MOVIE BOOKING APP

IIHT

CONTENTS

1 PROBLEM STATEMENT

Movie Booking App is an application that allows the users to register and login and search movies released. User can book the tickets for the movies they wish for. Admin shall view the tickets booked and update the pending tickets available to the system.

Guest users cannot book any tickets.

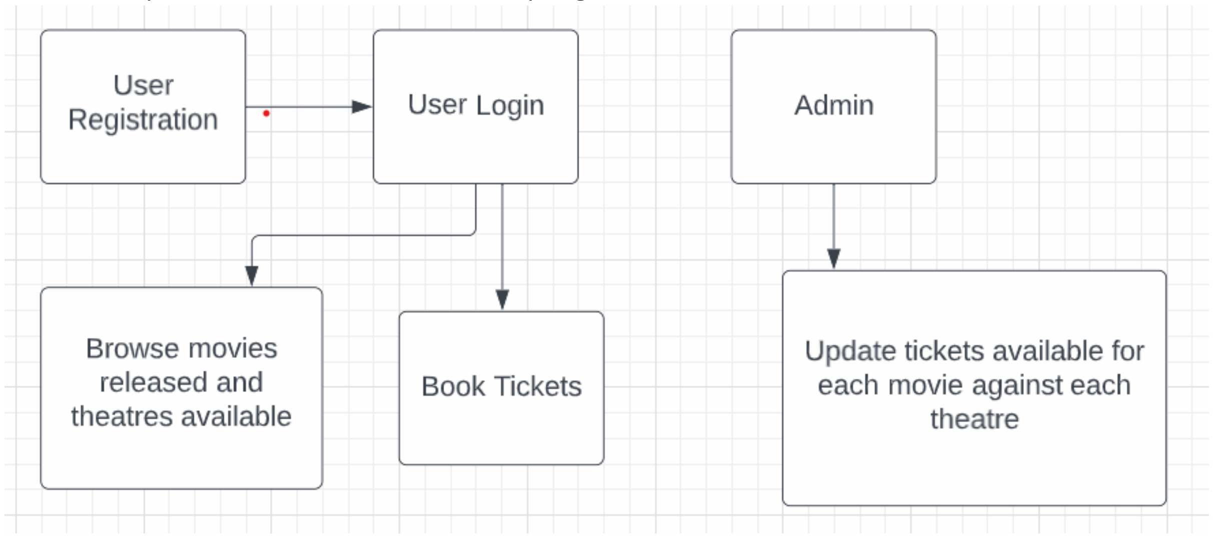
The core modules of movie booking app are:

1. User registration and login
2. Browse the movies released and list of theatres
3. User can book the tickets
4. Admin to update the available ticket status based on the tickets booked

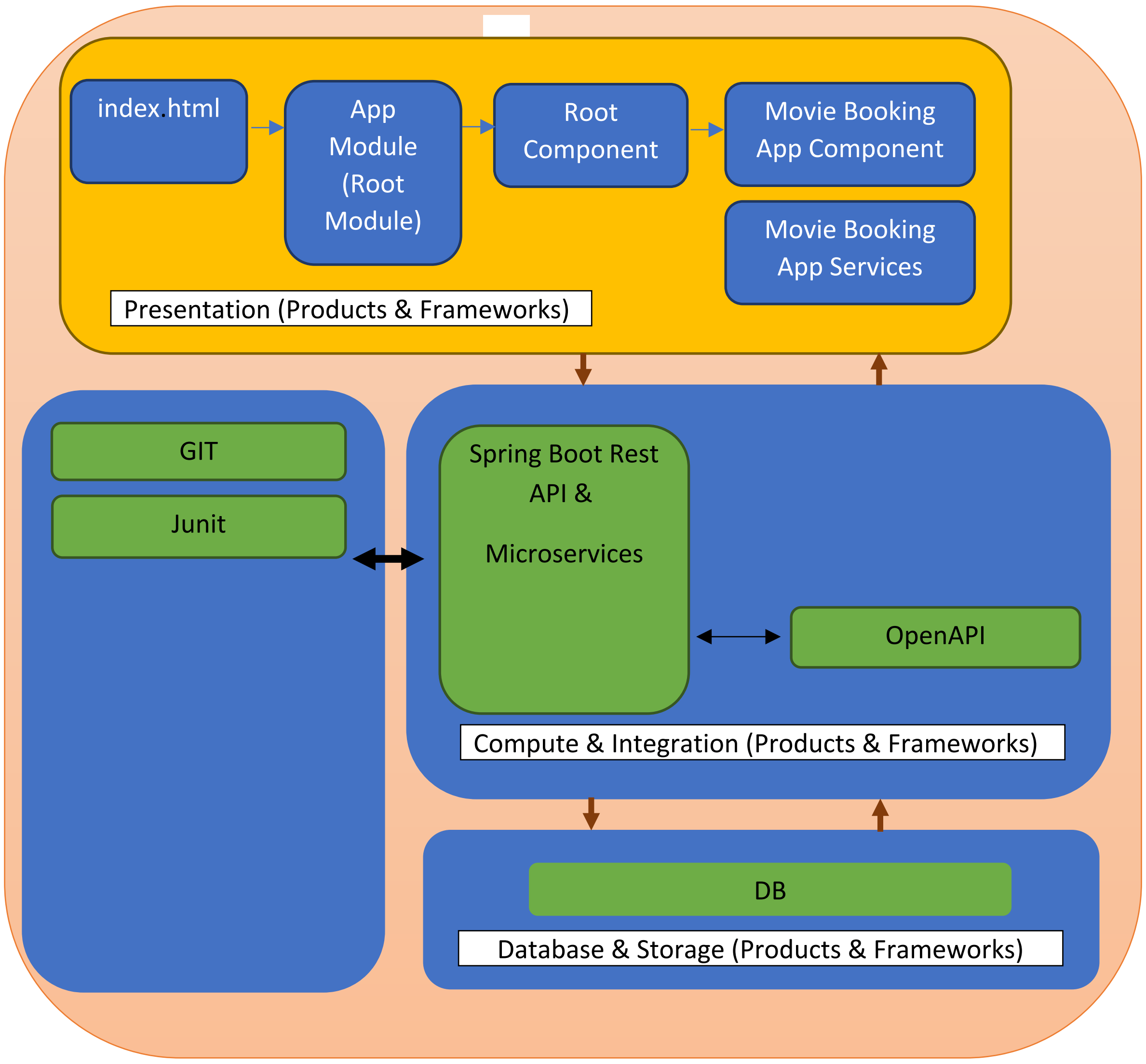
The scope includes developing the application using toolchain mentioned below.

2 PROPOSED MOVIE BOOKING APP WIREFRAME

1. UI needs improvisation and modification as per given use case.


User Registration
User Login
Admin
Browse movies released and theatres available
Book Tickets
Update tickets available for each movie against each theatre

3 APPLICATION ARCHITECTURE


index.html
App Module (Root Module)
Root Component
Movie Booking App Component
Movie Booking App Services
Presentation (Products & Frameworks)
GIT
Junit
Spring Boot Rest API & Microservices
OpenAPI
Compute & Integration (Products & Frameworks)
DB
Database & Storage (Products & Frameworks)

4 TOOL CHAIN

Competency	Skill	Skill Detail
Engineering Mindset	Networking and Content Delivery	
	Ways of Working	
	Consulting Mindset	
	DevOps	
Programming Languages	Application Language	Java
Products & Frameworks	Presentation	Angular/React
		Karma & Jasmine
		Jest & Cypress
	Compute & Integration	Spring Boot
	Database & Storage	MySQL/SQL-Server
	Governance & Tooling	Git
		Maven
		JUnit 5/Mockito

5 DEVELOPMENT FLOW

No	MC	Competency	Section	Indicative Mechanism for Evaluation (Passing score of 60% in each MC)
Business Requirement				
1	**Backend**	Rest API, Database, Testing, Debugging & Troubleshooting	Click here	**Code Submission and Evaluation, Panel Presentation**
2	**Front End**	Angular/React	Click here	**Code Submission and Evaluation, Panel Presentation**

6 BUSINESS-REQUIREMENT:

As an application developer, develop frontend, middleware and deploy the Movie Booking App with below guidelines:

Pre-requisite before implementing the user story:

Create a static database as "Movie" and add the below fields

- Movie name, Total number of tickets allotted and theatre name (Movie name and theatre name should be of composite primary key)

There should be a predefined set of tickets assigned to each theatre against each movie. Just create sample 2 movies with 2 theatres assigned. This will be used for the upcoming user story to fetch the data.

User Story #	User Story Name	User Story
US_01	Registration and Login	As a user I should be able to login/Register in the movie booking application Acceptance criteria: 1. A logged-in user can reset their password so they can login, even if they forget their password. 2. A logged-in user: a. Cannot change their username. b. Can logout from their account. 3. As a customer I should be able to furnish following details at the time of registration a. First Name b. Last Name c. Email d. Login Id e. Password f. Confirm Password g. Contact Number 4. All details fields must be mandatory 5. Login Id and Email must be unique 6. Password and Confirm Password must be same 7. If any constraint is not satisfied, validation message must be shown
US_02	View &Search Movies	As a user I should be able to view all the recent movies opened for booking. User can search any particular movies as well Acceptance criteria: a. User can view all the existing released movies. b. User can search any particular movie based on the movie name
US_03	Book Tickets for a movie	As a user I should be able to book tickets for a movie. Add this booking to a database table “Tickets”. Assign movie name and theatre name as a foreign key to database table “Movie” which is already created as a pre-requisite Acceptance criteria: a. Book a movie ticket b. Below are the details to be added • Movie Name • Theatre name • Number of tickets • Seat Number
US_04	View booked tickets and Update ticket status	As an admin I should be able to view booked tickets and update the balance tickets available for a movie

		Acceptance criteria: a. View number of booked tickets for a particular movie from table “Tickets” b. Check the tickets available from table “Movie” by calculating the total tickets booked c. If the quantity is 0, update the ticket status as ‘SOLD OUT’, else update as ‘BOOK ASAP’

7 RUBRICS/EXPECTED DELIVERABLES

7.1 REST API (PRODUCTS & FRAMEWORKS -> COMPUTE & INTEGRATION):

a. Use Spring Boot to version and implement the REST endpoints.

b. Implement HTTP methods like GET, POST, PUT, DELETE, PATCH to implement RESTful resources:

POST	**/api/v1.0/moviebooking/register**	**Register as new user**
GET	**/api/v1.0/ moviebooking /login**	**Login**
GET	**/api/v1.0/ moviebooking /<username>/forgot**	**Forgot password**
GET	**/api/v1.0/ moviebooking /all**	**View all movies**
GET	**/api/v/1.0/moviebooking/movies/search/moviename ***	**Search by movie name**
POST	**/api/v1.0/ moviebooking /<moviename>/add**	**Book tickets for a movie**
PUT	**/api/v1.0/moviebooking /<moviename>/update/<ticket>**	**Update ticket status**
DELETE	**/api/v1.0/ moviebooking /<moviename>/delete/<id>**	**Delete movie**

c. ***Movie name may be partial or complete username**

d. Use necessary configuration in place for REST API in application. Properties or bootstrap. Properties or application.yml; whichever is applicable.

e. Package Structure for Spring Boot Project will be like com. moviebookingapp. * With proper naming conventions for package and beans.

f. Use configuration class annotated with @Configuration and @Service for business layer.

g. Use constructor-based dependency injection in few classes and setter-based dependency injection in few classes.

h. Follow Spring Bean Naming Conventions

7.2 DATABASE (PRODUCTS & FRAMEWORKS -> DATABASE & STORAGE):

1. As an application developer:
   a. Implement ORM with Spring Data MongoRepositoryand MongoDB / JPARepository for other databases like SQL server/ MySQL. For complex and custom queries, create custom methods and use @Query, Aggregations (Aggregation Operation, Match

Operation, Aggregation Results), implementation of MongoTemplateetc as necessary.

b. Have the necessary configuration in place for REST API in application.properties or bootstrap.properties or application.yml OR Java based configuration; whichever is applicable.

c. Implement SQL procedures, triggers, and other performance-enhancing techniques.

d. Implement at least 2 role-based authentications/authorisations in your Spring Boot applications.

e. Implement cache and session management techniques using the Spring Boot API.

7.3 MAVEN (TOOLING):

1. As an application developer:
   a. Create the spring boot project using Maven CLI
   b. Generate Surefire test reports and share it as a part of deliverables
   c. Using Maven CLI generate the project documentation, and share it as a part of the deliverables.

7.4 UNIT TESTING:

1. As an application developer:

   a. Implement Unit testing using JUnit 5 for all the layers Exp: Controller layer, Service layer, and Repository layer.
   b. Generate the test results with positive and negative test scripts.

7.5 DEBUGGING & TROUBLESHOOTING

1. Generate bug reports & error logs - Report must be linked with final deliverables, which should also suggest the resolution for the encountered bugs and errors.

