TheatreTicketManagementSystem

A Mini Project Report

Submitted by

MADHAVS-23CDR083

MANOHARANR-23CDR088

in partial fulfillment of the requirements

for the award of the degree

Of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND DESIGN

DEPARTMNET OF COMPUTER SCIENCE AND DESIGN

KONGU ENGINEERING COLLEGE

(Autonomous)

PERUNDURAI, ERODE – 638060

MAY2025

KONGU ENGINEERING COLLEGE

(Autonomous)

PERUNDURAI, ERODE – 638060

DEPARTMENTOFCOMPUTERSCIENCEANDDESIGN

BONAFIDE CERTIFICATE

Name : MADHAV S (23CDR083),
MANOHARANR(23CDR088)

CourseCode :22CDL41

CourseName :DATABASEMANAGEMENTSYSTEMLABORATORY

Semester IV

Certifiedthatthisisabonafiderecordofworkforapplicationprojectsdonebytheabove students for **22CDL41 – DATABASE MANAGEMENT SYSTEM LABORATORY** duringtheacademicyear**2024-2025.**

SubmittedfortheVivaVoceExaminationheldon_____

FacultyIn-Charge **HeadoftheDepartment**

TheatreTicketManagementSystem

- ABSTRACT
- INTRODUCTION
- SYSTEMREQUIREMENTSPECIFICATION
- DATAFLOWDIAGRAM
- ER-DIAGRAM
- IMPLEMENTATION
- SAMPLEINPUTANDOUTPUT
- CONCLUSION

ABSTRACT

The Theatre BookingManagement System(TBMS) is acomprehensive web-based applicationdeveloped to fully digitize and automate the operations of a theatre, with a particular focus on ticket and snackbooking. This project utilizes Structured Query Language (SQL) as the core for managing data storage and retrieval efficiently, ensuring smooth interaction between the frontend and backend systems. Designed to replace traditionalmanualbookingmethods,thesystemstreamlinesallkeyfunctionalitiessuchasmoviescheduling, seatallocation,customerregistration,ticketgeneration,andsnackordering.Theapplicationaimstodelivera seamless and error-free booking experience to users, while also reducing the operational load on theatre staff.At the backend, a MySQL database is used to manage structured information across multiple interconnected tables such as Theatre,show,seat,booking,snacks.These tables are linked logically to enable consistentandsecuredatahandling.Thesystemensuresthatallbookingsarevalidatedinreal-time,preventing issues like overbooking or duplicate seat allocation. In addition, advanced features such as booking history tracking,cancellationandmodificationofbookings,snackmanagement,andautomatedticketgenerationare integrated to provide a robust solution.The frontend of the TBMS is designed using HTML, CSS, and JavaScript, offering an interactive and responsive interface that works across devices. Users can browse movies, view available seats and snacks, make bookings, and manage their profiles with ease. The platform includes a user authentication system that protects user accounts and restricts access to administrative functions.Theadminpanelallowstheatremanagerstoaddorremoveshows,updatescreentimings,configure seating arrangements, manage snack inventory, and generate daily or weekly reports.This project is implementedandtestedontheXAMPPplatform,whichcombinesApacheserver,PHP,andMySQLdatabase providing a lightweight yet powerful development environment. The use of PHP for server-side scripting ensures smooth communication between theinterface and database, enabling real-time updates and secure transactions.TheTBMSnotonlyimprovestheaccuracy,speed,andconvenienceofticketandsnackbookings butalsoenablesscalabilityforfutureenhancements.Futureversionsofthissystemcanincludefeatureslike online payment gateways,multi-theatre support, and advanced analytics. Overall, this project serves as a practicalsolutiontothegrowingneedfordigitaltransformationintheentertainmentindustryandprovidesan efficient platform for both users and theatre administrators.

1. INTRODUCTION

OverviewOfTheProject

The Theatre Booking Management System is a web-based application developed to streamline and automate the movieticket booking process alongwith snackorderingservices. Thissystemisdesignedtoenhanceuserconveniencebyofferinganeasy-to-useinterfacewhere users can browse currently running movies, check available seats, book their tickets, and also ordersnacksinadvance.Theprojectincorporatesawell-structuredfrontendusingHTML,CSS, and JavaScript to provide a visuallyengaging and responsive user experience. The backend is developed using PHP, with data management handled through a MySQL database hosted on theXAMPPplatform,ensuringarobustandsecureenvironment.Registereduserscanmanage their profiles, view their booking history, and interact with the support team through a contact/queryform.Ontheadministrativeside,anadminpanelallowsauthorizedpersonnelto manage movie listings, seat arrangements, snack inventories, and monitor all user bookings. Thissystemnotonlyreducestheerrorsassociatedwithmanualticketingbutalsosavestimefor both users and theatre staff. It is scalable, secure, and can be further enhanced by integrating features like online payment, digital ticket generation, and mobile app support in the future..

AboutExistingSystem

MANUALSYSTEM:

The existing theatre ticketing system primarily operates on a manual basis, where staff members handle most operations such as ticket booking, seat allocation, and snack sales without the support of afullyintegrated digitalplatform. This manualsystemrelies heavilyon humaninvolvement, which makes it slow, prone to errors, and inefficient, especially during peak hours or when multiple customersarebeingservedsimultaneously.Customersarerequiredtophysicallyvisitthetheatreormake phonecallstoinquireaboutshowavailabilityandbooktickets,whichisnotonlyinconvenientbutalso lacks transparency regarding real-time seat availability. There is no centralized database to manage bookingrecords,makingitdifficulttotrackcustomerhistory, generatereports, ormanage cancellations and modifications effectively. Additionally, snack ordering is usually done separately at counters, resultinginlongqueuesandwastedtimeforbothcustomersandstaff.Overall,themanualsystemcannot meet the demands of modern theatre operations where speed, accuracy, and customer convenience are essential.

DrawbacksOfTheExistingSystem

- Manualbookingandseatallocation
- Noreal-time updates
- Overbookingerrors
- Time-consuminganderror-prone
- Nosnack bookingintegration

2. SystemRequirementSpecification

HardwareSpecification

Processor	:DualCoreorhigher
RAM	:2 GB+
HardDisk	:512GBor more
Floppydisk	:NotRequired*(modernsystemsuseUSBorcloudstorage)*
KeyBoard	:StandardQWERTYKeyboard
Monitor	:15”oraboveLEDMonitor(HDrecommended)

