingPNumber`, `bookingEmail`,`amount`, `ORDERID`,`DATE-TIME`) VALUES (38, 1, 'private-hall', '7d', '13-3', '15-00', 'Bart', 'Simpson', '0182829892', 'bartsimpson@gmail.com', '200.00', 'BKSH74294887', '2022-09-25 18:07:24');

1. movietable

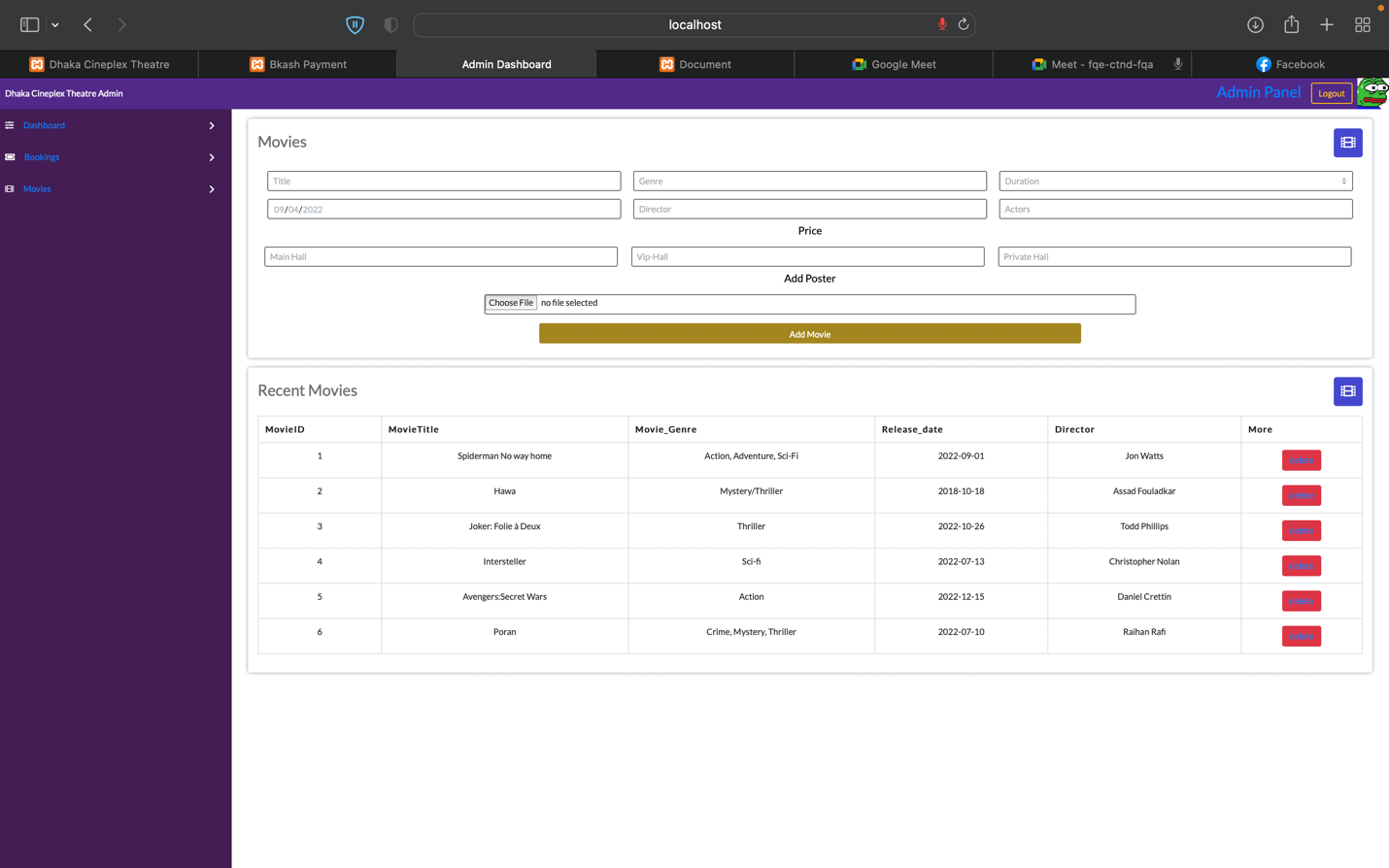
INSERT INTO `movietable` (`movieID`, `movieImg`, `movieTitle`, `movieGenre`, `movieDuration`, `movieRelDate`, `movieDirector`, `movieActors`, `mainhall`, `viphall`, `privatehall`) VALUES (1, 'img/movie-poster-1.jpg', 'Avengers:Secret Warsl', ' Action, Adventure, Sci-Fi ', 220, '2022-10-18', ‘Tom Holland’, Chris Hemsworth', 'Chris Patt, Robery Downey, Chris Evans, 0, 0, 0);