7.6 CODE QUALITY & CODE COVERAGE

1. Ensure code quality with static analysis tools
2. Code coverage for API should be > 80%

7.7 GOOD TO HAVE:

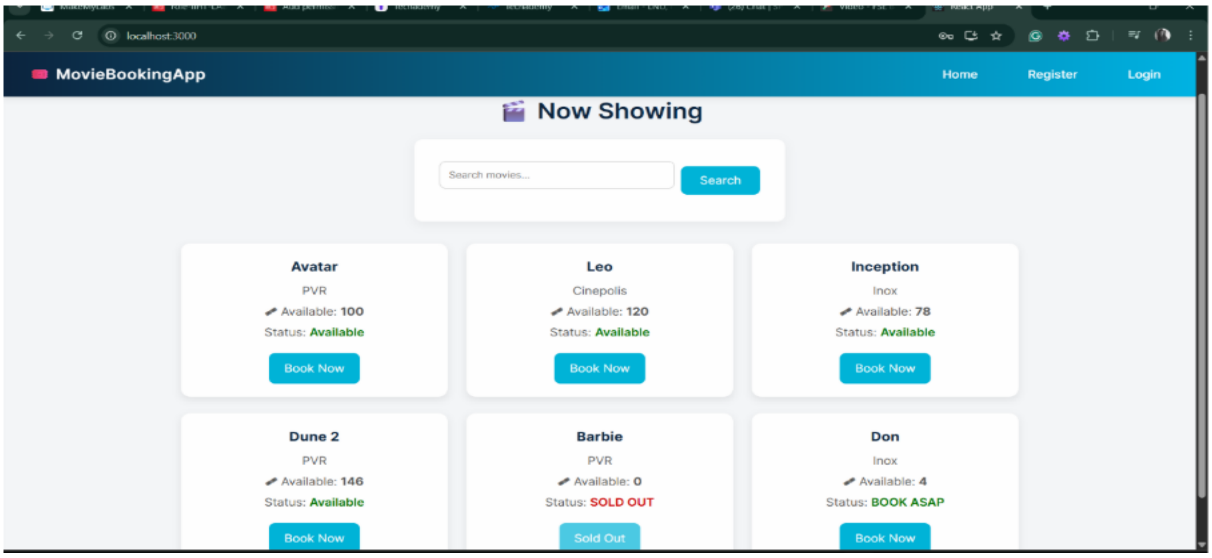
1. Implement error/exception handling mechanisms
2. Implement logging and monitoring to detect and troubleshoot issues
3. Implement rate limiting to prevent abuse of API endpoints
4. Blogs should be fetched and displayed within 30 seconds
5. Use containerization to package and deploy the application
6. Improve maintainability with modular codebase, automated testing, CI/CD pipelines

8 FRONTEND

1. Develop the front end for all user stories.
2. Implement using either Angular or React
3. Implement all the Front-End validation rules
4. Proper naming conventions and folder structures
5. Implement using proper SOLID design principles
6. Perform unit and integration testing for the front-end application

9 EXPECTED OUTCOMES:

9.1.1 MovieBookingApp Home Page



localhost:3000

MovieBookingApp

Home Register Login

Now Showing

Search movies... Search

Avatar
PVR
Available: 100
Status: Available
Book Now

Leo
Cinepolis
Available: 120
Status: Available
Book Now

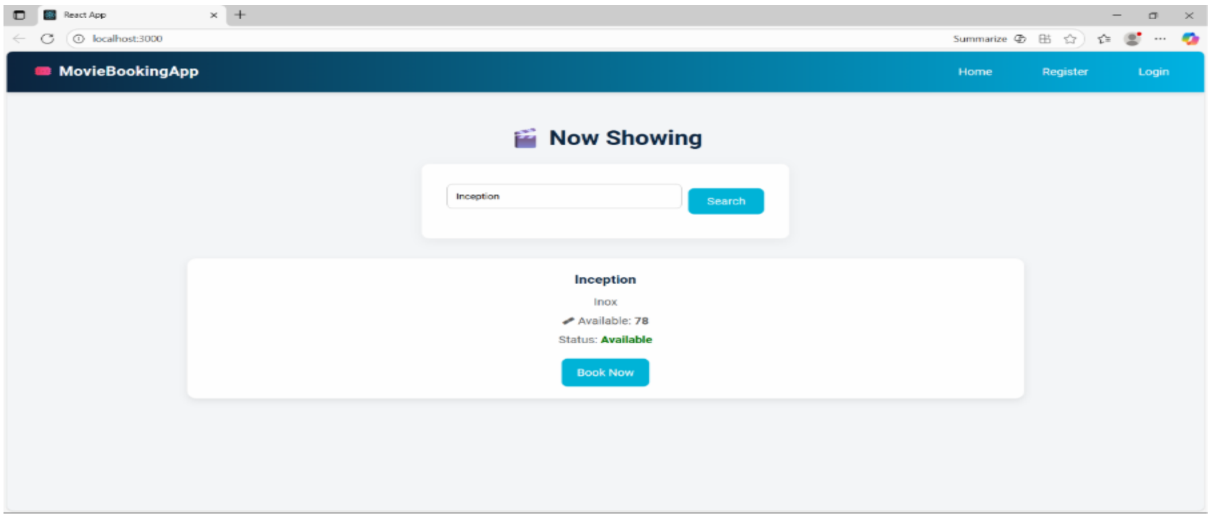
Inception
Inox
Available: 78
Status: Available
Book Now

Dune 2
PVR
Available: 146
Status: Available
Book Now

Barbie
PVR
Available: 0
Status: SOLD OUT
Sold Out

Don
Inox
Available: 4
Status: BOOK ASAP
Book Now

9.1.2 Search By Movie Name



React App

localhost:3000

MovieBookingApp

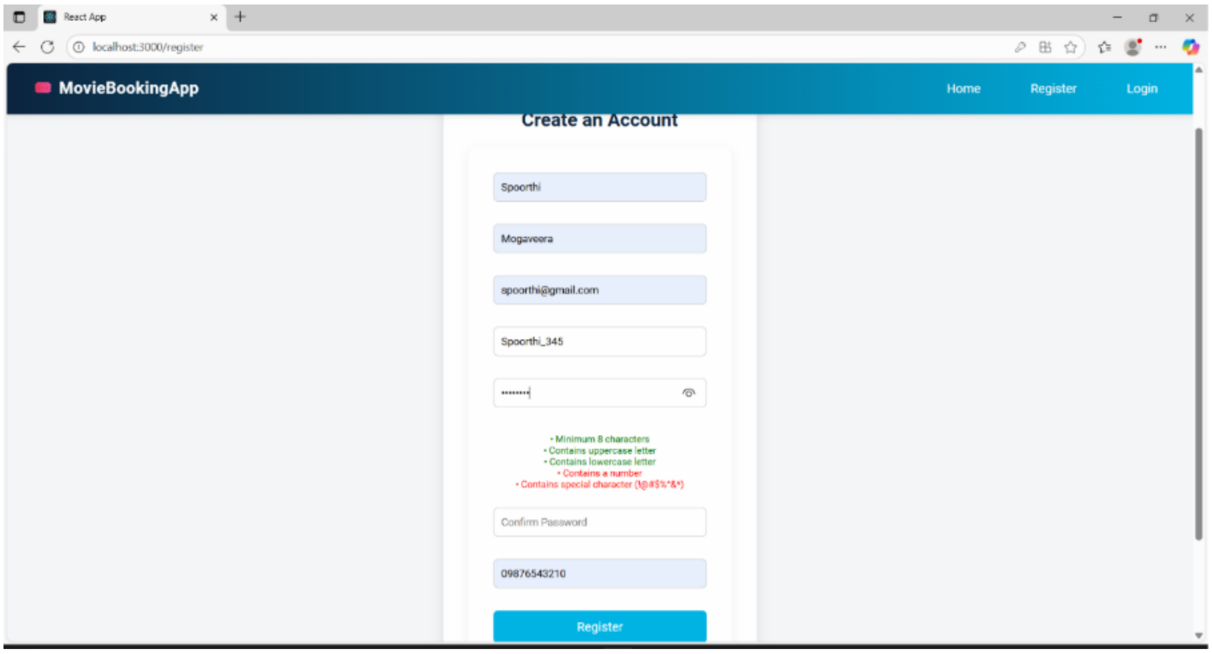
Home Register Login

Now Showing

Inception Search

Inception
Inox
Available: 78
Status: Available
Book Now

9.1.3 Registration Page Validation



Create an Account

Spoorthi

Mogaveera

spoorthi@gmail.com

Spoorthi_345

••••••••

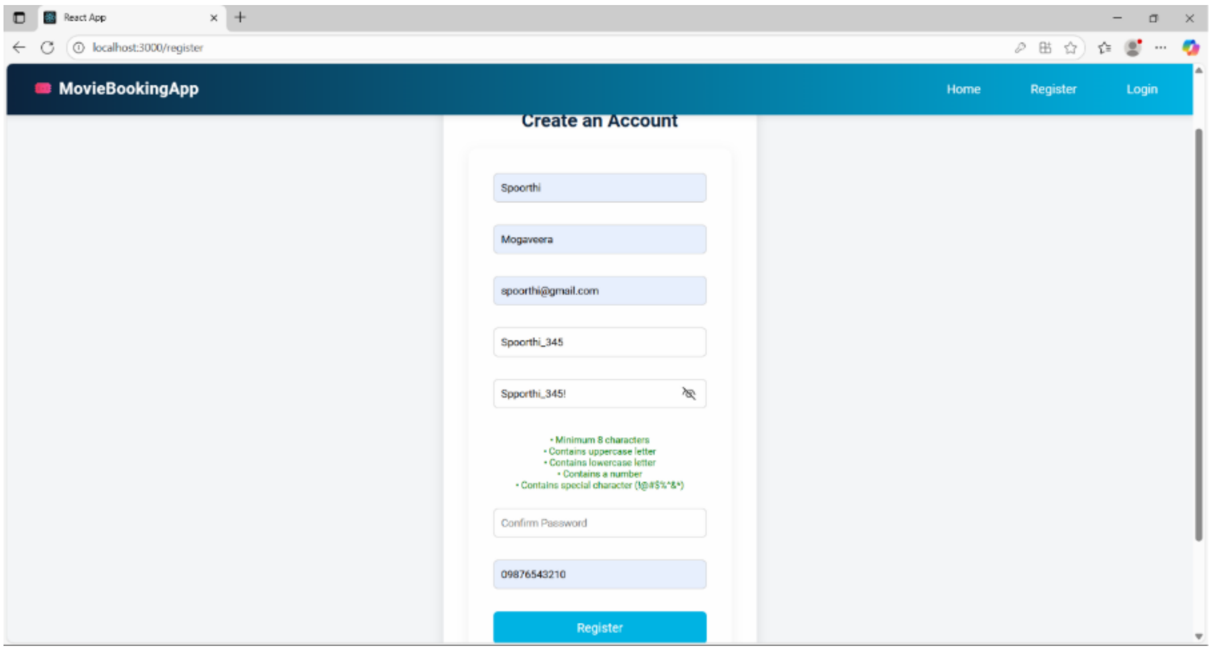
• Minimum 8 characters
• Contains uppercase letter
• Contains lowercase letter
• Contains a number
• Contains special character (!@#$%^&*)

Confirm Password

09876543210

Register

9.1.4 Registration Page Validation (Password Validation criteria)



Create an Account

Spoorthi

Mogaveera

spoorthi@gmail.com

Spoorthi_345

Spporthi_345!

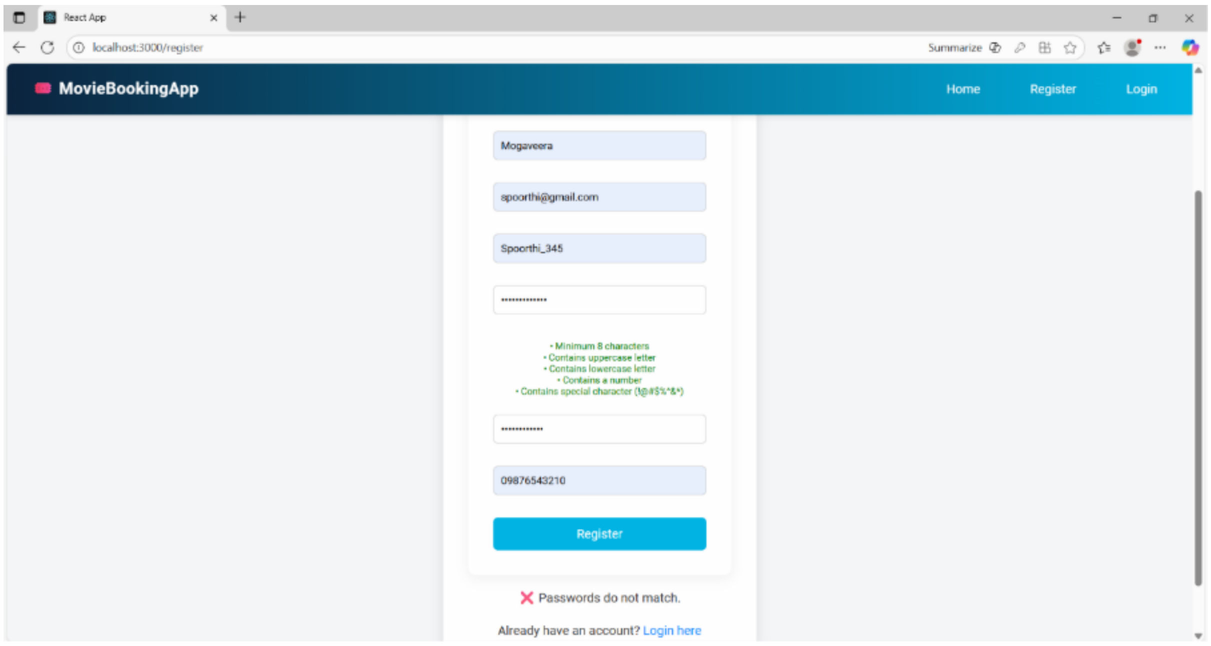
• Minimum 8 characters
• Contains uppercase letter
• Contains lowercase letter
• Contains a number
• Contains special character (!@#$%^&*)