Software Specification

Backend	:XAMPP(Apache+MySQL+PHP)
Frontend	: HTML,CSS,JS
Browser	: Webbrowser

3. DataFlowDiagram

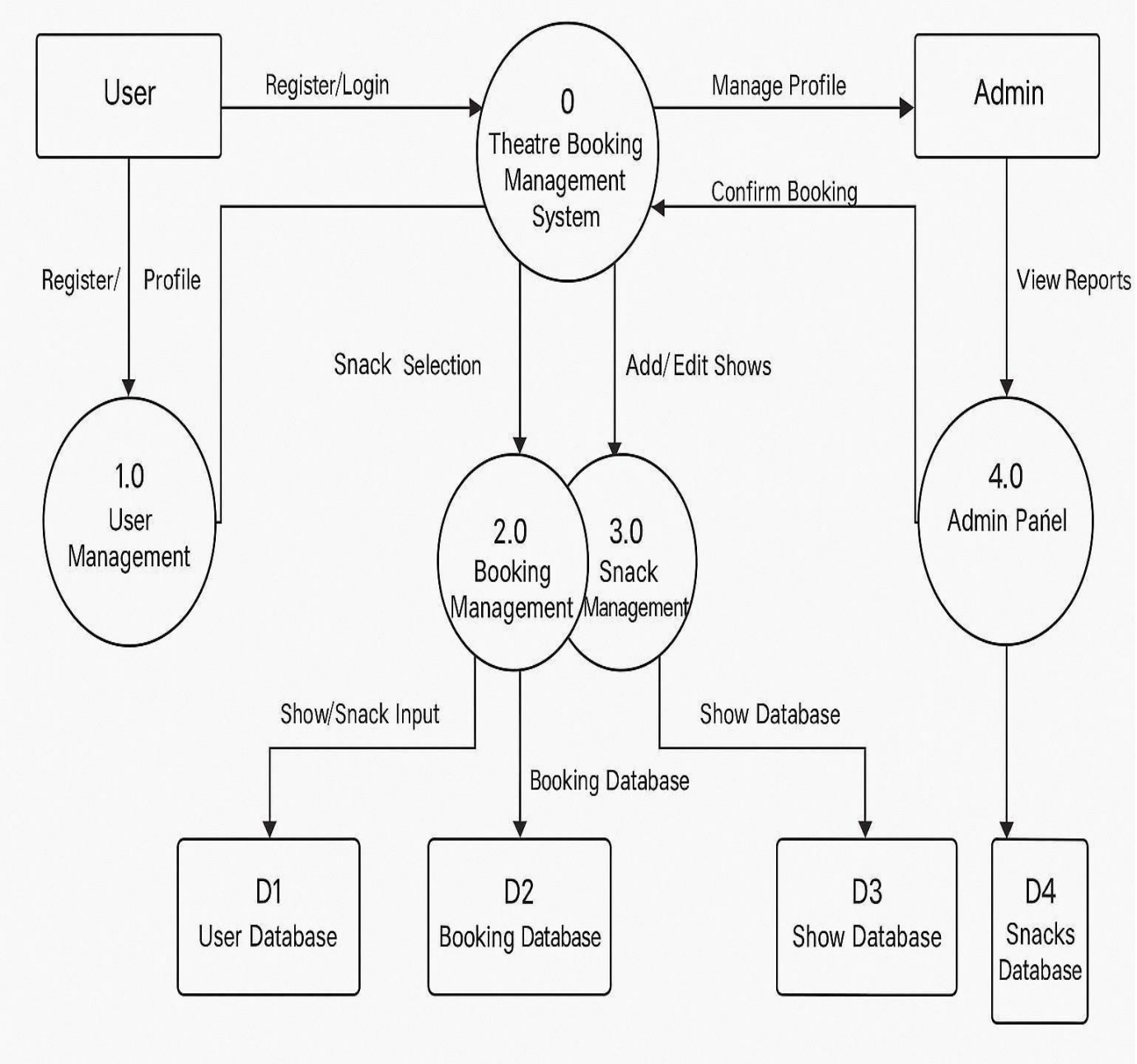

Fig3.1 Dataflowdiagram

4. ER-Diagram

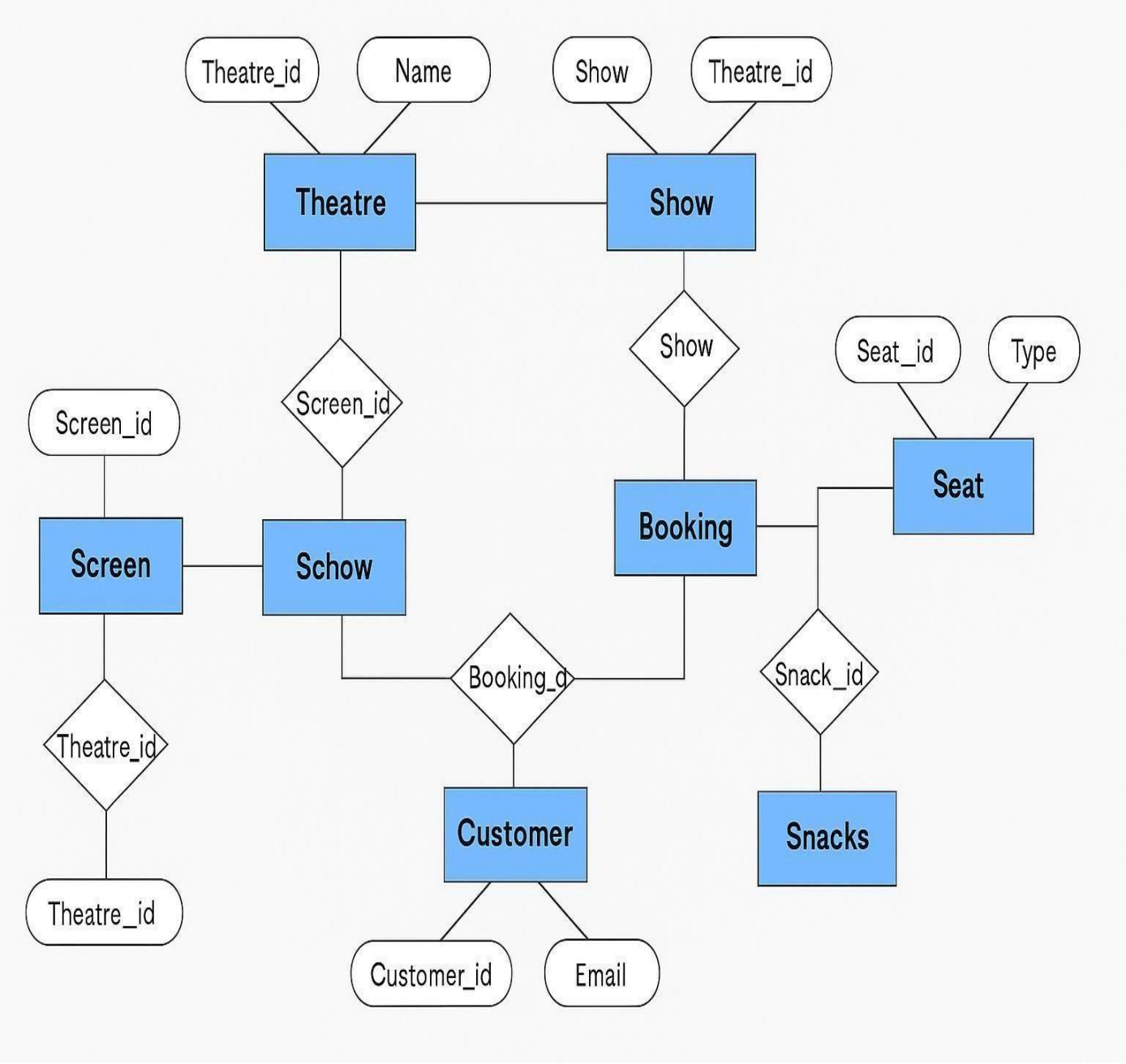

Fig4.1E-RDiagramforTheatreTicketManagement

5. Implementation

The Theater Booking Management System is a comprehensive web-based application developed with a combination of modern web technologies to simplify and automate the process of movie ticket and snack booking.ThebackendispoweredbyPHP,awidelyusedserver-sidescriptinglanguage,whichfacilitatesthe core business logic of the application. Data storage and retrieval are managed using MySQL, a robust relationaldatabasesystemidealforhandlingstructureddataandensuringdataintegrity.Theentireapplication ishostedandtestedlocallyusingXAMPP,across-platformdevelopmentenvironmentthatincludesApache, PHP,andMySQL,makingitconvenientfordevelopmentanddeployment.ThefrontendisbuiltusingHTML for structuring the web pages, CSS for styling and layout, and JavaScript for adding interactivity and client-sidevalidation.Together,thesetechnologiescreatearesponsiveanduser-friendlyinterfacethatenhancesthe overall user experience. The system starts by designing andimplementing a well-structured relational database containing essential tables such as Users, Seats ,Shows, Booking, Snacks, Snack_orderandContact_message.Eachtableiscarefullydesignedtohandlespecificfunctionalities.suchas managing user profiles, storing show timings and seat availability, processing bookings, handling snack orders, and recording user queries or feedback.PHP scripts are used to connect the frontend to the backend, enabling users to register and log in securely, view currently available movies, check real-time seat availability,selectseats,andoptionallyaddsnackstotheirbookings.Thesnackbookingfeatureallowsusers topre-orderitemslikepopcorn,drinks,nachos,orcombopacks,whichareaddedtotheirfinalbill.JavaScript plays a vital role in validating input fields, dynamically updating seat selections, and providing instant feedbacktousersduringinteractions.Ontheadministrativeside,asecureadminpanelallowsauthorizedstaff to manage movie listings, update seat maps, adjust snack availability and pricing, and review user queries submitted through the contact form. This ensures that the management has full control over the system's operations. The modular design of the system not only promotes easier maintenance and scalability but