1. users

INSERT INTO `users` (`id`, `username`, `name`, `password`) VALUES (1, '123', 'Bart', '123');

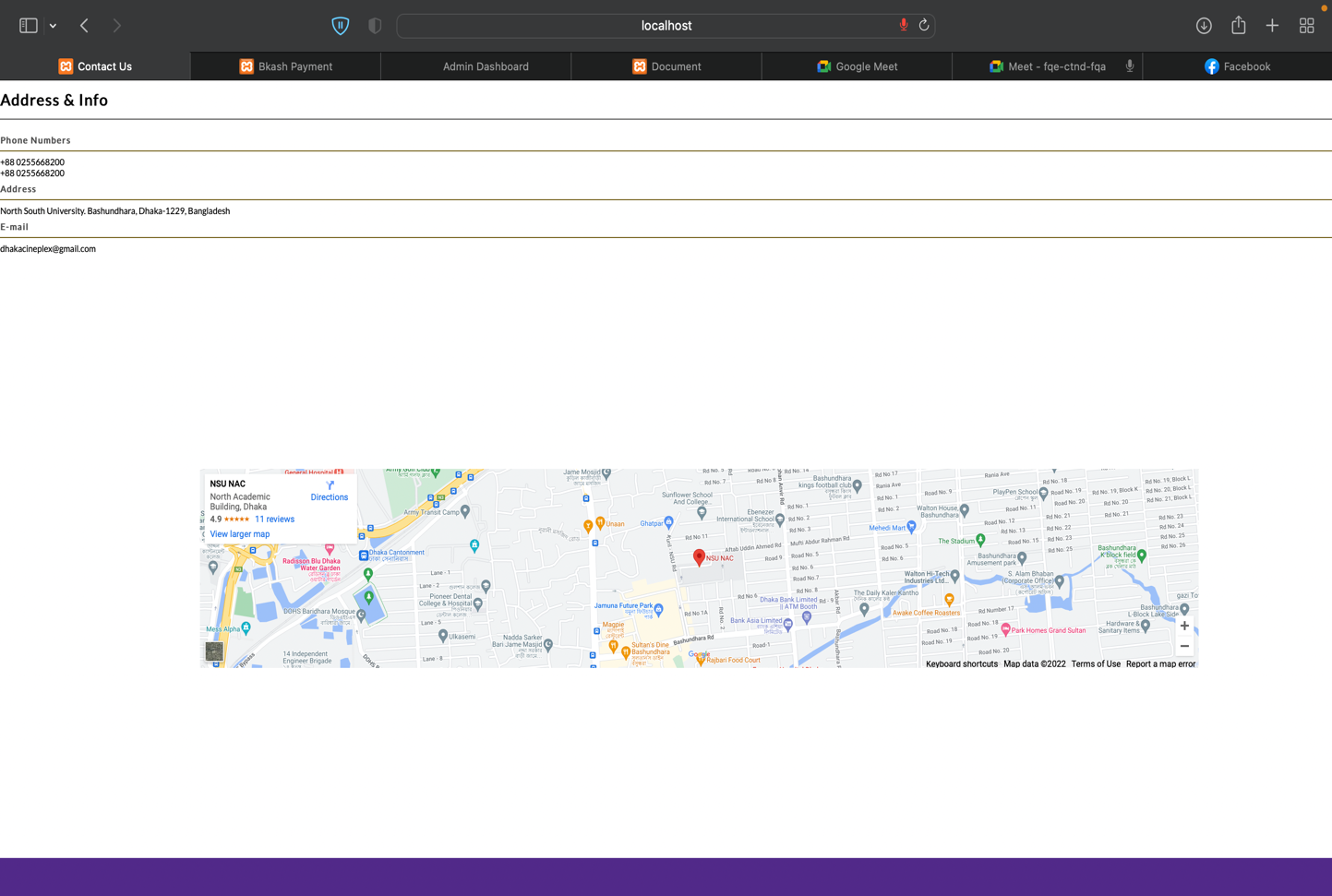
**Project Screen Shots**

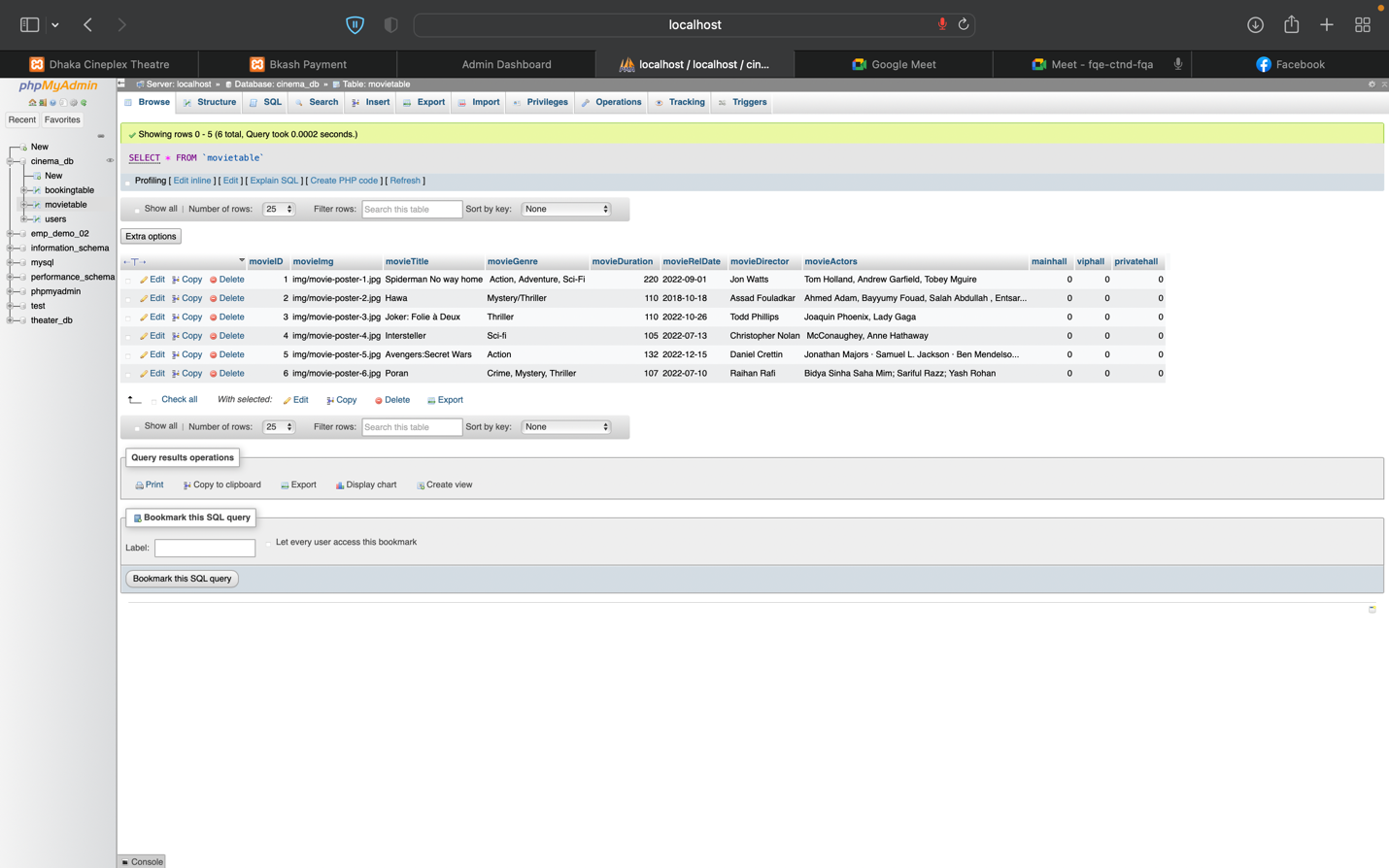