Confirm Password

09876543210

Register

9.1.5 Registration Page Validation (Password Do not Match)



Mogaveera

spoorthi@gmail.com

Spoorthi_345

••••••••••••••

• Minimum 8 characters
• Contains uppercase letter
• Contains lowercase letter
• Contains a number
• Contains special character (!@#$%^&*)

•••••••••••••

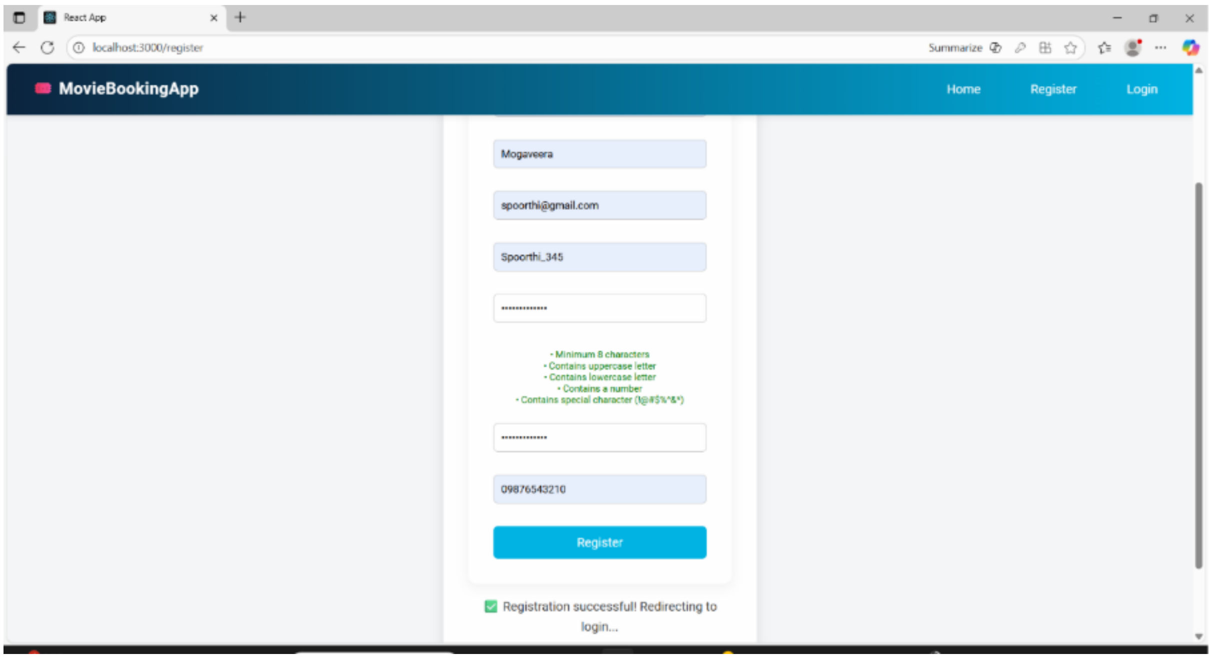
09876543210

Register

❌ Passwords do not match.

Already have an account? Login here

9.1.6 Successful Registration



Mogaveera

spoorthi@gmail.com

Spoorthi_345

••••••••••••••

• Minimum 8 characters
• Contains uppercase letter
• Contains lowercase letter
• Contains a number
• Contains special character (!@#$%^&*)

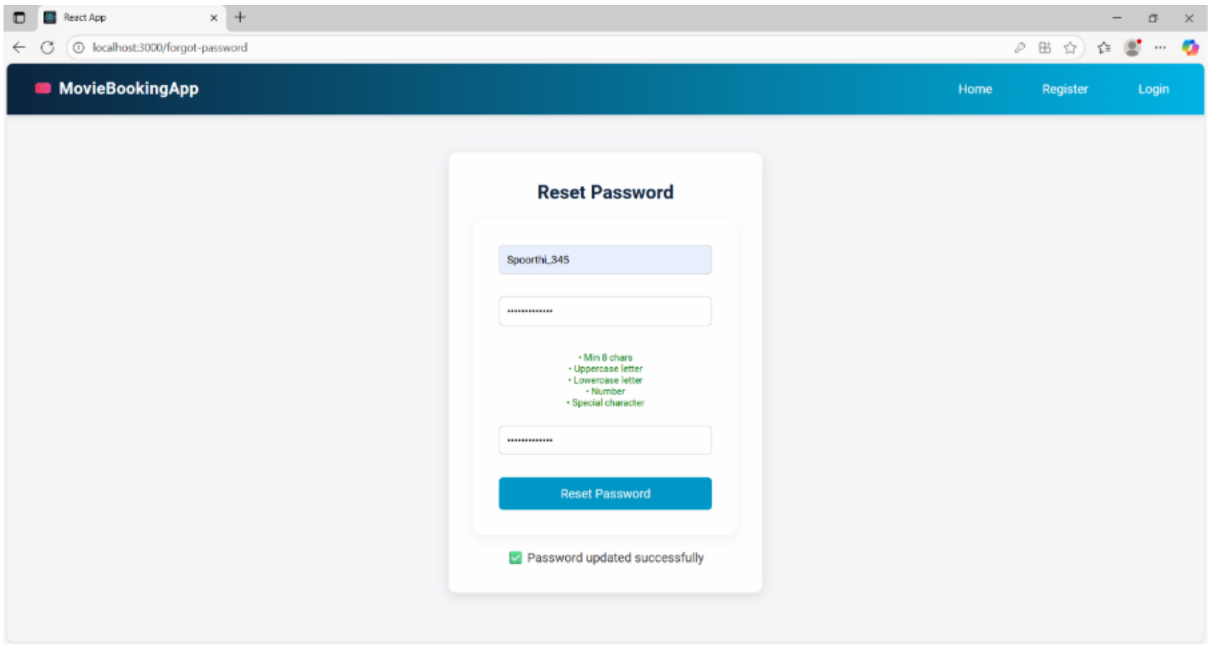
••••••••••••••

09876543210

Register

✅ Registration successful! Redirecting to login...

9.1.7 Forgot Password _Reset_option


React App

localhost:3000/forgot-password

MovieBookingApp

Home Register Login

Reset Password

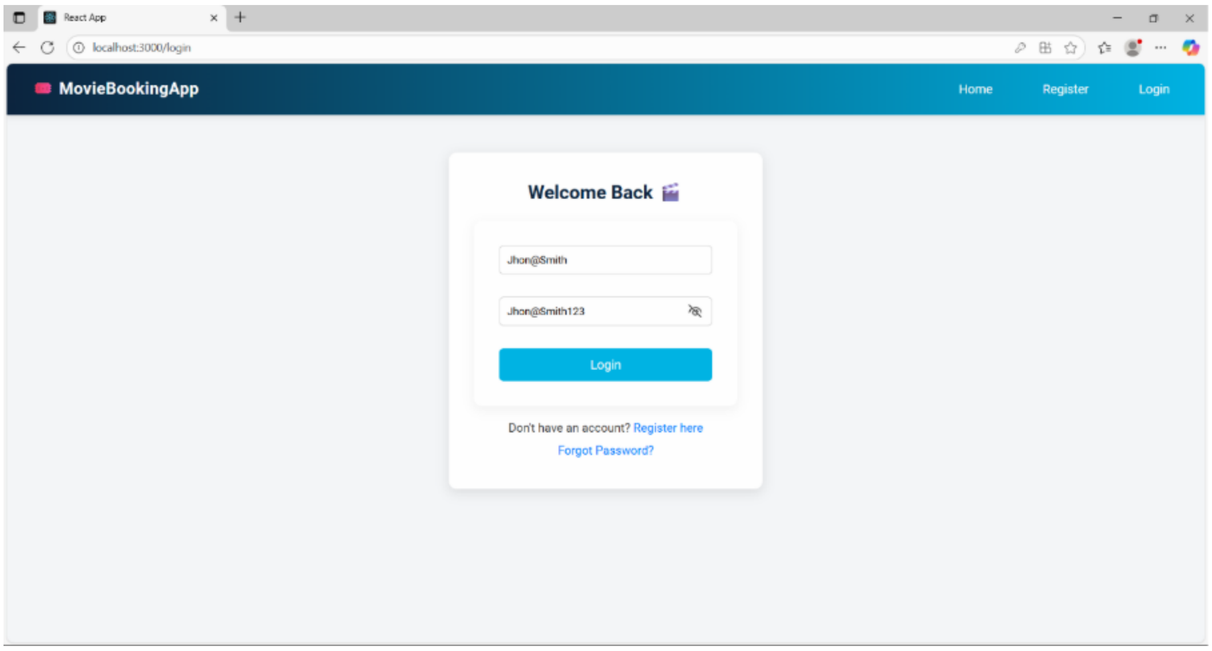
Spoorthi_345

• Min 8 chars
• Uppercase letter
• Lowercase letter
• Number
• Special character

Reset Password

✓ Password updated successfully

9.1.8 Login Page


React App

localhost:3000/login

MovieBookingApp

Home Register Login

Welcome Back

Jhon@Smith

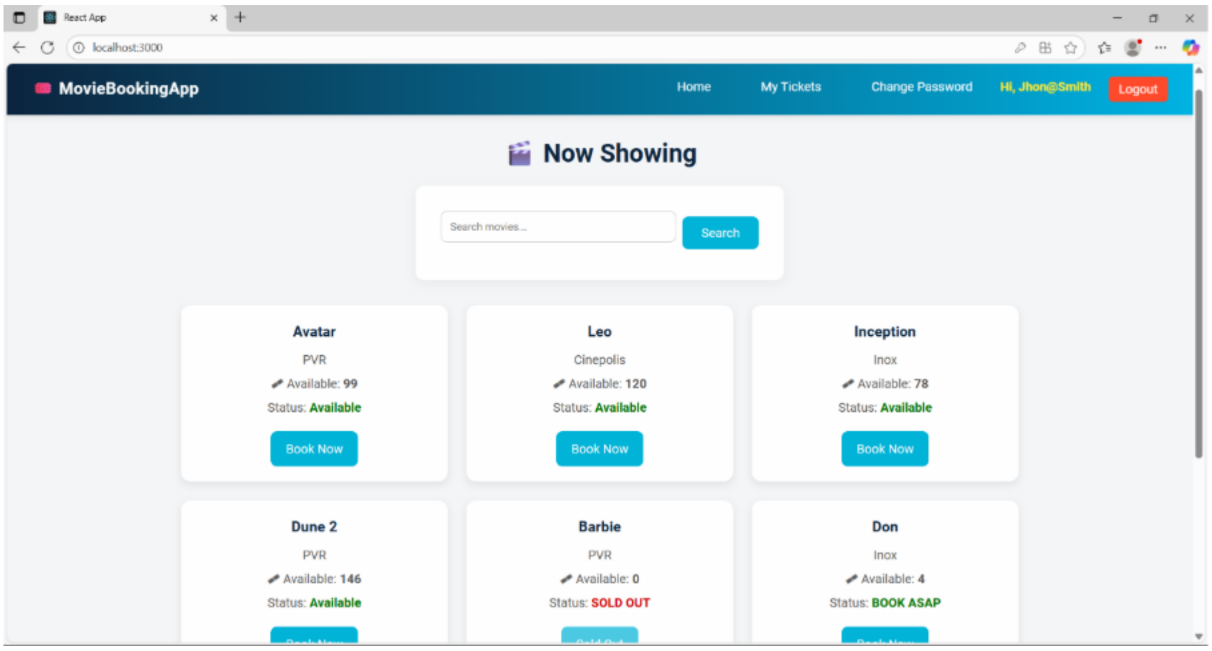
Jhon@Smith123

Login

Don't have an account? Register here

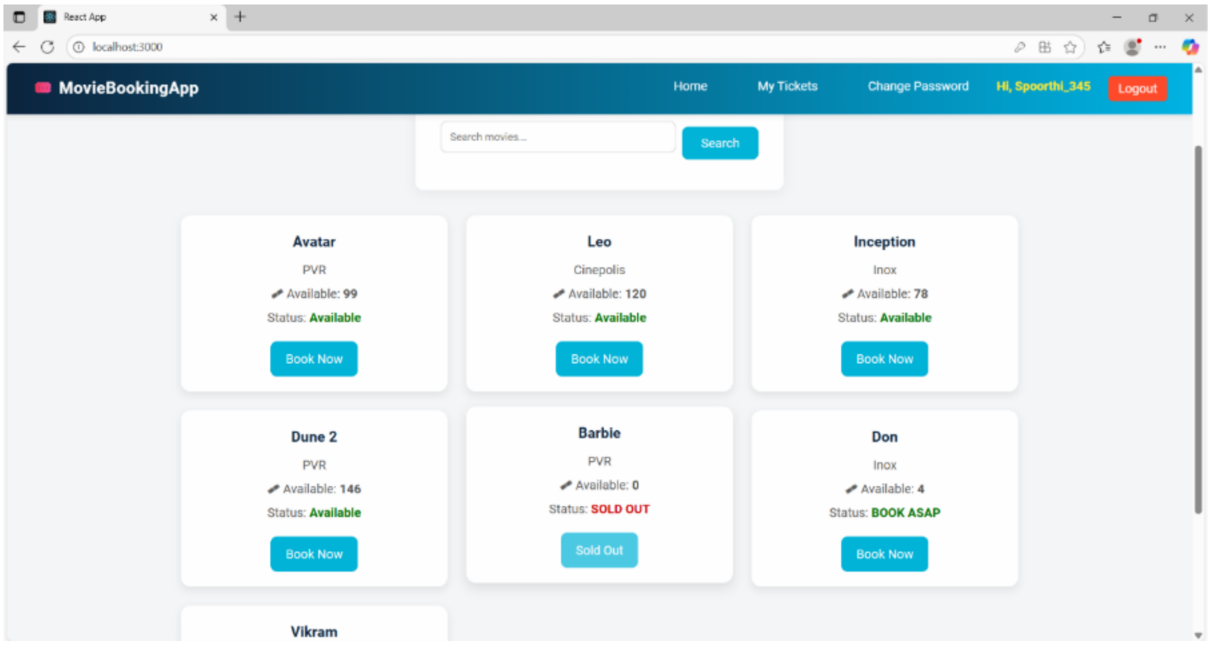
Forgot Password?