also allows for the integration of advanced features in the future, such as online payment gateways, mobile app connectivity, and digital ticket scanning. Rigorous testing is conducted to verify that each module performs as expected, with special attention to error handling, data validation, and user flow. Overall, the system deliversaseamlessbookingexperienceforuserswhilestreamliningthemanagementprocessfortheaterstaff.

SampleTable:

user_id(int)-PrimaryKey	UniqueIDforeachuser(auto-incremented).
user_name(varchar)	Fullnameoftheuser.
user_email(varchar)	Emailaddressoftheuser(mustbeunique).
user_password(varchar)	Hashedpasswordforloginsecurity.
user_create_at(timestamp)	Timestampwhentheuseraccountwascreated.

TableName: Users

PrimaryKey:user_id

Description: AddUsersdetail

profile_id(int)-PrimaryKey	Uniqueidentifierforeachprofile
profile_name(varchar)	Nameoftheuser
profile_email(varchar)	EmailIDassociatedwiththeprofile
profile_phone(varchar)	Phonenumberoftheuser
profile_created_at(TIMESTAMP)	Dateandtimewhentheprofilewascreated
profile_picture(varchar)	PathorURLtotheuser'sprofileimage

Table Name:profiles

PrimaryKey:profile_id

Description: Usersprofilesdetails

contact_messages(int)-PrimaryKey	UniqueIDforeachcontactmessage
contact_messages_name(varchar)	Nameofthepersonsubmittingthemessage
contact_messages_email(varchar)	Emailaddressofthesender
contact_messages_message(TEXT)	Messageorquerycontentsubmitted bytheuser
contact_messages_submitted_at(TIMESTAMP)	Timestampwhenthemessagewassubmitted

Table Name: contact_messages

Primary Key: contact_messages_id

Description:contact_messagesdetails

snack_orders_id(int)–PrimaryKey	UniqueID foreachsnackorder
popcorn(int)	Quantityofpopcornordered(defaultis0)
drink(int)	Quantityofdrinksordered(defaultis0)
nachos(int)	Quantityofnachosordered(defaultis 0)
combo(int)	Quantityofcombopacksordered(defaultis0)
total(int)	Totalpriceofthesnackorder
order_time(TIMESTAMP)	Timestampwhenthesnackorderwas placed

TableName : Snack

Primary Key: Snack_orders_id(int)

Description : Snakesorderingdetails

booking_id(int)PrimaryKey	Unique bookingID(auto-incremented)
user_id(int)	IDoftheuserwhomadethebooking(foreignkeytousers.id)
show_id(int)	IDoftheselectedmovieshow(foreignkeytoshows.id)
seat_numbers(varchar)	Commaseparatedlist ofseatnumbersbooked
snack_order_id(int)	IDoftheassociatedsnack order(foreignkeytosnack_orders.id)
total_amount(int)	Totalcostincludingticketandsnacks
booking_time(TIMESTAMP)	Dateandtimewhenthebookingwasmade