**1.Admin Dashboard**

****

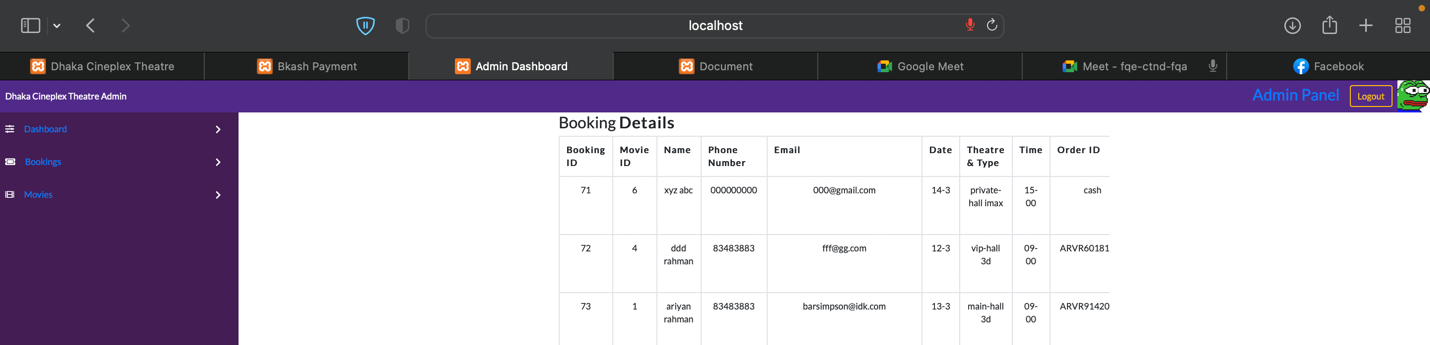
**2.Admin Login**

****