9.1.9 Home Page After Successful Login



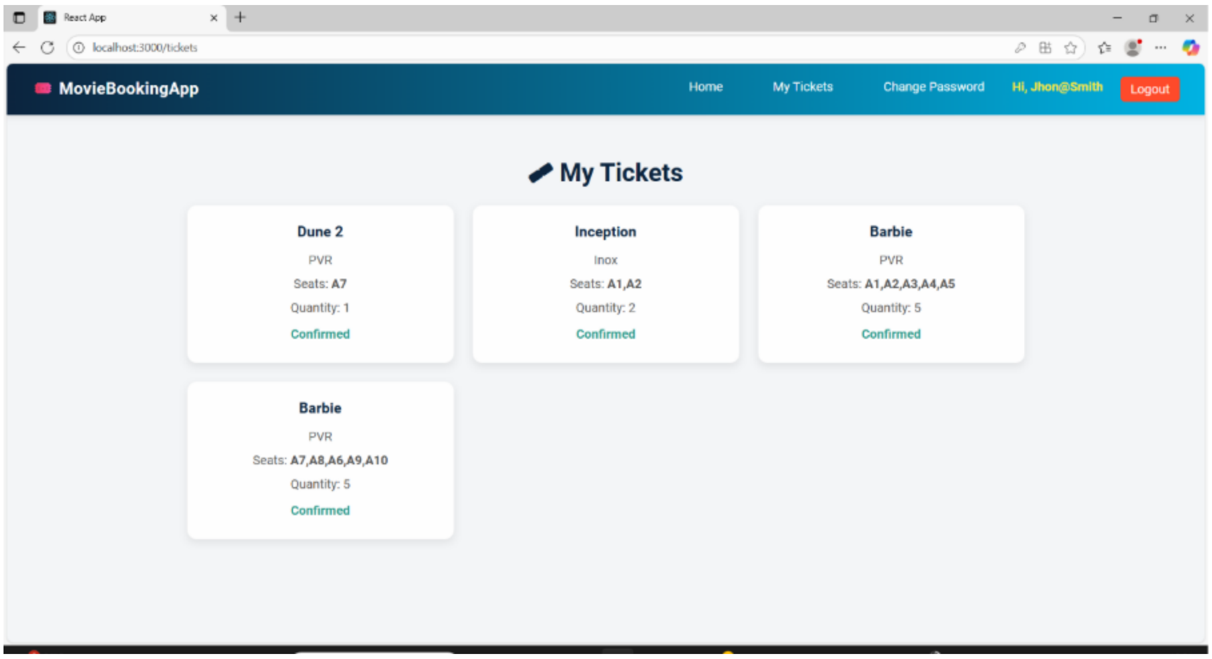
Avatar	**Leo**	**Inception**
PVR	Cinepolis	Inox
Available: **99**	Available: **120**	Available: **78**
Status: **Available**	Status: **Available**	Status: **Available**
Book Now	Book Now	Book Now
Dune 2	**Barbie**	**Don**
PVR	PVR	Inox
Available: **146**	Available: **0**	Available: **4**
Status: **Available**	Status: **SOLD OUT**	Status: **BOOK ASAP**

9.1.10Sold Out Option Disables Tickets Booking



Avatar	**Leo**	**Inception**
PVR	Cinepolis	Inox
Available: **99**	Available: **120**	Available: **78**
Status: **Available**	Status: **Available**	Status: **Available**
Book Now	Book Now	Book Now
Dune 2	**Barbie**	**Don**
PVR	PVR	Inox
Available: **146**	Available: **0**	Available: **4**
Status: **Available**	Status: **SOLD OUT**	Status: **BOOK ASAP**
Book Now	Sold Out	Book Now
Vikram		

9.1.11Can View the Tickets Booked by Logged In User



My Tickets

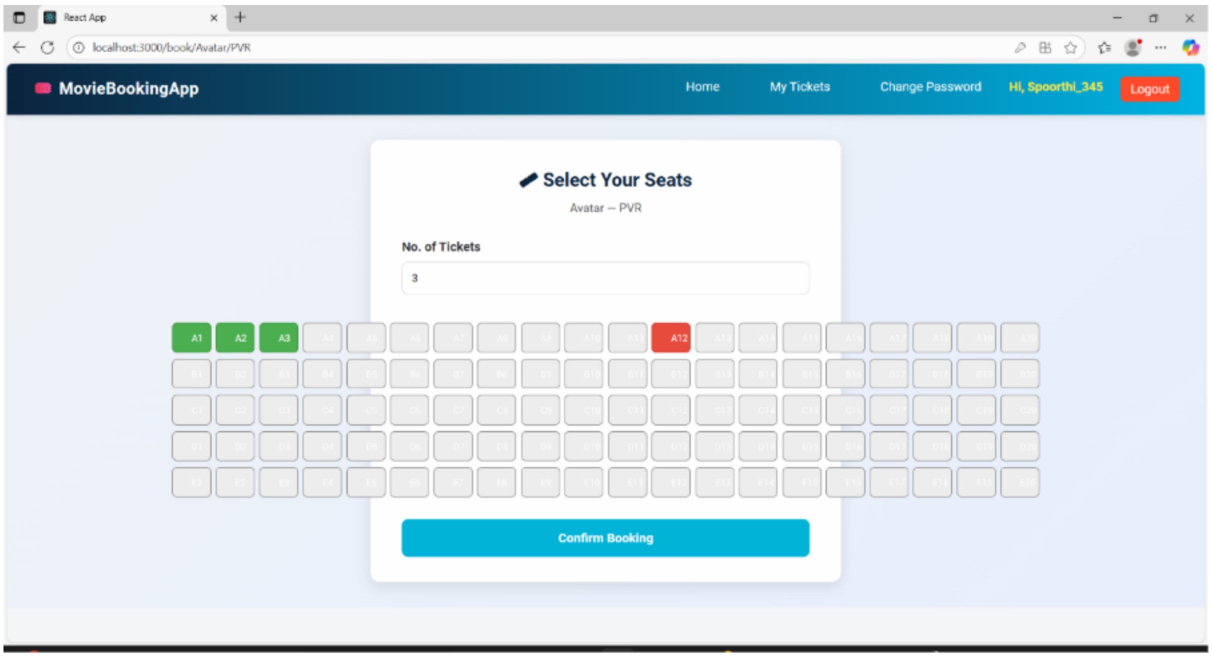
Dune 2
PVR
Seats: **A7**
Quantity: 1
Confirmed