TableName :booking

Primary Key: snack_orders_id(int**)**

Description : Snakesorderingdetails

6. SAMPLEINPUTANDOUTPUT

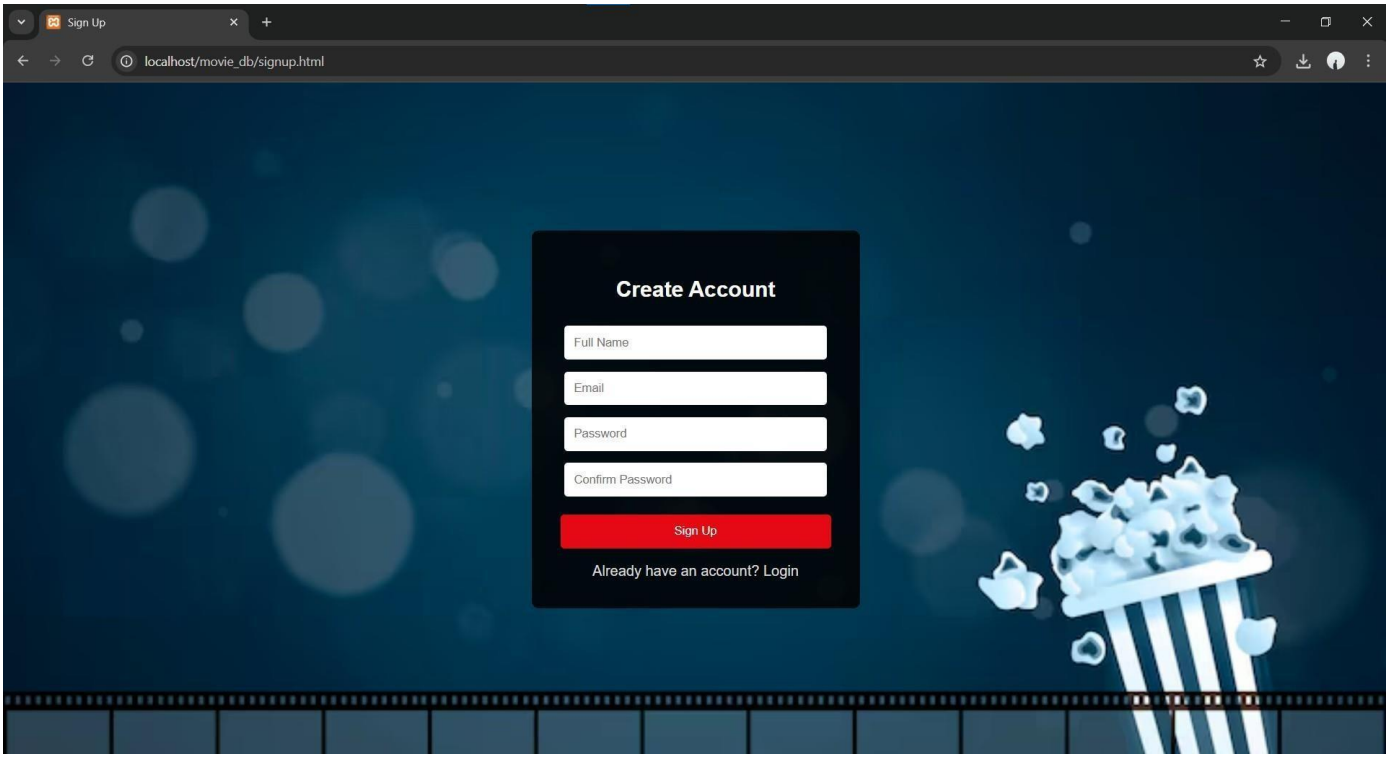

Fig6.1signuppagefortheatreticketbookingapp

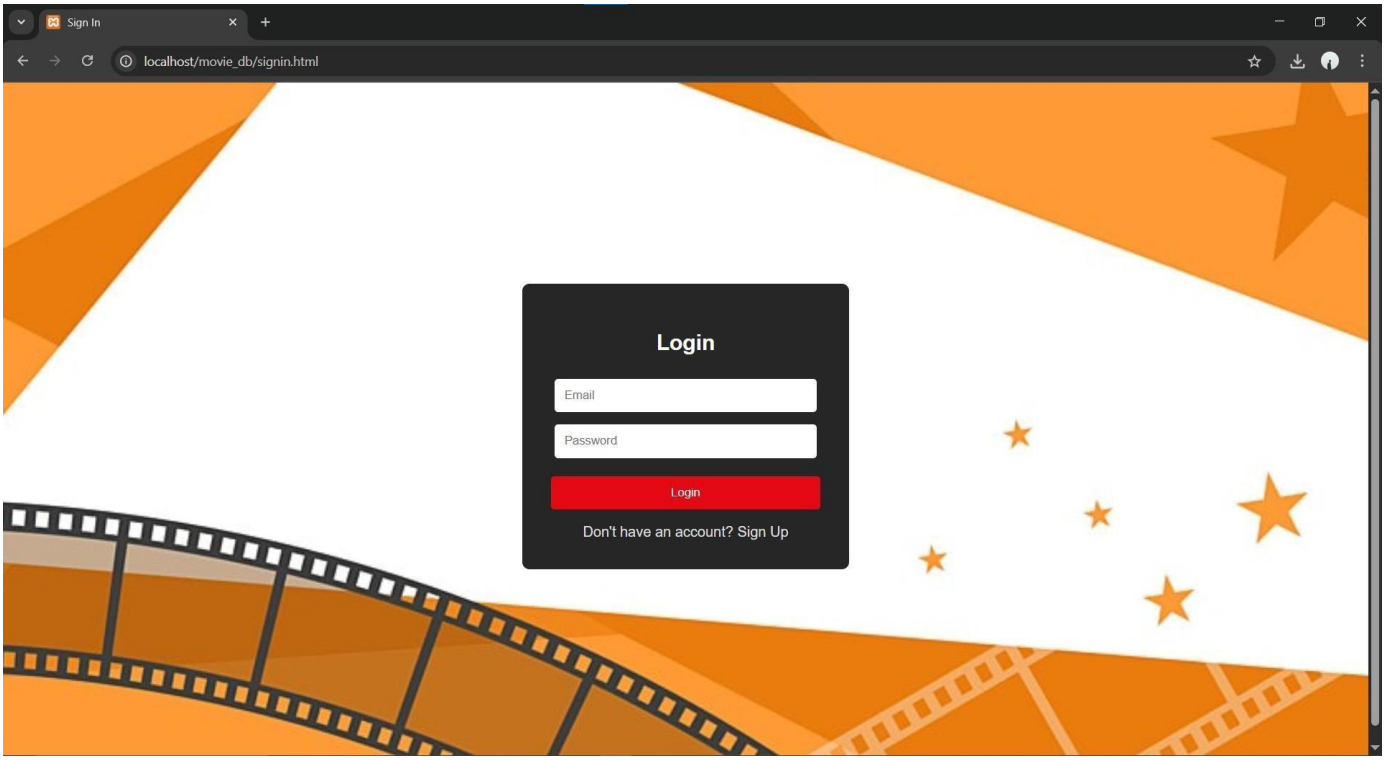

Fig6.2 Loginpage fortheatreticketbookingapp

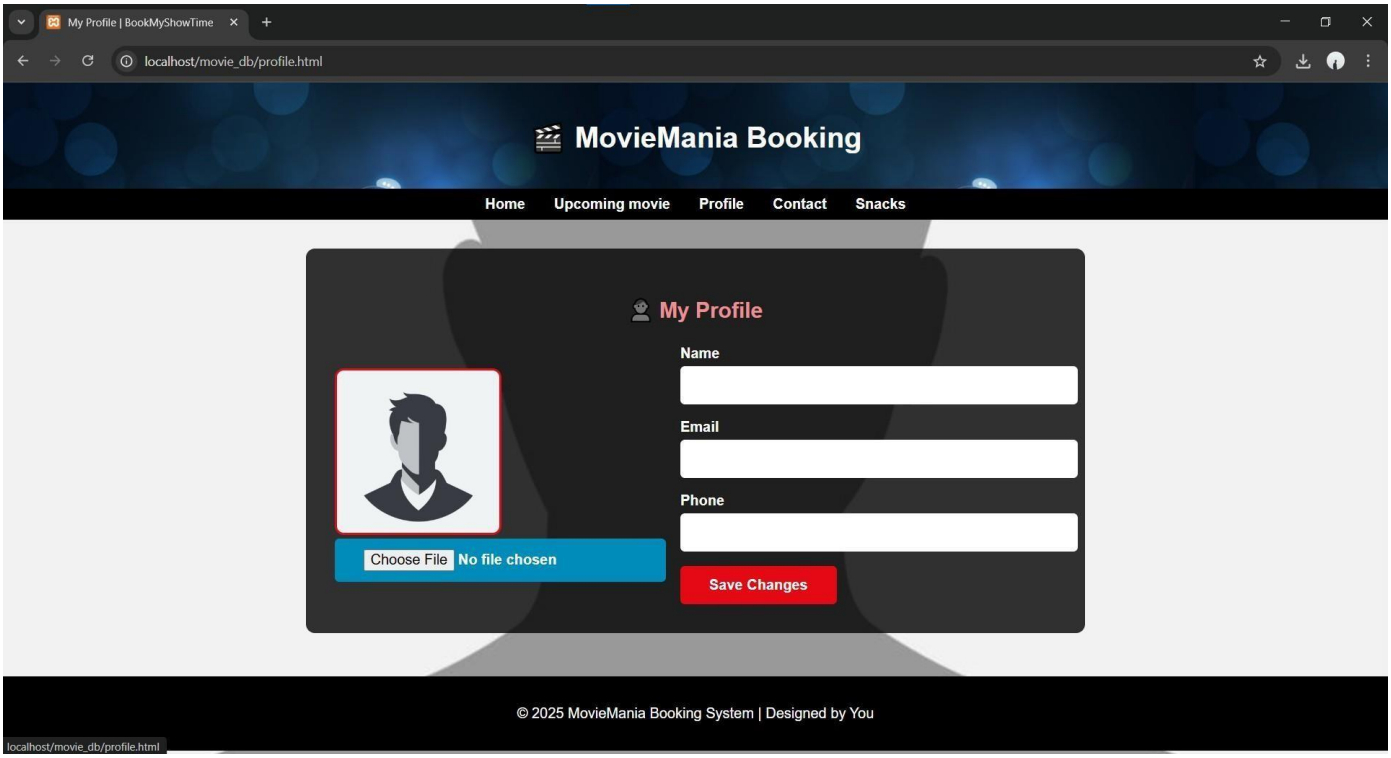

Fig6.3 Profilepage fortheatreticketbookingapp

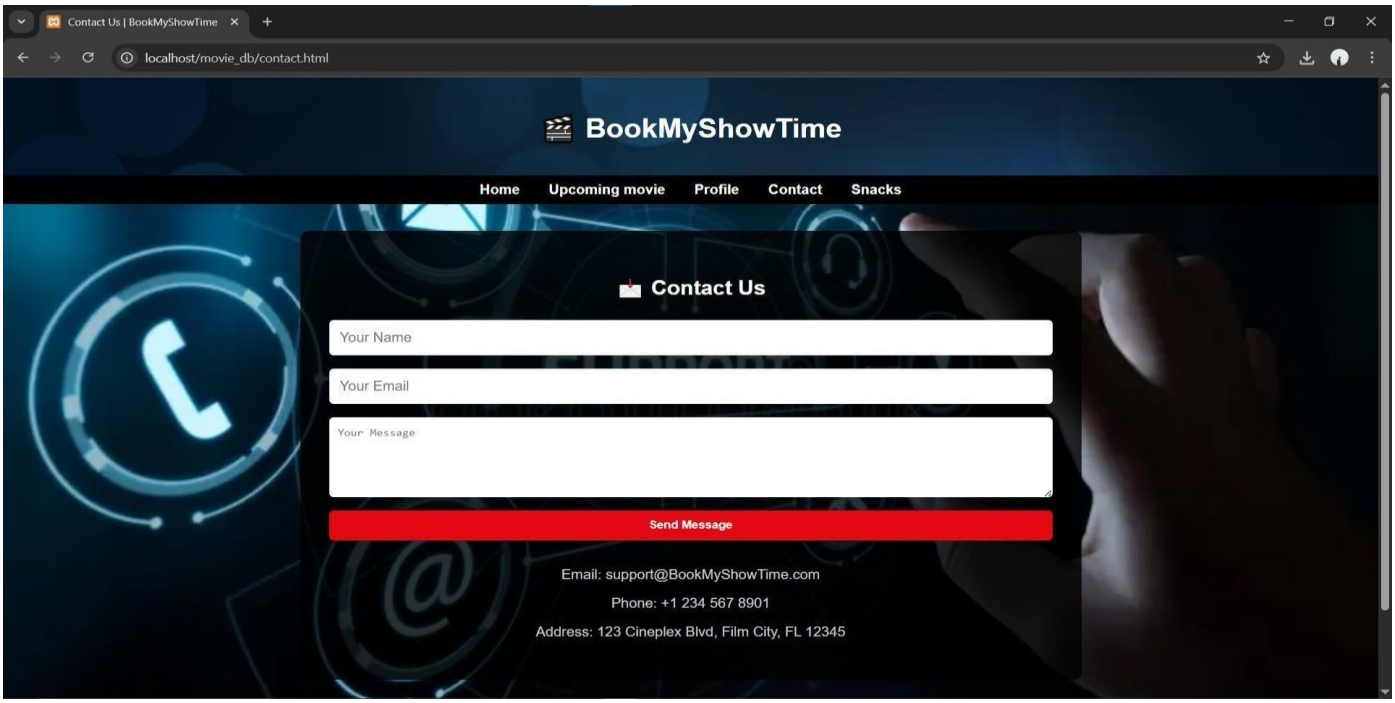

Fig6.4 Contactuspagefortheatreticketbookingapp

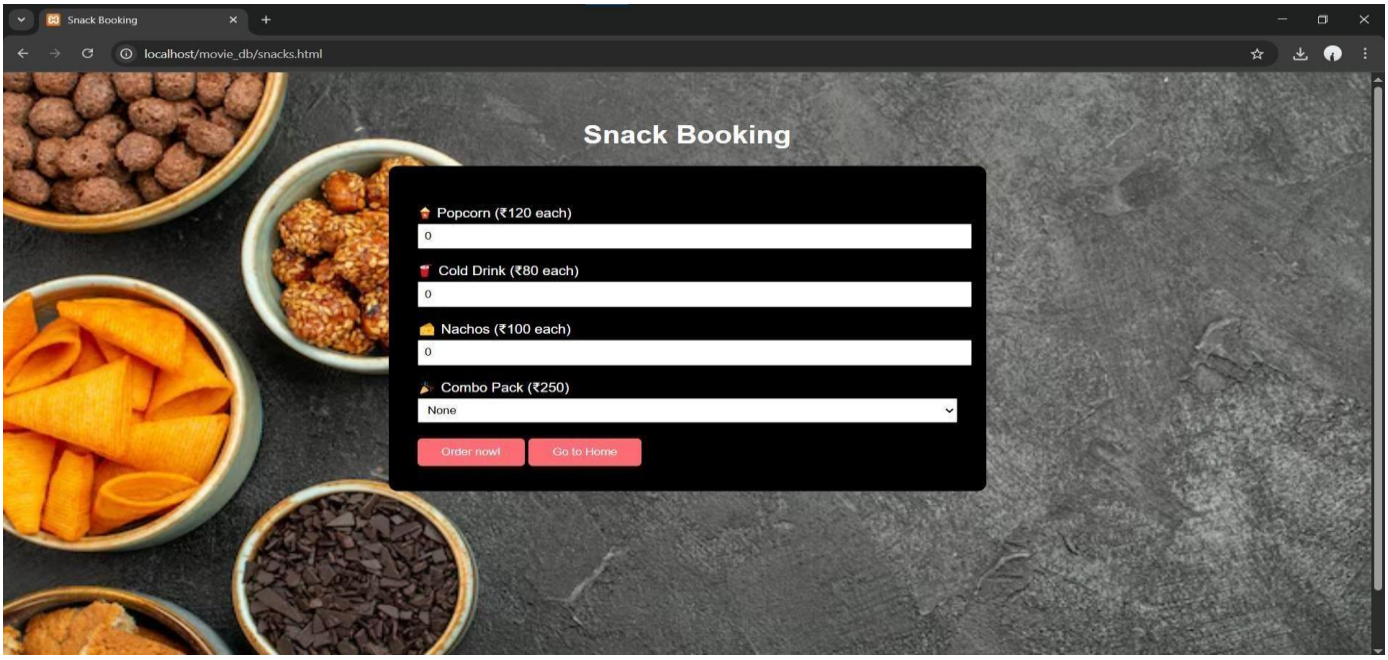

Fig6.5 Snacksbookingpagefortheatreticketbookingapp

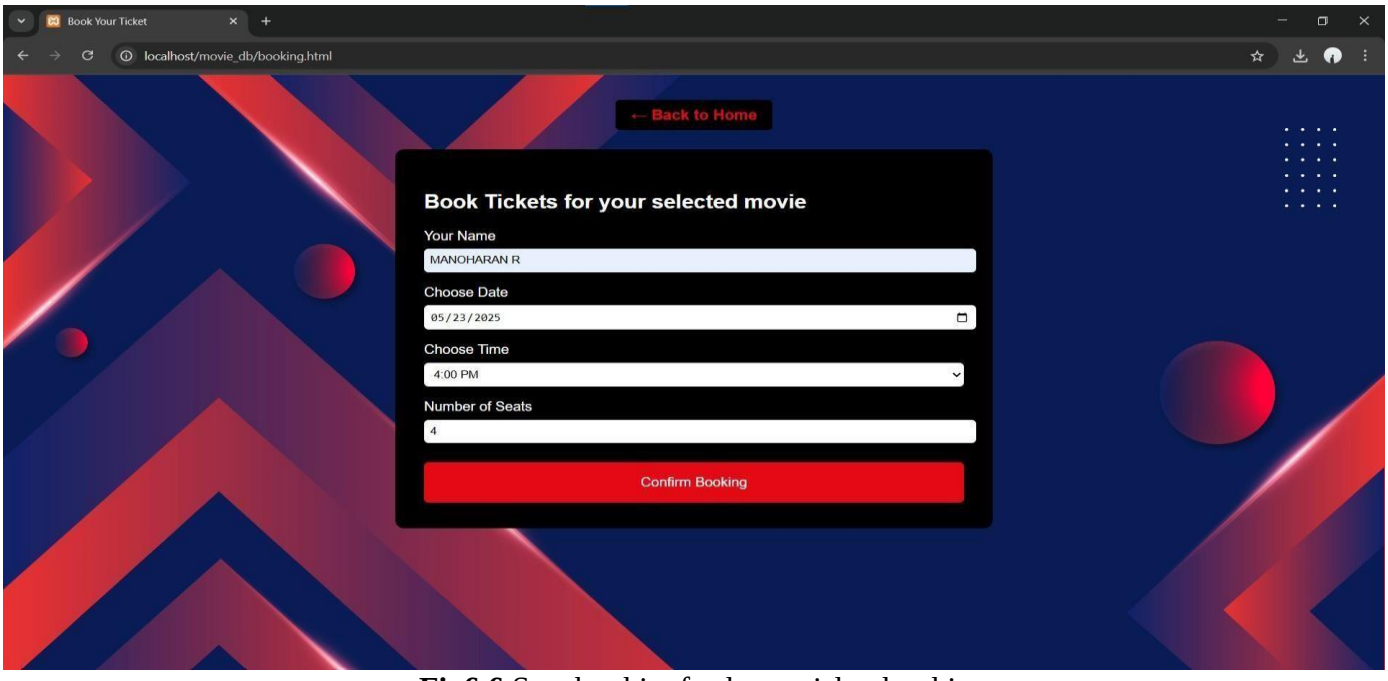

Fig6.6 Seat bookingfortheatreticket booking app

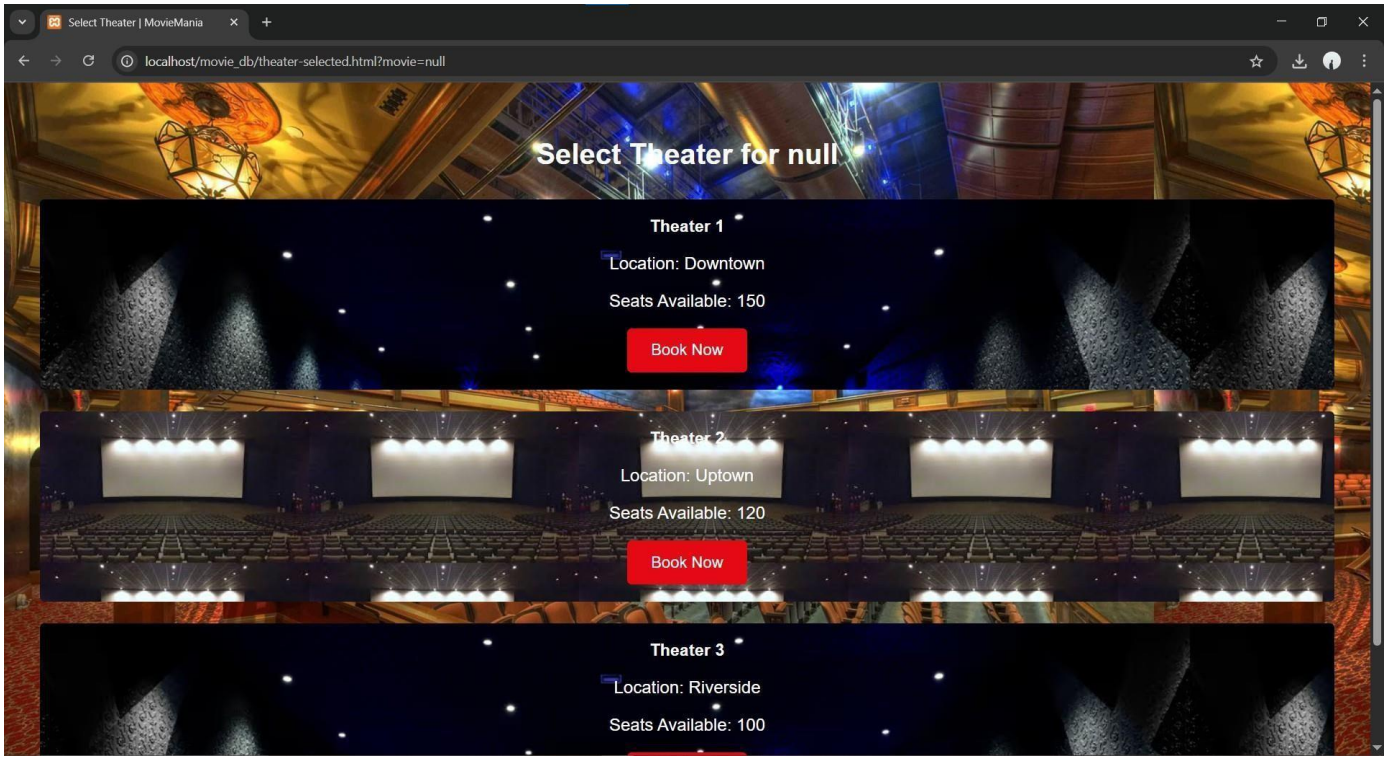