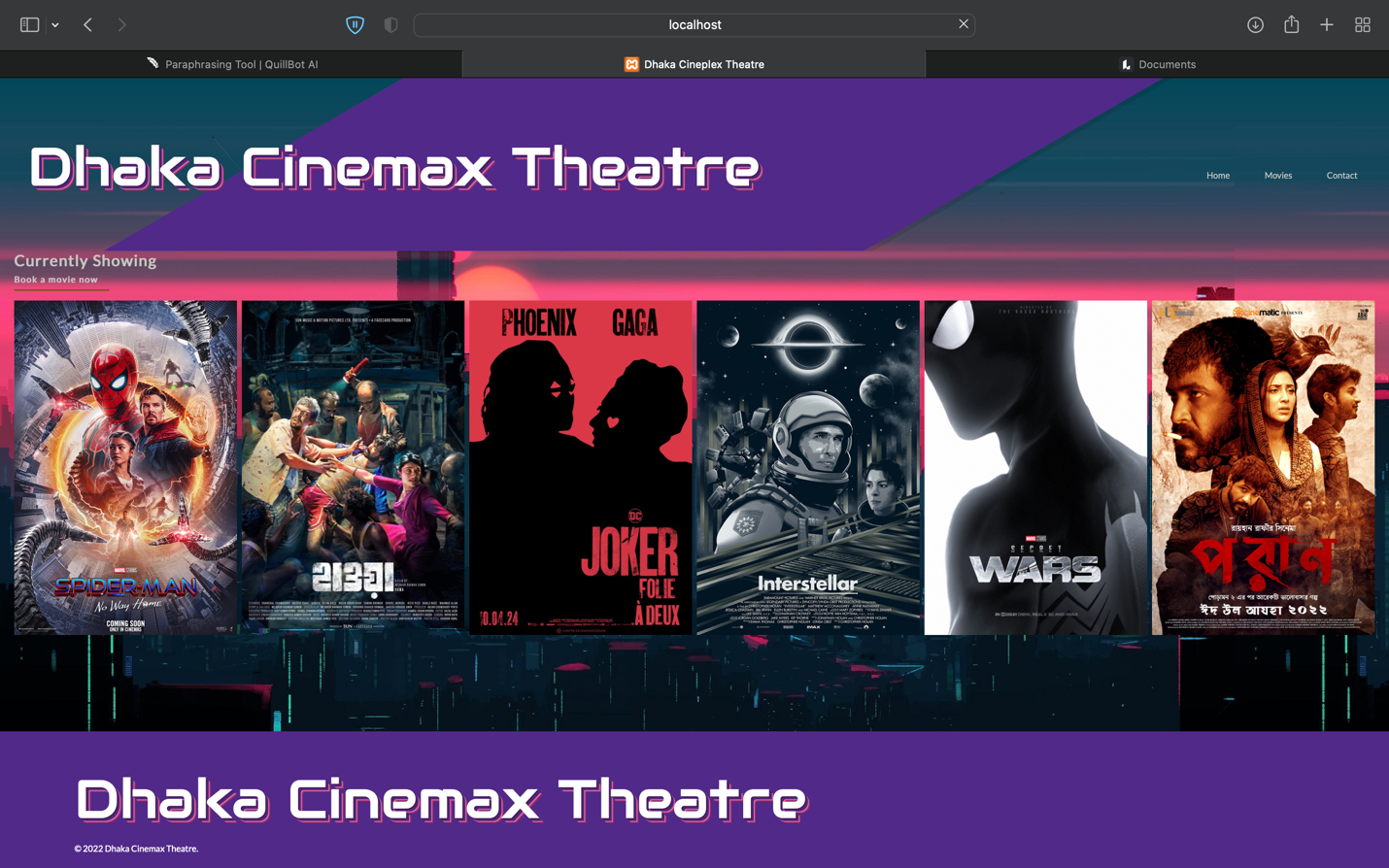
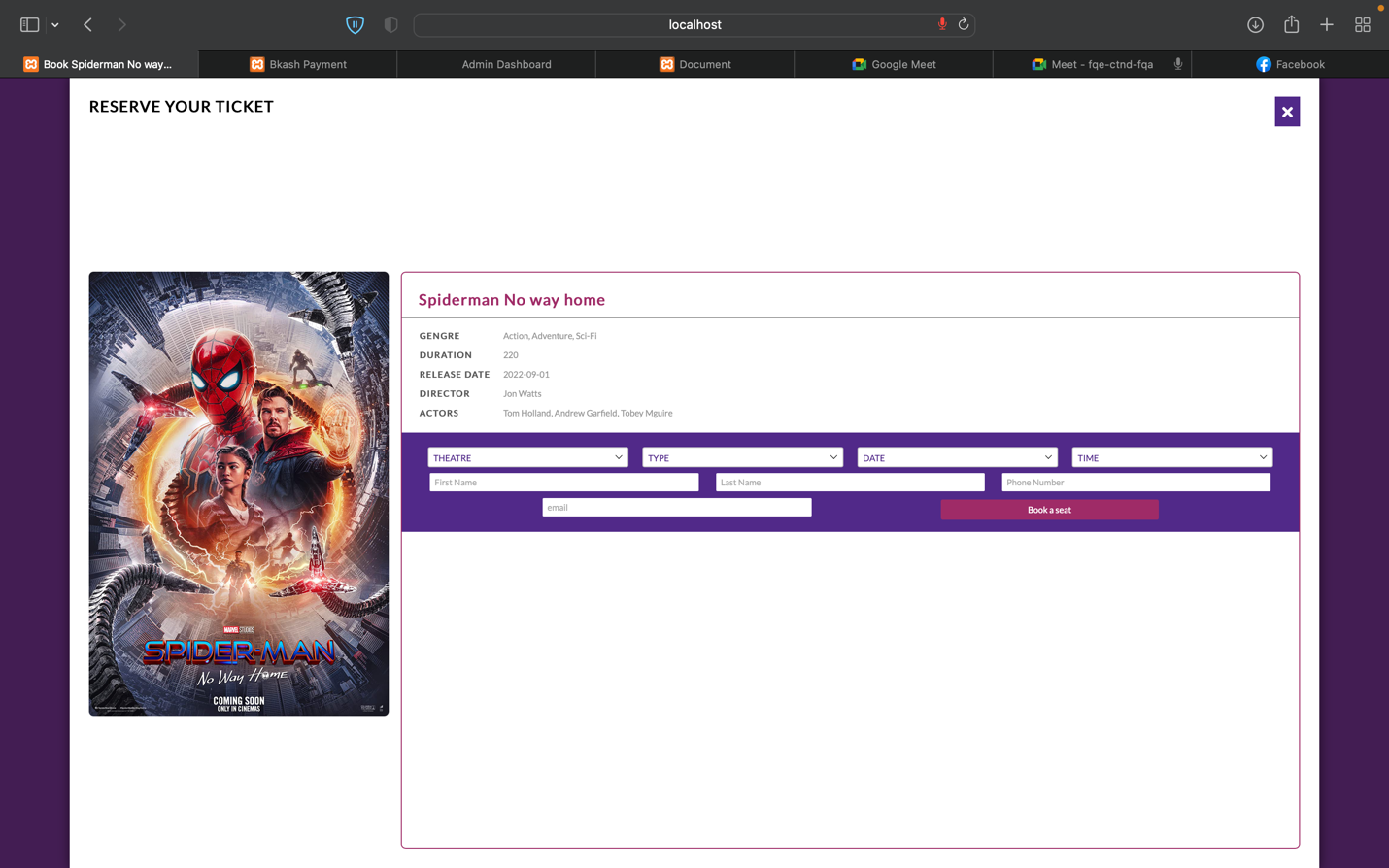
**3.Contact Page with Map location**