Inception
Inox
Seats: **A1,A2**
Quantity: 2
Confirmed

Barbie
PVR
Seats: **A1,A2,A3,A4,A5**
Quantity: 5
Confirmed

Barbie
PVR
Seats: **A7,A8,A6,A9,A10**
Quantity: 5
Confirmed

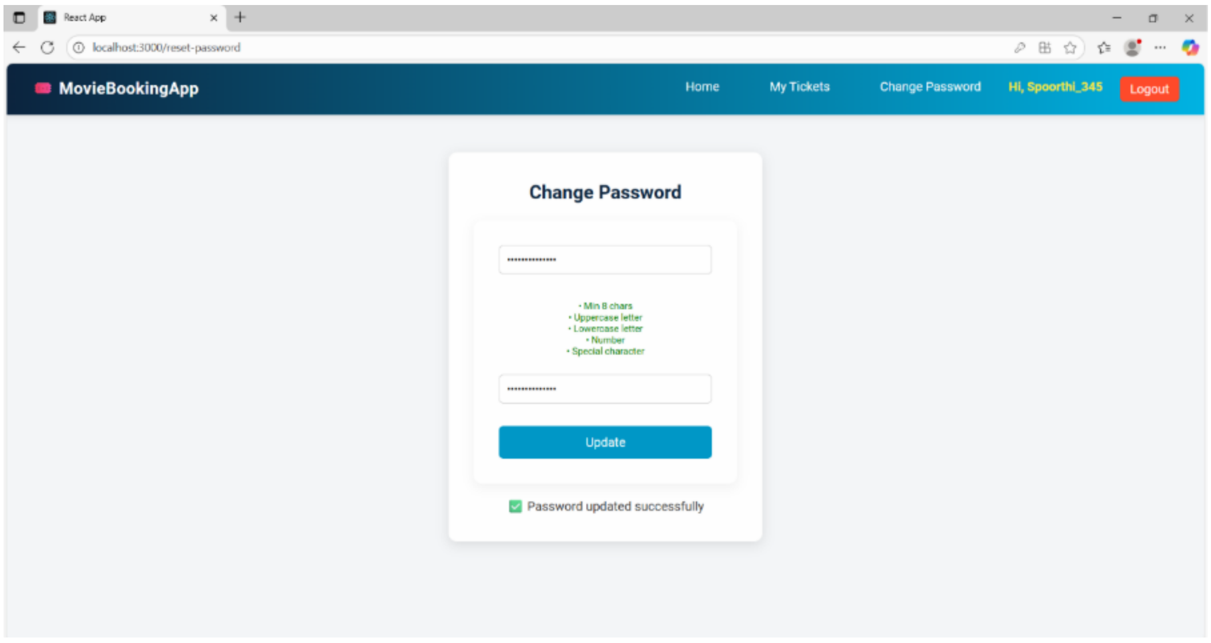
9.1.12Ticket Book Page



Select Your Seats
Avatar – PVR

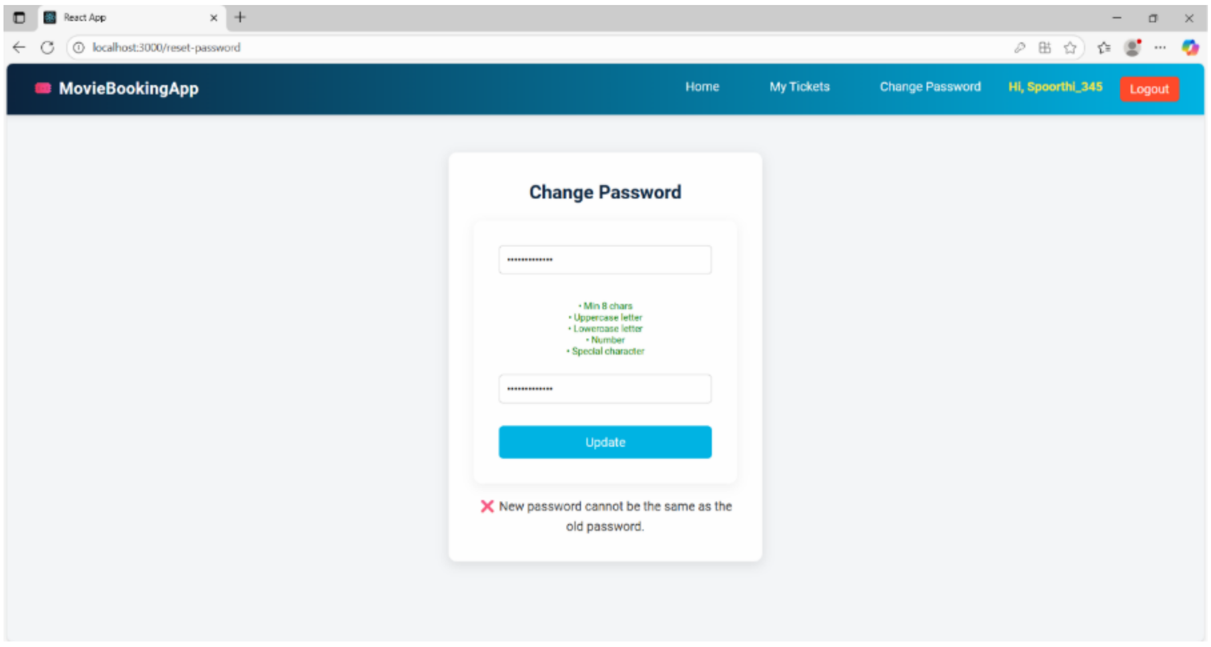
No. of Tickets
3

9.1.13Logged In User Can reset the Password Using Change Password Option



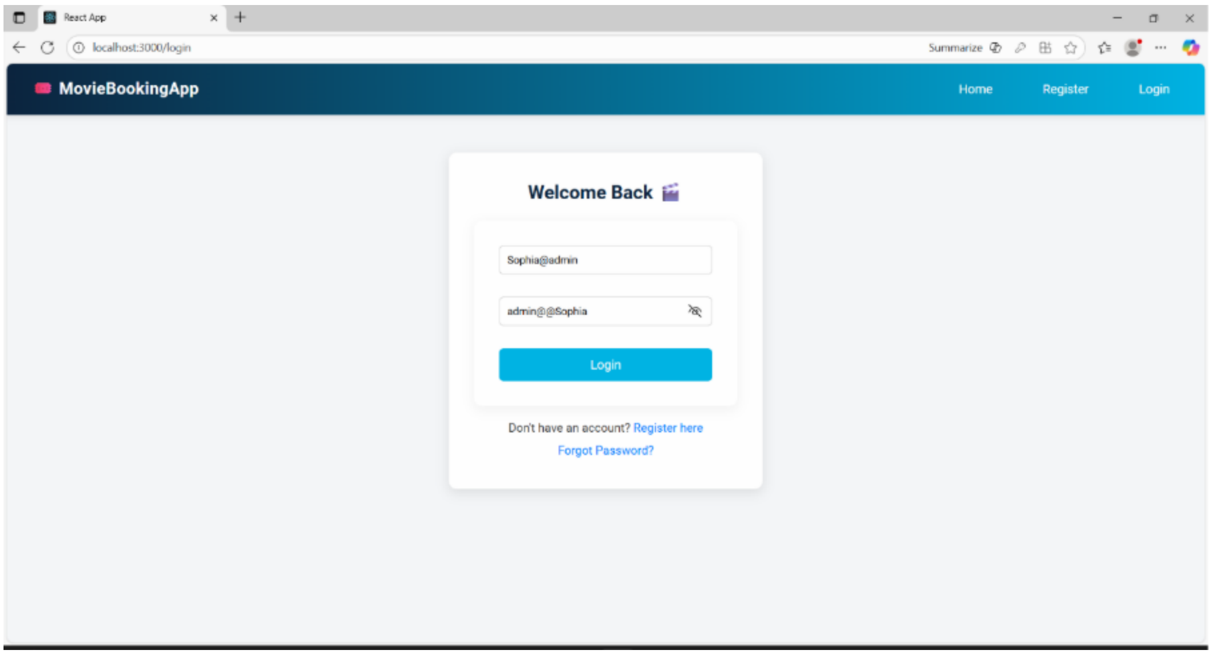
✓ Password updated successfully

9.1.14Change Password Validation



✗ New password cannot be the same as the old password.

9.1.15Admin Login



React App

localhost:3000/login

MovieBookingApp — Home — Register — Login

Welcome Back 🎬

Sophia@admin

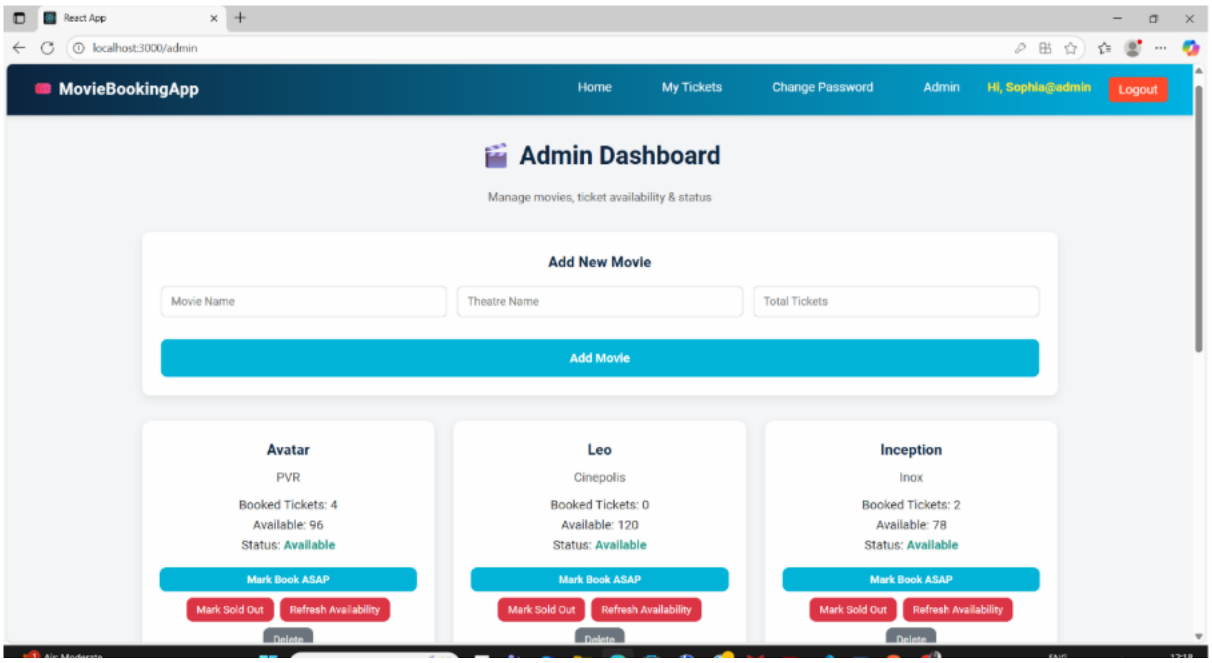
admin@@Sophia

Login

Don't have an account? Register here

Forgot Password?

9.1.16Admin Dashboard After Login



React App

localhost:3000/admin

MovieBookingApp — Home — My Tickets — Change Password — Admin — Hi, Sophia@admin — Logout

🎬 Admin Dashboard

Manage movies, ticket availability & status

Add New Movie

Movie Name — Theatre Name — Total Tickets

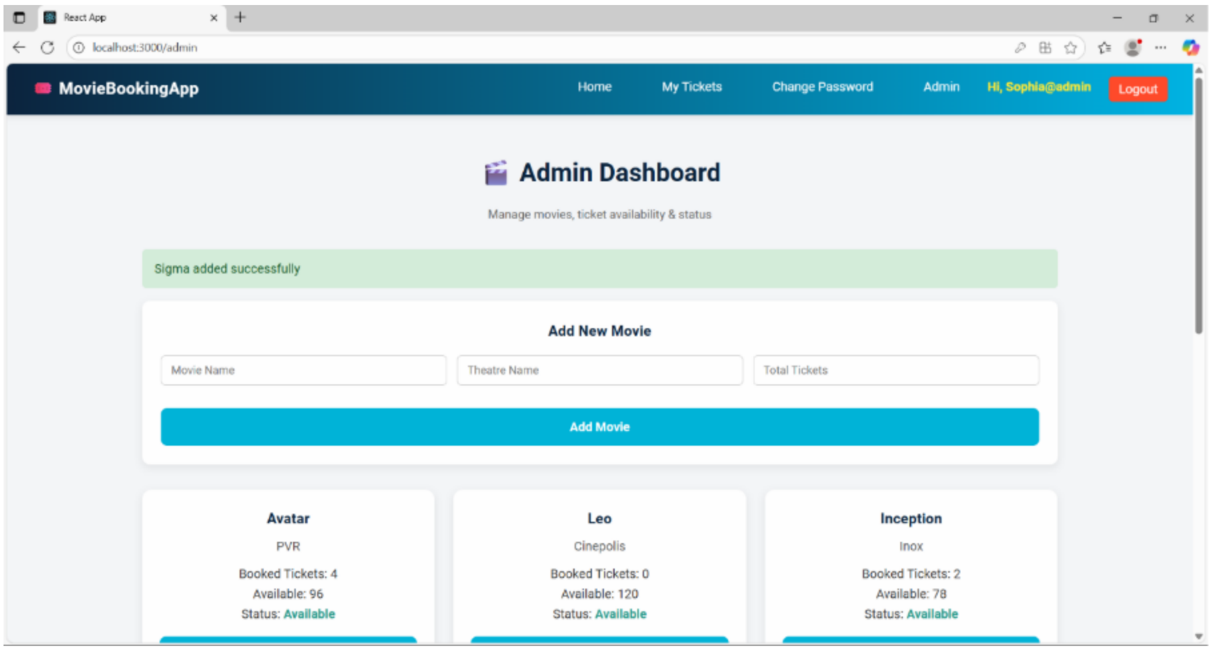
Add Movie

Avatar
PVR
Booked Tickets: 4
Available: 96
Status: Available

Leo
Cinepolis
Booked Tickets: 0
Available: 120
Status: Available

Inception
Inox
Booked Tickets: 2
Available: 78
Status: Available

9.1.17Movie added By Admin (Movie Name-Sigma)


React App

localhost:3000/admin

MovieBookingApp

Home My Tickets Change Password Admin Hi, Sophia@admin Logout

Admin Dashboard

Manage movies, ticket availability & status

Sigma added successfully

Add New Movie

Movie Name | Theatre Name | Total Tickets

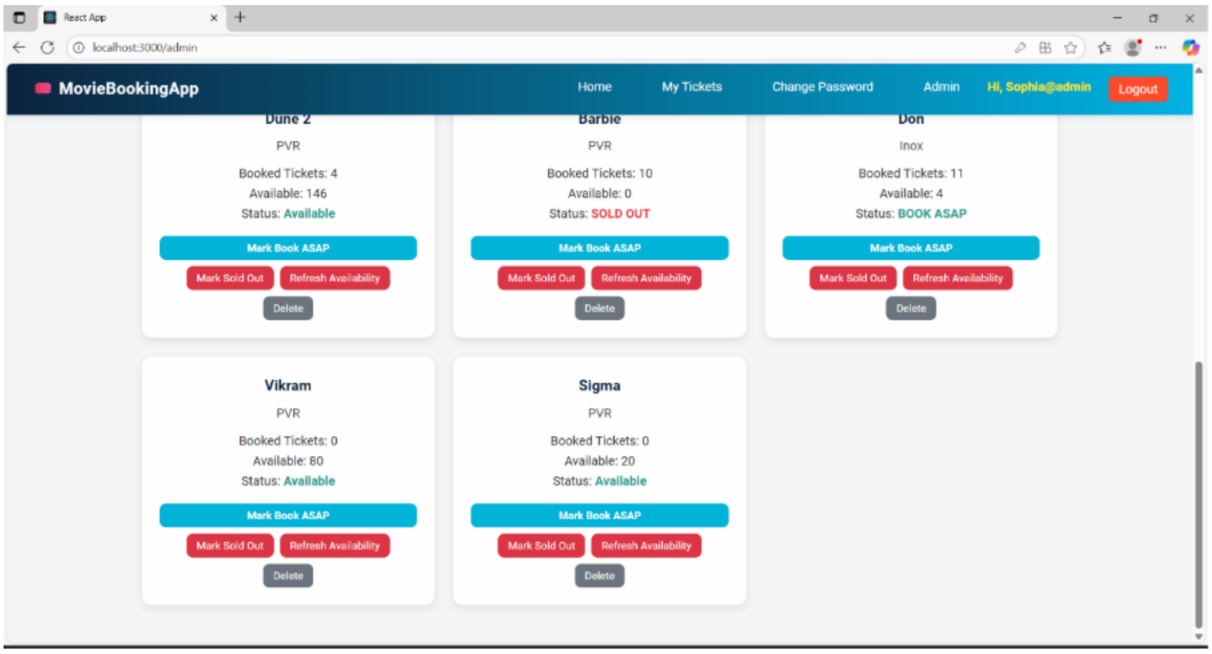
Add Movie

Avatar
PVR
Booked Tickets: 4
Available: 96
Status: Available

Leo
Cinepolis
Booked Tickets: 0
Available: 120
Status: Available

Inception
Inox
Booked Tickets: 2
Available: 78
Status: Available

9.1.18Added Movie Reflected in Dashboard


React App

localhost:3000/admin

MovieBookingApp

Home My Tickets Change Password Admin Hi, Sophia@admin Logout

Dune 2
PVR
Booked Tickets: 4
Available: 146
Status: Available

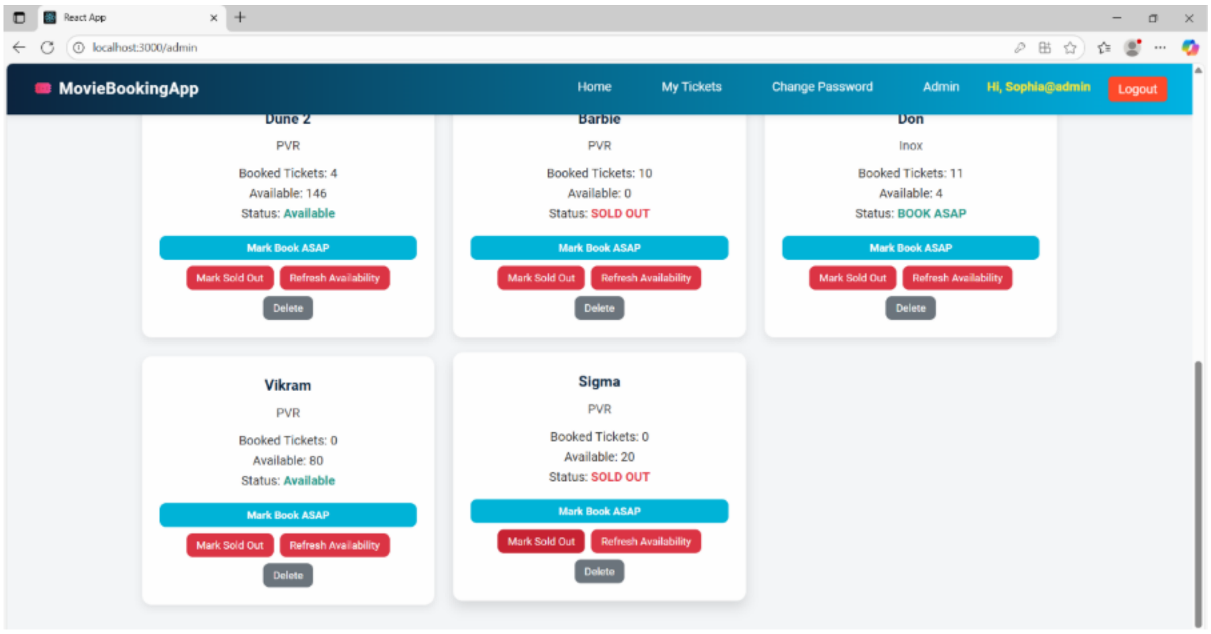
Barbie
PVR
Booked Tickets: 10
Available: 0
Status: SOLD OUT