Fig6.7 Theatreticketbooking page

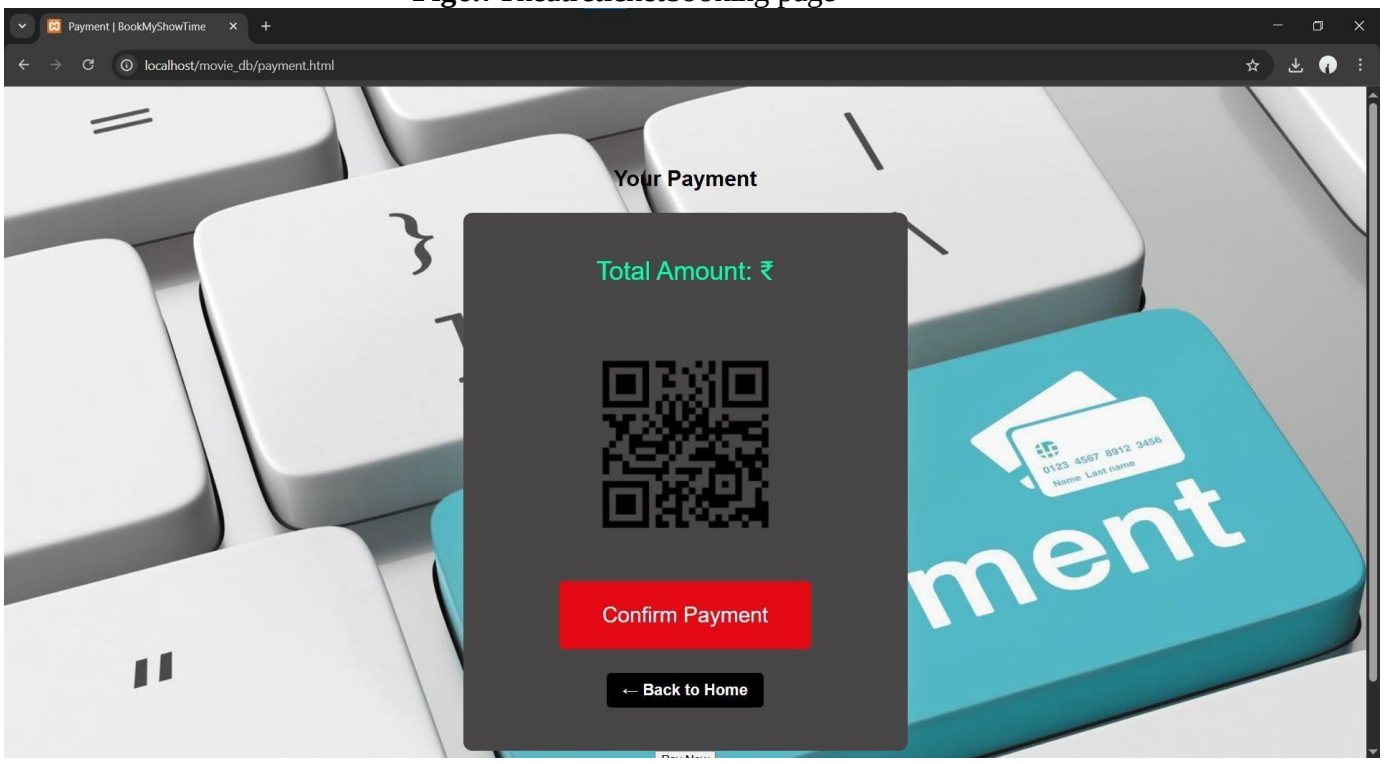

Fig6.8 Paymentpagefortheatreticketbooking app

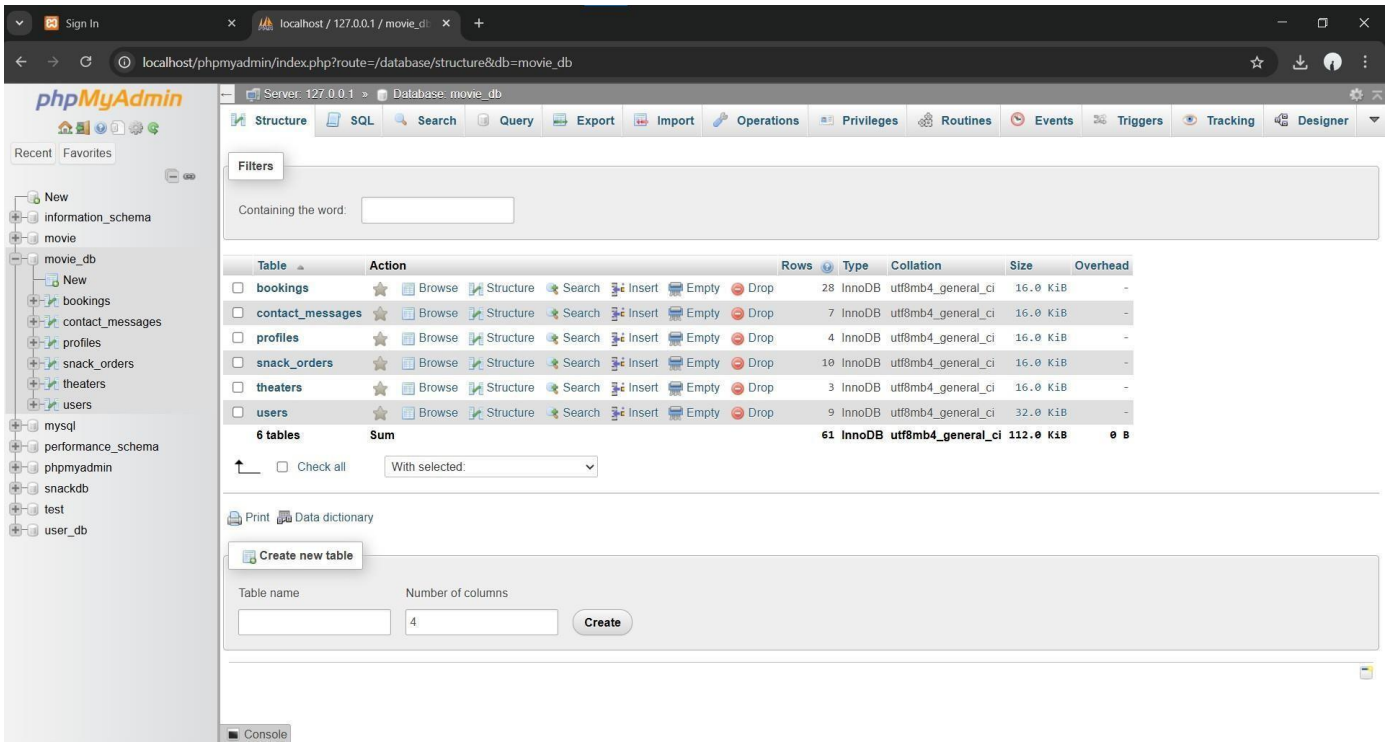

Fig6.9 Phpmyadminpagefortheatreticketbookingapp

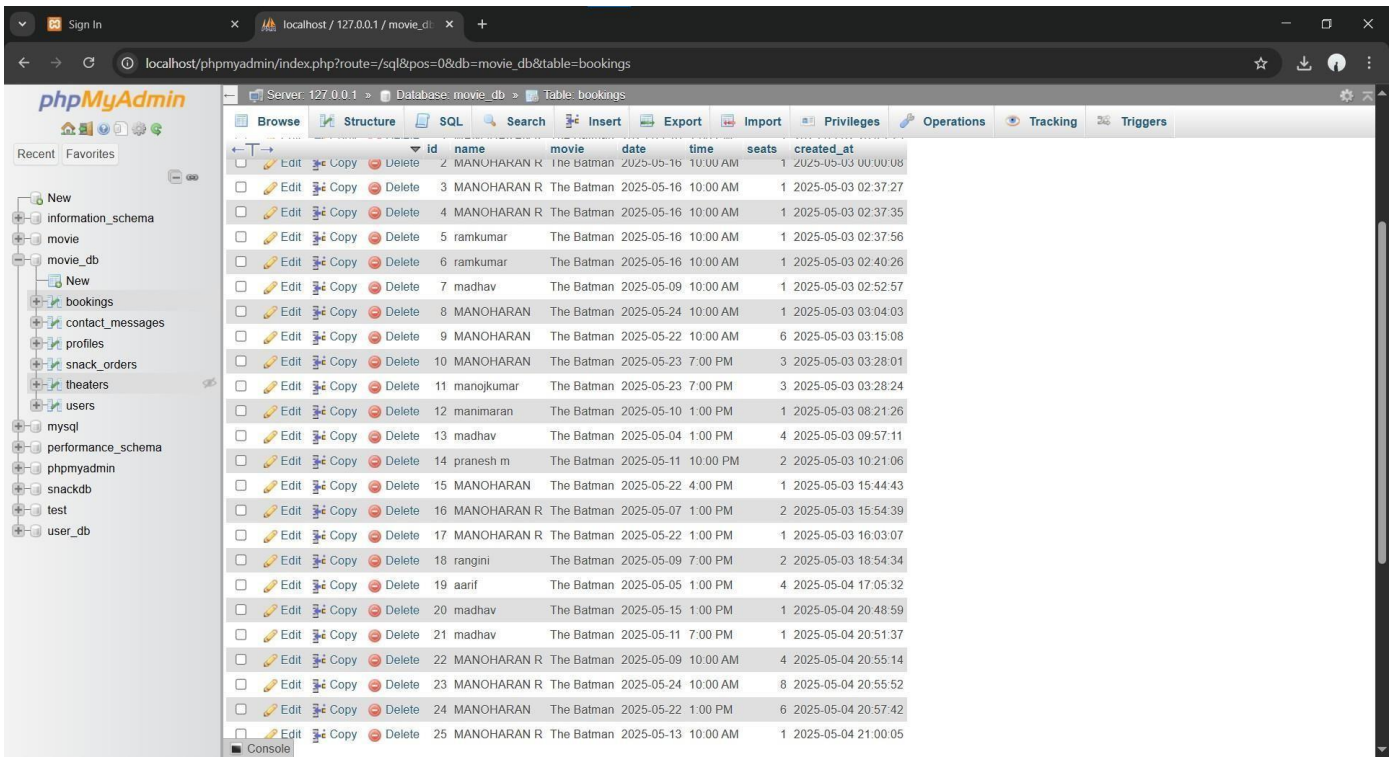

Fig6.10 Databasefortheatreticketbooking app

7. Conclusion

The Theater Booking Management System successfully fulfills the essential needs ofa modern movie ticket reservation platform. It has been developed and tested using sample data,ensuringoptimalperformance and usability. The system offers a seamless interface for users to browse movies, select showtimes, book seats, and place snack orders. It also provides administrative