**4.Database Screenshon**

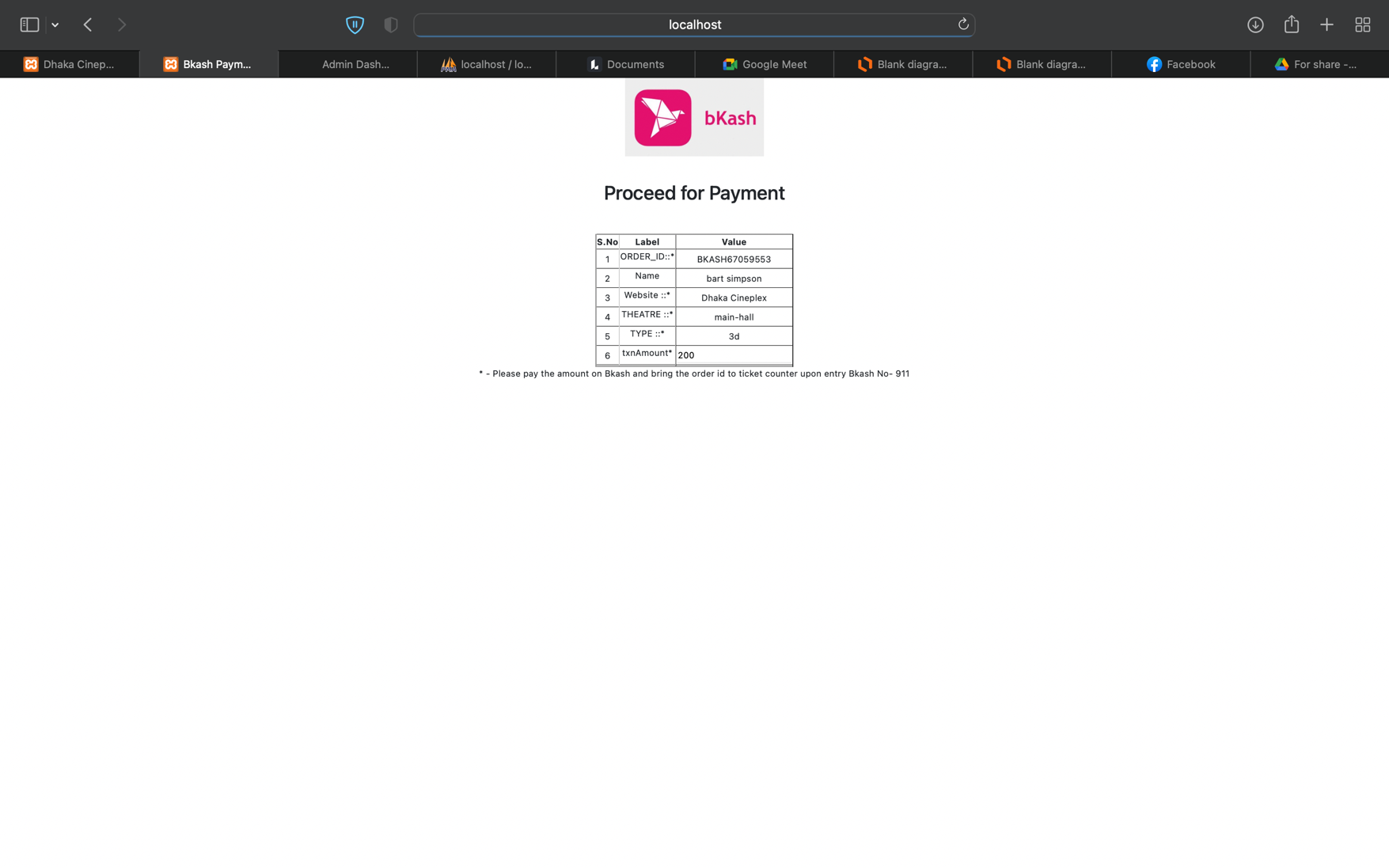
5.Booking Details

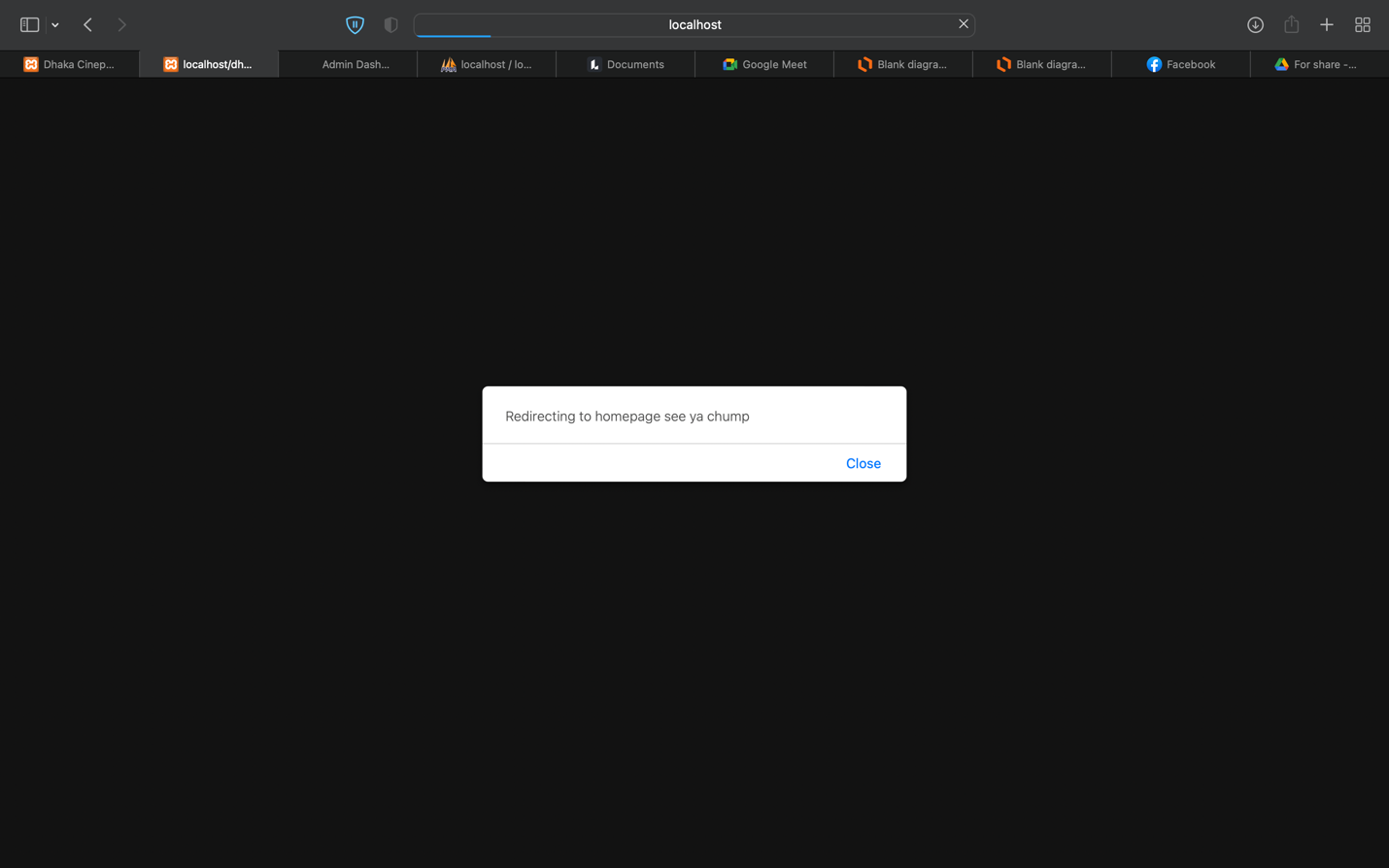
****

**6.Website Homepage**

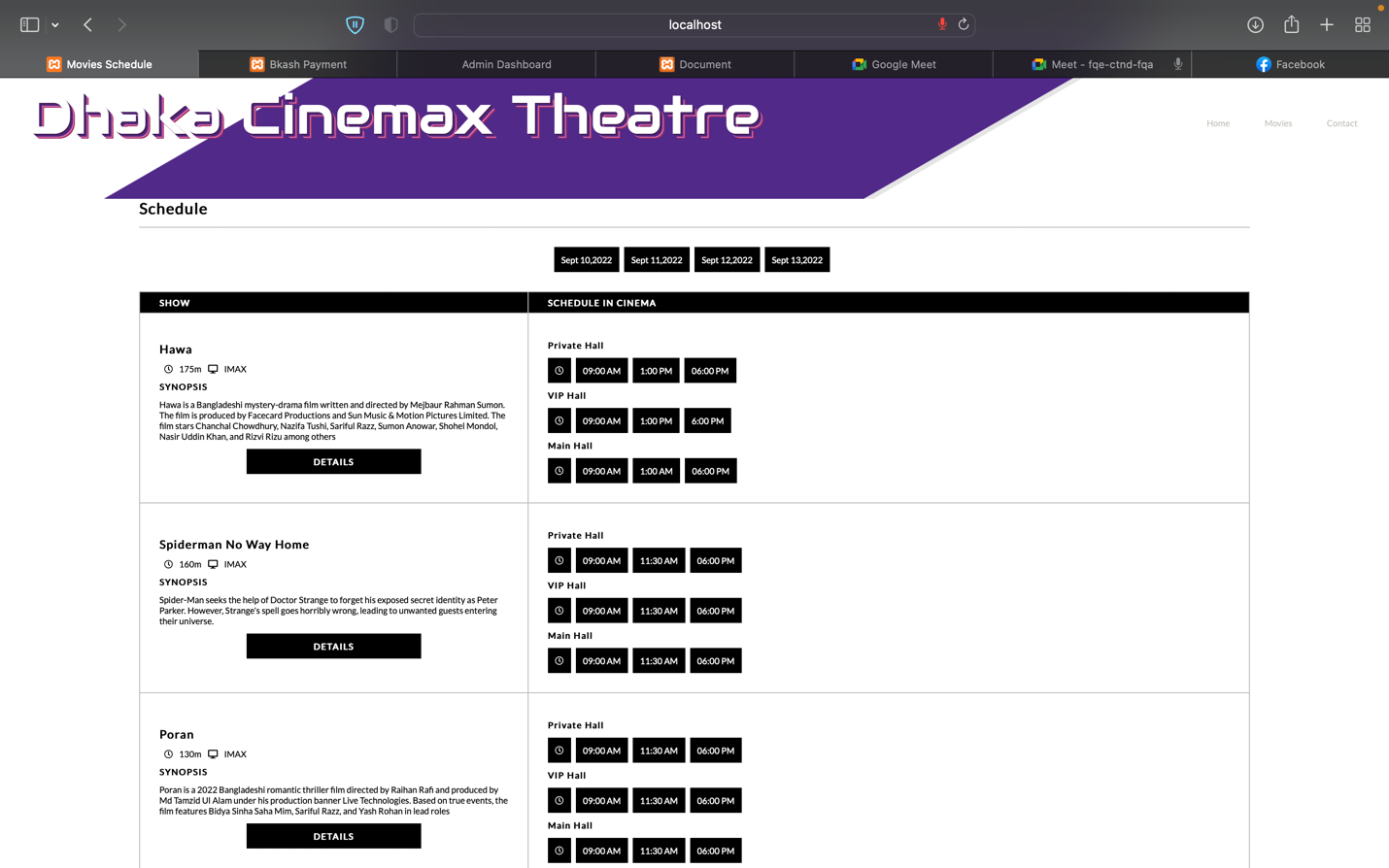
**7.Movie Ticket Reservation**

**8.Bkash Receipt Page**

****

**9.Page reload prompt**

**10.Movie Schedule Page**

****

**Technologies Stack:**

**Front End:**

**1.HTML**

**2.CSS**

**3.Java Script**

**Database:**

**1.MySql**

**2.PhpMyAdmin(XAMPP)**

**Backend:**

**1.Php**

**Limitations:**

With the short amount of time, we received we could only implement the front-end UI index of the homepage of the website having 6 movies showing. In further down the line, we would like to implement a modern parallax UI design with arrow keys showing new upcoming movies. We would further would like to add an option to show user reviews out of 10 of the selected cinemas and also add trailer links of the website. We would also like to add snack options for people to purchase foods in advance. Also, user feedback which the admin can receive directly or a live chat feature to help the customers. Finally, we would like to implement a proper Bkash transaction API which would help customers pay fully and send them an email of their purchased ticket containing details of their booking id, phone, movie name and hall

**Conclusion**

In our project we have kept track of all the information regarding movie times, users who are purchasing tickets, and the current situation regarding the availability of seats for a specific theater, film, class type, etc. This system will be useful for customers who wish to reserve cinema tickets in the simplest manner possible, as well as learn about the most current movie releases and learn a little bit about movies. Only the most crucial needs have been taken into account thus far; many additional features and information can be added to our project in order to produce even more user-friendly applications. These applications are currently being developed, and in the future, they may be improved and integrated with cutting-edge technology.