Don
Inox
Booked Tickets: 11
Available: 4
Status: BOOK ASAP

Vikram
PVR
Booked Tickets: 0
Available: 80
Status: Available

Sigma
PVR
Booked Tickets: 0
Available: 20
Status: Available

9.1.19Admin Can Manually Mark Movie As SOLD OUT



Dune 2
PVR
Booked Tickets: 4
Available: 146
Status: Available

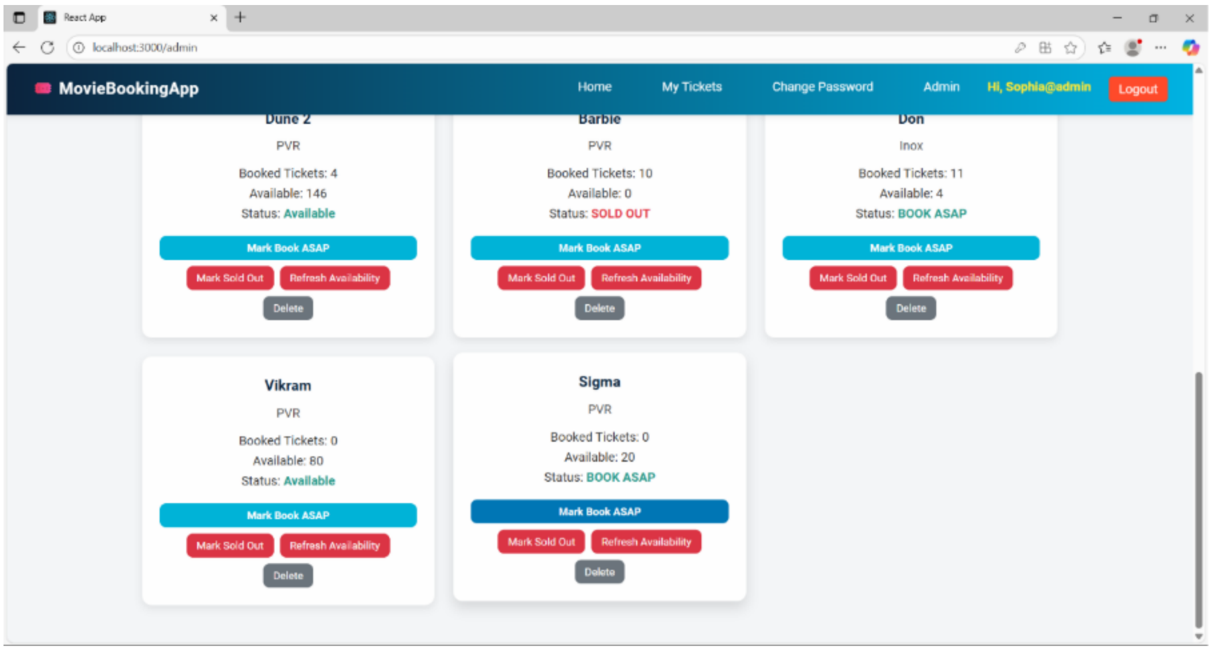
Barbie
PVR
Booked Tickets: 10
Available: 0
Status: SOLD OUT

Don
Inox
Booked Tickets: 11
Available: 4
Status: BOOK ASAP

Vikram
PVR
Booked Tickets: 0
Available: 80
Status: Available

Sigma
PVR
Booked Tickets: 0
Available: 20
Status: SOLD OUT

9.1.20Admin Can Manually Mark Movie As BOOK ASAP



Dune 2
PVR
Booked Tickets: 4
Available: 146
Status: Available

Barbie
PVR
Booked Tickets: 10
Available: 0
Status: SOLD OUT

Don
Inox
Booked Tickets: 11
Available: 4
Status: BOOK ASAP

Vikram
PVR
Booked Tickets: 0
Available: 80
Status: Available

Sigma
PVR
Booked Tickets: 0
Available: 20
Status: BOOK ASAP

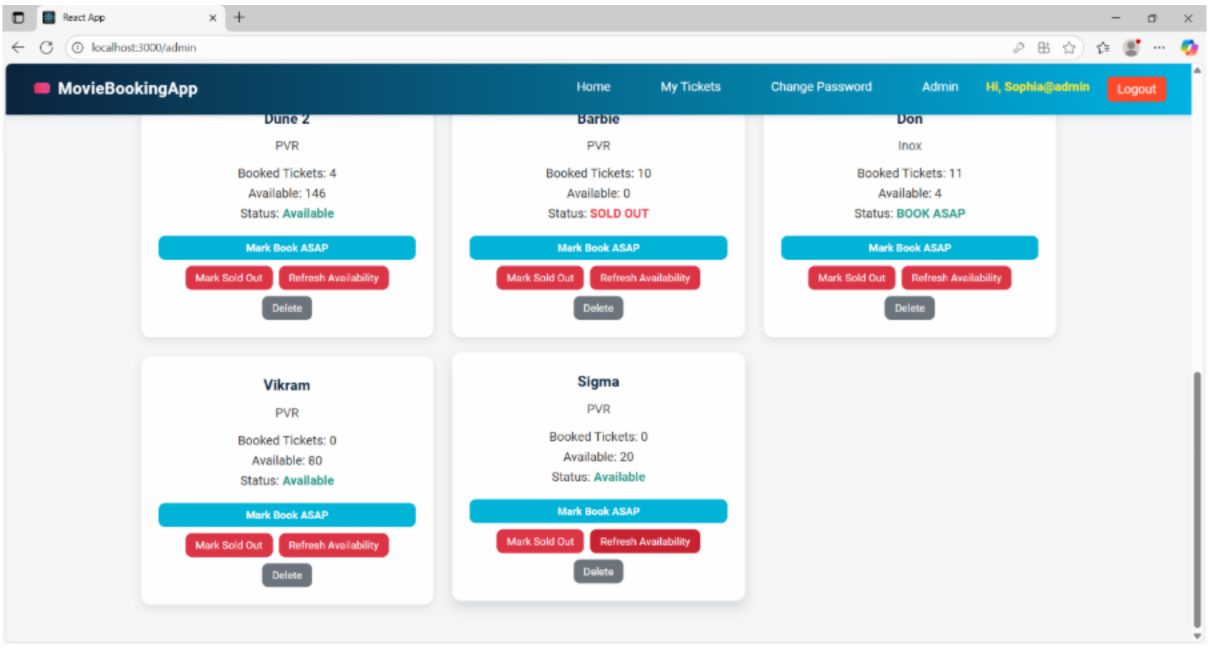
9.1.21Through Refresh Availability Option Ticket Status will be Set again

Condition

I. If Available tickets equal to Zero, Status Will be marked as SOLD OUT

II. If Available tickets less than or equal to Half of the Total Tickets marked as BOOK ASAP.

III. If Available tickets more than Booked Tickets, status will be marked as Available



Dune 2
PVR
Booked Tickets: 4
Available: 146
Status: Available

Barbie
PVR
Booked Tickets: 10
Available: 0
Status: SOLD OUT

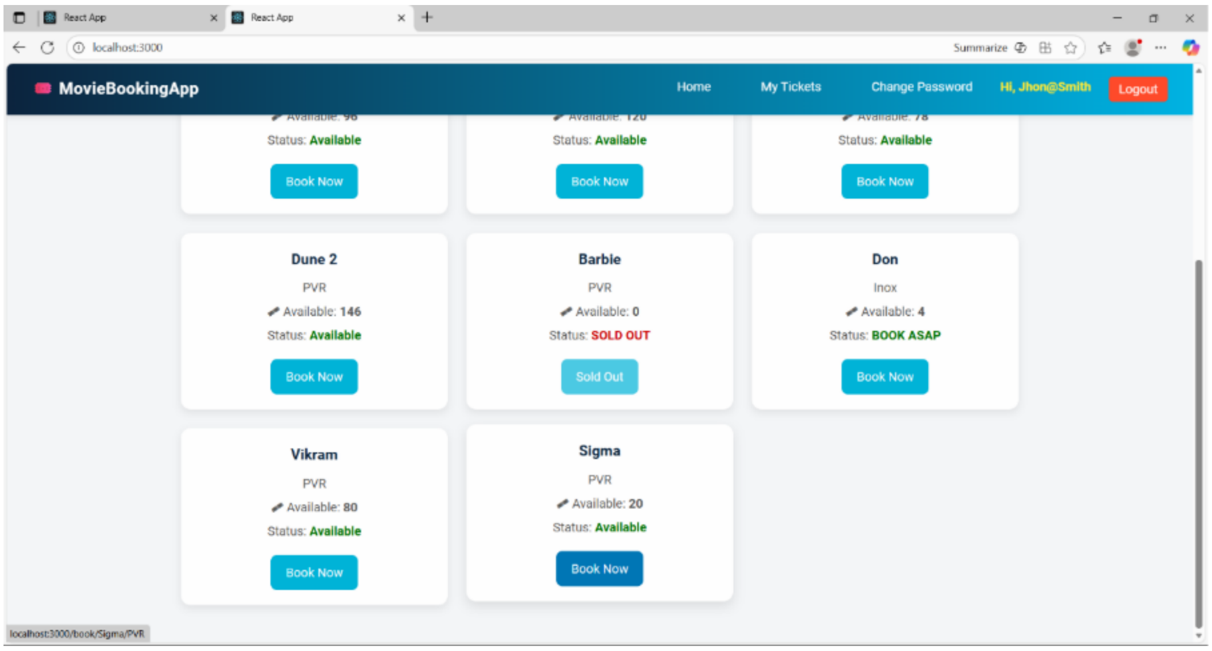
Don
Inox
Booked Tickets: 11
Available: 4
Status: BOOK ASAP

Vikram
PVR
Booked Tickets: 0
Available: 80
Status: Available

Sigma
PVR
Booked Tickets: 0
Available: 20
Status: Available

So here we are providing flexibility to admin, of Manually setting the status of Movie tickets booking and Automatically status changes based on ticket availability.