functionality for managing shows, monitoring snack inventory, and respondingto customerqueriesvia acontact form.Withitscleaninterface,real-timeupdates, andreliablebackendintegrationusingPHPandMySQL,thesystemsignificantlyreducesthe manual effort involved in ticket booking and snack sales. It eliminates double bookings, ensures better record-keeping, and enhances customer convenience. By digitizing and automatingthe entire process, the platformnot onlysaves time but also enhances the overall theater experience for users and streamlines operations for staff.

FutureEnhancements:

- **OnlinePaymentGatewayIntegration**
  Allowuserstopayforticketsandsnacksonlineusingdebit/creditcards,UPI,or wallets for added convenience.
- **QRCode–BasedTicketing**
  GenerateQRcodesforeachbookingtoenablecontactlessentryand efficient ticket validation at the theater gate.
- **MobileAppDevelopment**
  DevelopAndroidandiOSmobileapplicationsforuserstobookticketsand snacks anytime, anywhere.
- **SeatLayout Visualization**
  Offeradynamicvisualrepresentationoftheseatinglayouttoenhancethe seat selection experience.
- **CustomerFeedbackandRatingSystem**
  Letusersratemoviesandprovidefeedbackontheirexperiencetohelp theaters improve service quality.
- **CloudHostingandScalability**

Deploytheapplicationoncloudserverstohandlehightrafficduring blockbuster releases and ensure 24/7 availability.

- **LoyaltyProgramIntegration**

  Rewardfrequentuserswithdiscountsorfreesnacks,enhancingcustomerretention.

References

1. PHPGroup.(2023).*PHPManual*.Retrievedfromhttps://www.php.net
2. MySQL.(2023).*MySQLDocumentation*.Retrievedfromhttps://www.mysql.com
3. W3Schools.(2023).*WebDevelopmentTutorials*.Retrievedfromhttps://www.w3schools.com
4. ApacheFriends.(2023).*XAMPPDocumentation*.Retrievedfromhttps://www.apachefriends.org
5. PythonSoftwareFoundation.(2023).*PythonDocumentation*.Retrievedfrom https://www.python.org
6. MySQL.(2023).*MySQLConnector/PythonDeveloperGuide*.Retrievedfrom https://dev.mysql.com/doc/connector-python/en/
7. GitHub.(2023).*GitHubDocumentation*.Retrieved fromhttps://docs.github.com