9.1.22 Normal Logged in User Can see the newly added Movie and Can book the tickets



React App | React App

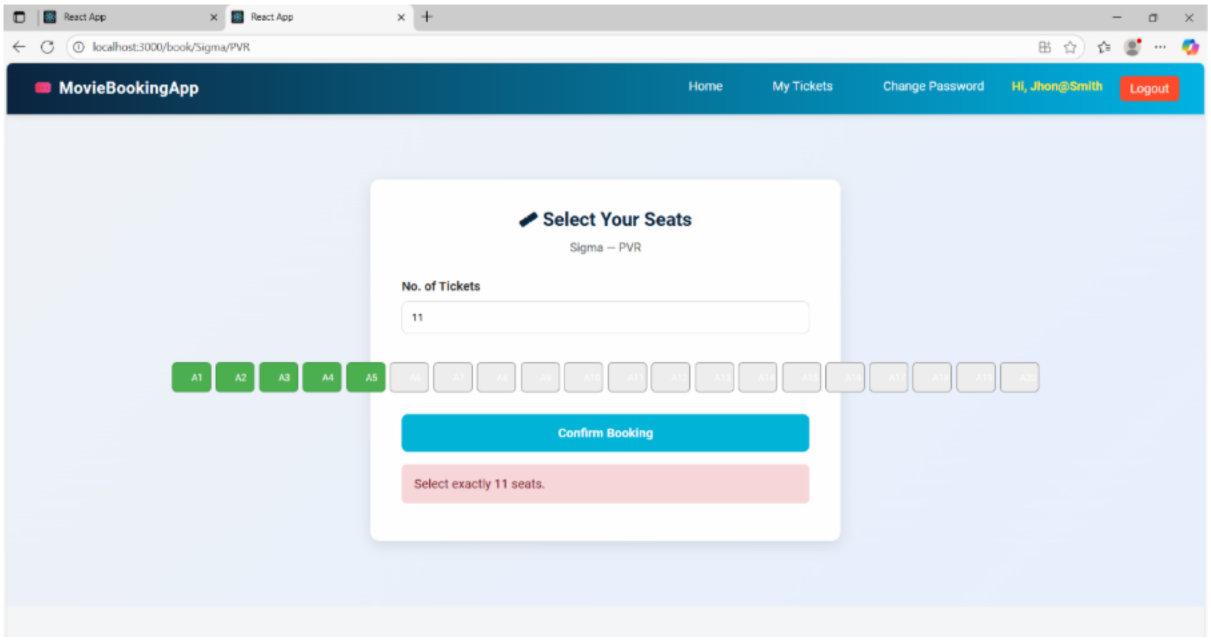
localhost:3000

MovieBookingApp — Home | My Tickets | Change Password | Hi, Jhon@Smith | Logout

Movie	Theatre	Available	Status	Action
Dune 2	PVR	146	Available	Book Now
Barbie	PVR	0	SOLD OUT	Sold Out
Don	Inox	4	BOOK ASAP	Book Now
Vikram	PVR	80	Available	Book Now
Sigma	PVR	20	Available	Book Now

localhost:3000/book/Sigma/PVR

9.1.23 Ticket Book Validation Functionality



localhost:3000/book/Sigma/PVR

MovieBookingApp — Home | My Tickets | Change Password | Hi, Jhon@Smith | Logout

Select Your Seats

Sigma – PVR

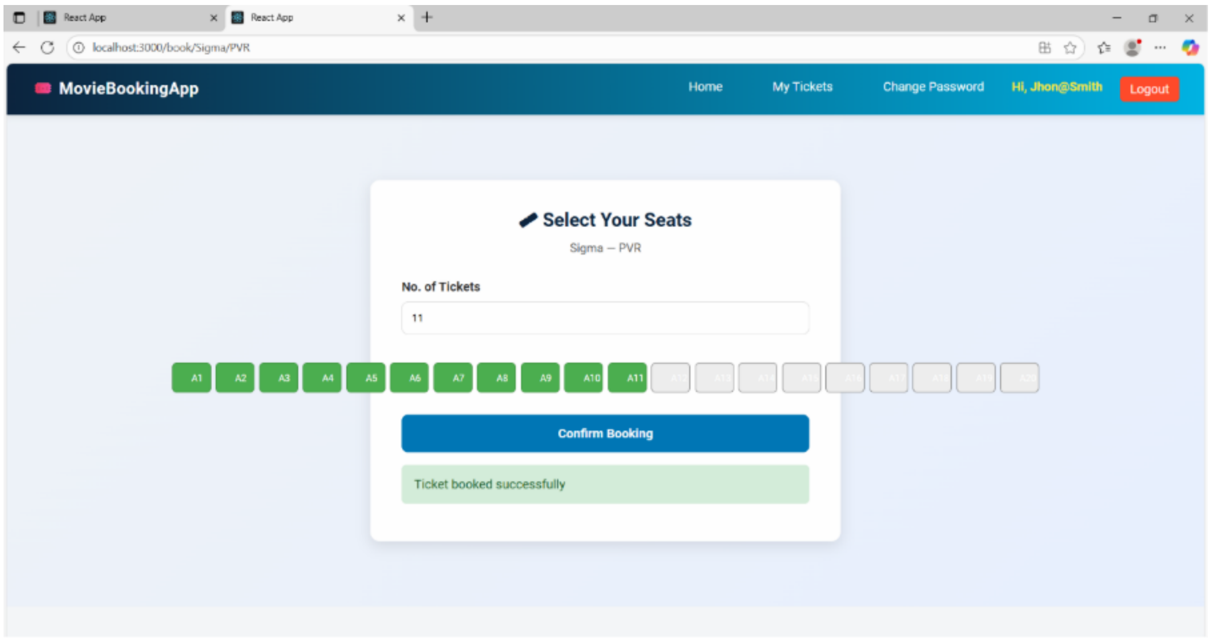
No. of Tickets: 11

A1 A2 A3 A4 A5 A6 A7 A8 A9 A10 A11 A12 A13 A14 A15 A16 A17 A18 A19 A20

Confirm Booking

Select exactly 11 seats.

9.1.24 Successful Ticket Booked



React App | React App

localhost:3000/book/Sigma/PVR

MovieBookingApp — Home — My Tickets — Change Password — Hi, Jhon@Smith — Logout

Select Your Seats

Sigma — PVR

No. of Tickets

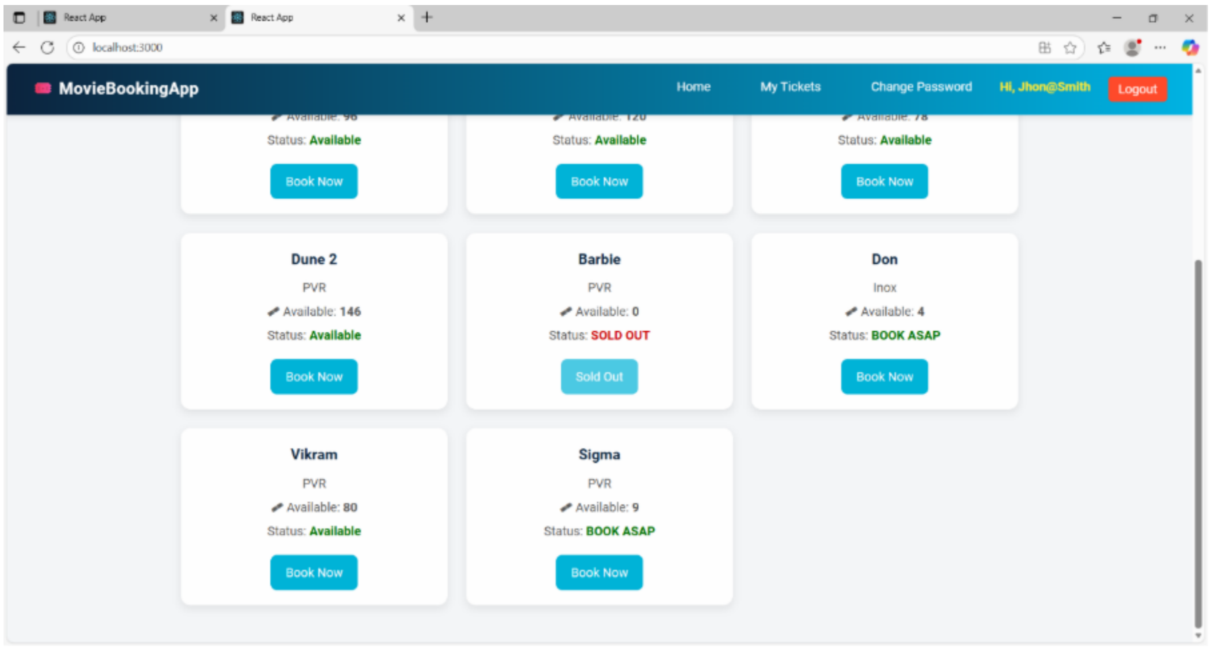
11

A1 A2 A3 A4 A5 A6 A7 A8 A9 A10 A11 A12 A13 A14 A15 A16 A17 A18 A19 A20

Confirm Booking

Ticket booked successfully

9.1.25 Now the Movie Ticket Booking Status Changes Based on Ticket Availability



React App | React App

localhost:3000

MovieBookingApp — Home — My Tickets — Change Password — Hi, Jhon@Smith — Logout

Available: 96
Status: Available
Book Now

Available: 120
Status: Available
Book Now

Available: 78
Status: Available
Book Now

Dune 2
PVR
Available: 146
Status: Available
Book Now

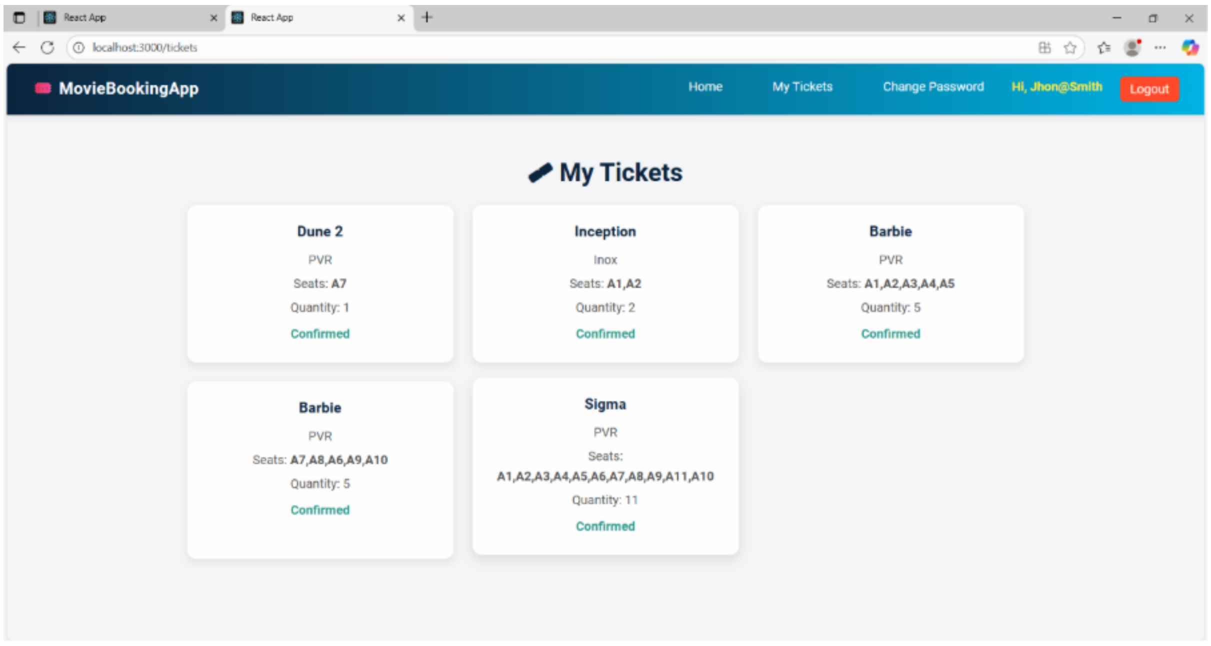
Barbie
PVR
Available: 0
Status: SOLD OUT
Sold Out

Don
Inox
Available: 4
Status: BOOK ASAP
Book Now

Vikram
PVR
Available: 80
Status: Available
Book Now

Sigma
PVR
Available: 9
Status: BOOK ASAP
Book Now

9.1.26 Ticket Also reflected in Ticket List


MovieBookingApp

Home | My Tickets | Change Password | Hi, Jhon@Smith | Logout

My Tickets

Dune 2
PVR
Seats: A7
Quantity: 1
Confirmed

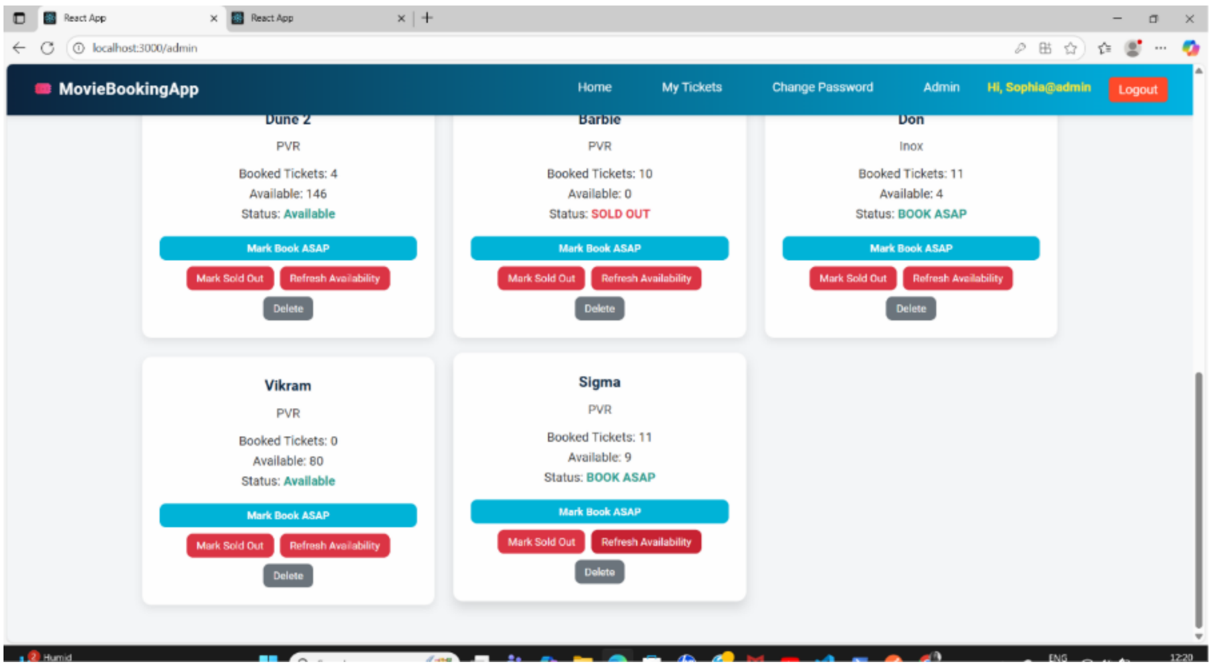
Inception
Inox
Seats: A1,A2
Quantity: 2
Confirmed

Barbie
PVR
Seats: A1,A2,A3,A4,A5
Quantity: 5
Confirmed

Barbie
PVR
Seats: A7,A8,A6,A9,A10
Quantity: 5
Confirmed

Sigma
PVR
Seats: A1,A2,A3,A4,A5,A6,A7,A8,A9,A11,A10
Quantity: 11
Confirmed

9.1.27In Admin Dashboard Also, changes Reflected through Refresh Availability option


MovieBookingApp

Home | My Tickets | Change Password | Admin | Hi, Sophia@admin | Logout

Dune 2
PVR
Booked Tickets: 4
Available: 146
Status: Available

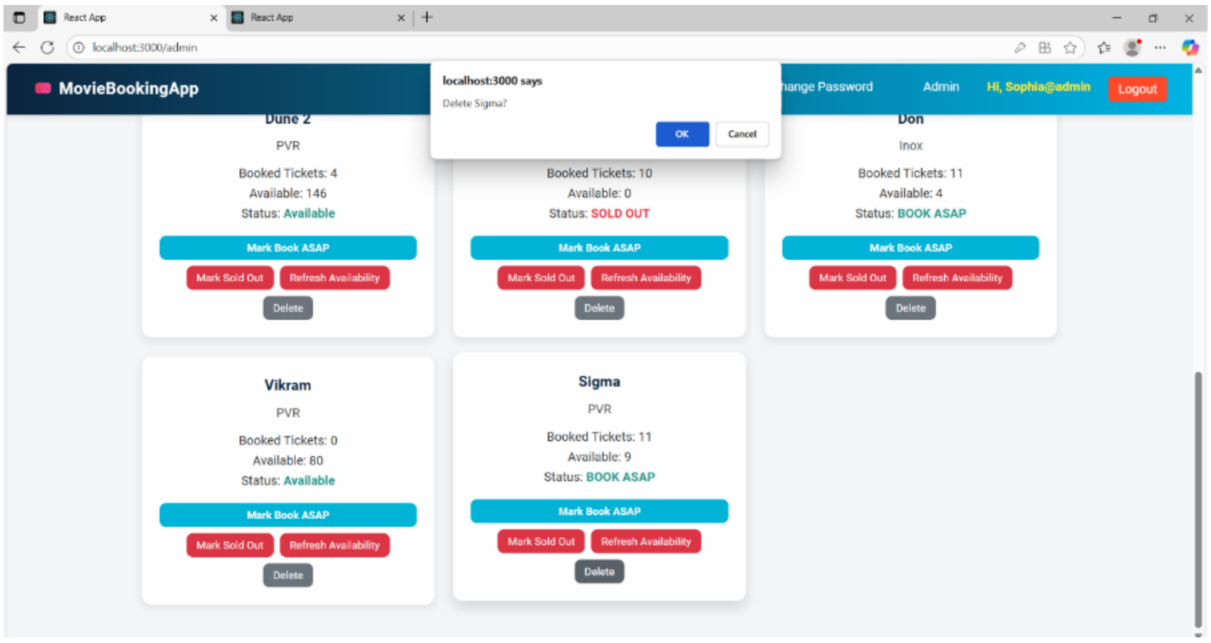
Barbie
PVR
Booked Tickets: 10
Available: 0
Status: SOLD OUT

Don
Inox
Booked Tickets: 11
Available: 4
Status: BOOK ASAP

Vikram
PVR
Booked Tickets: 0
Available: 80
Status: Available

Sigma
PVR
Booked Tickets: 11
Available: 9
Status: BOOK ASAP

9.1.28Admin Can Delete the Movie



localhost:3000 says

Delete Sigma?

OK Cancel

Dune 2
PVR
Booked Tickets: 4
Available: 146
Status: Available

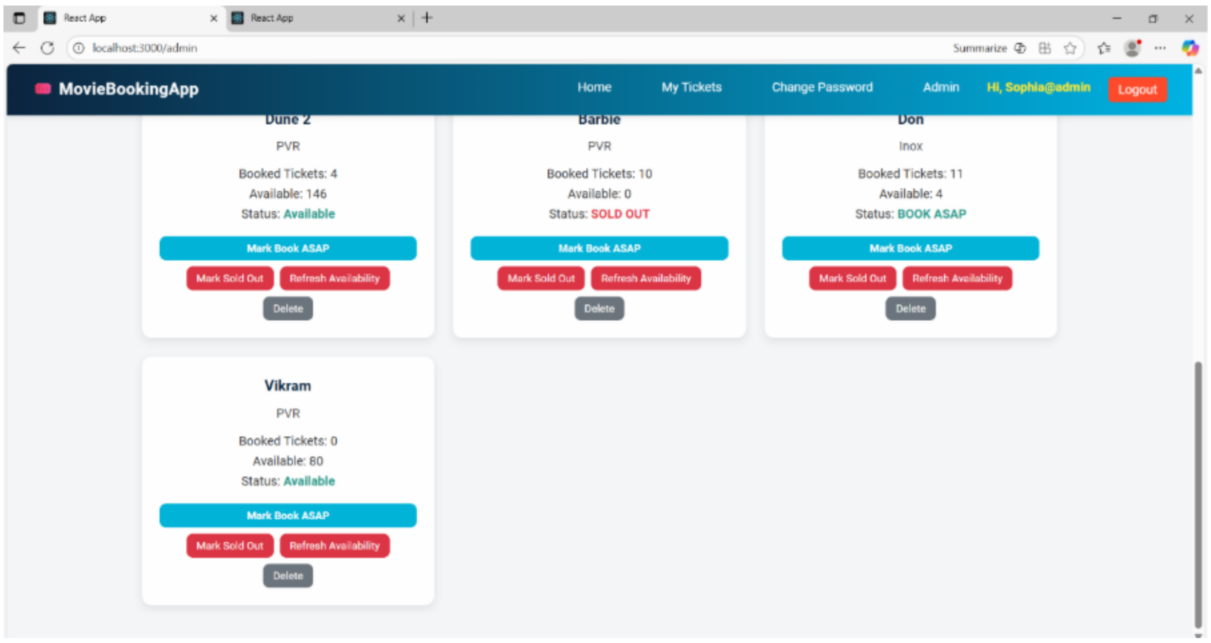
Booked Tickets: 10
Available: 0
Status: SOLD OUT

Don
Inox
Booked Tickets: 11
Available: 4
Status: BOOK ASAP

Vikram
PVR
Booked Tickets: 0
Available: 80
Status: Available

Sigma
PVR
Booked Tickets: 11
Available: 9
Status: BOOK ASAP

9.1.29Admin Dashboard After Deleting Sigma Movie



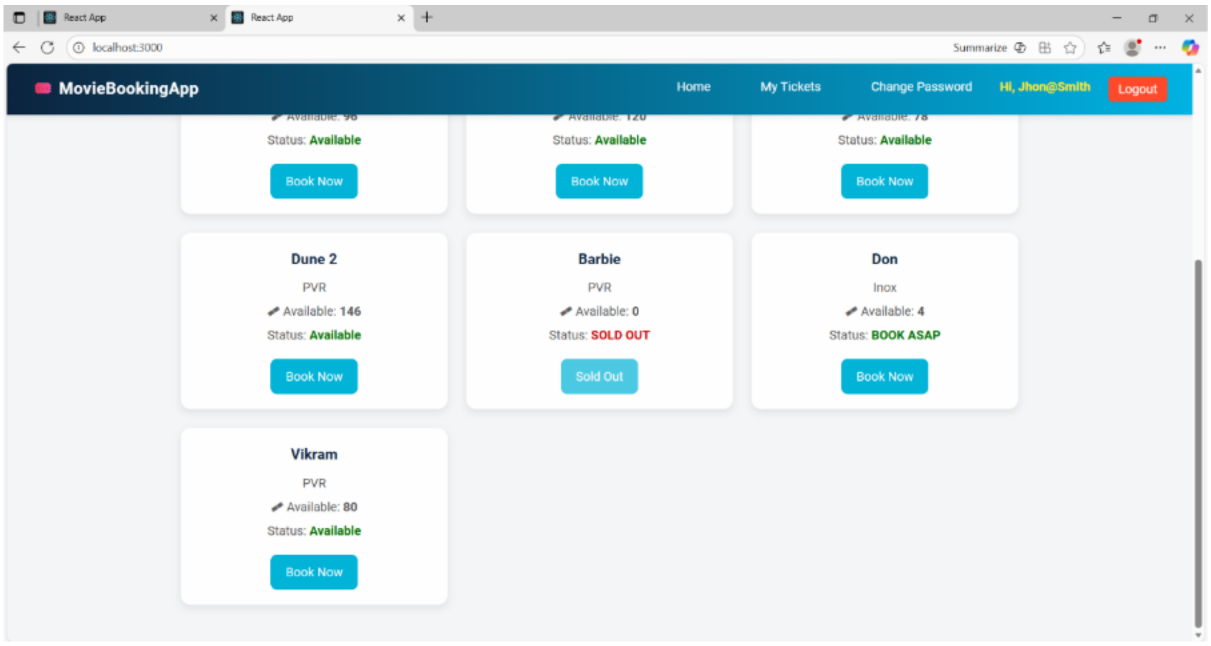
Dune 2
PVR
Booked Tickets: 4
Available: 146
Status: Available

Barbie
PVR
Booked Tickets: 10
Available: 0
Status: SOLD OUT

Don
Inox
Booked Tickets: 11
Available: 4
Status: BOOK ASAP

Vikram
PVR
Booked Tickets: 0
Available: 80
Status: Available

9.1.30In User Home Page Movie Removed



MovieBookingApp

Home My Tickets Change Password Hi, Jhon@Smith Logout

Status: Available

Book Now

Status: Available

Book Now

Status: Available

Book Now

Dune 2

PVR

Available: 146

Status: Available

Book Now

Barbie

PVR

Available: 0

Status: SOLD OUT

Sold Out

Don

Inox

Available: 4

Status: BOOK ASAP

Book Now

Vikram

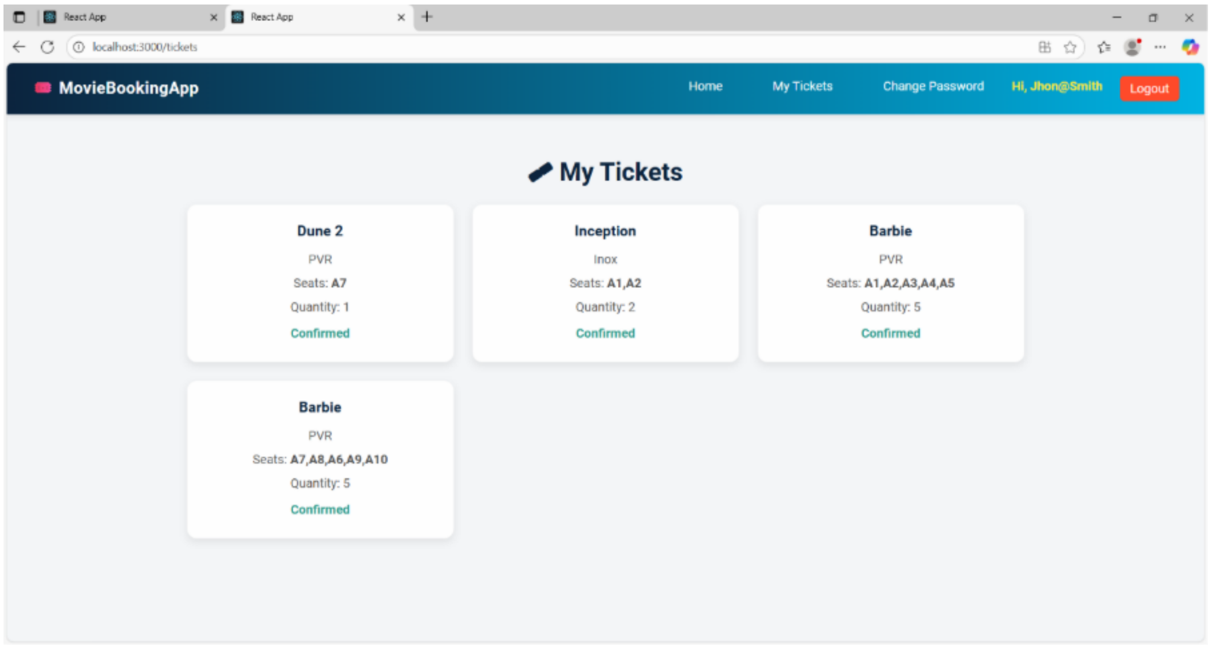
PVR

Available: 80

Status: Available

Book Now

9.1.31 The ticket was also Removed for the Respective Deleted Movie



MovieBookingApp

Home My Tickets Change Password Hi, Jhon@Smith Logout

My Tickets

Dune 2

PVR

Seats: A7

Quantity: 1

Confirmed

Inception

Inox

Seats: A1,A2

Quantity: 2

Confirmed

Barbie

PVR

Seats: A1,A2,A3,A4,A5

Quantity: 5

Confirmed

Barbie

PVR

Seats: A7,A8,A6,A9,A10

Quantity: 5

Confirmed