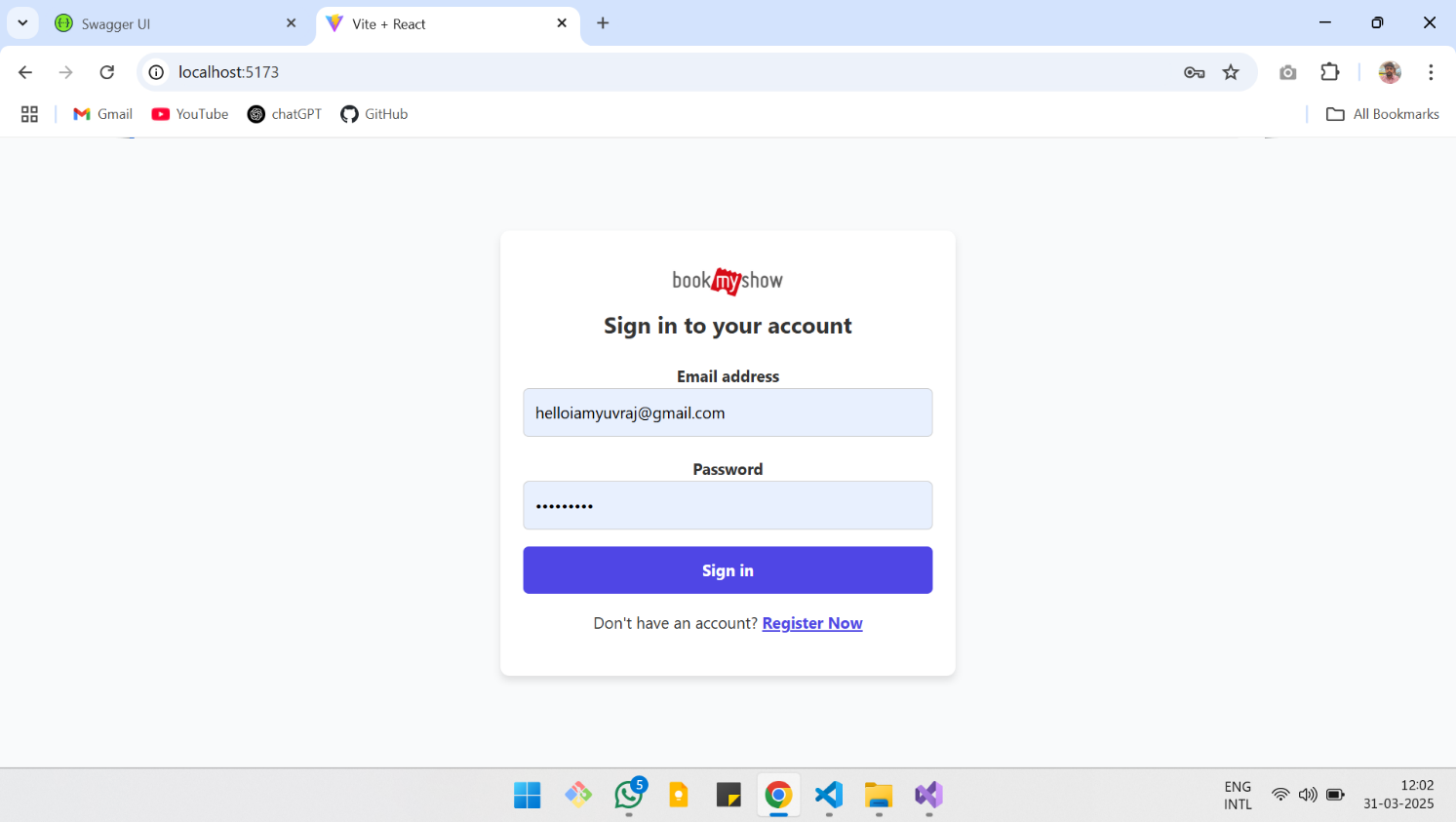
Movie Ticket Booking System Documentation

Overview

This project is a full-stack movie ticket booking system inspired by BookMyShow. It allows users to browse movies, view available slots, select seats, and book tickets seamlessly.

Frontend

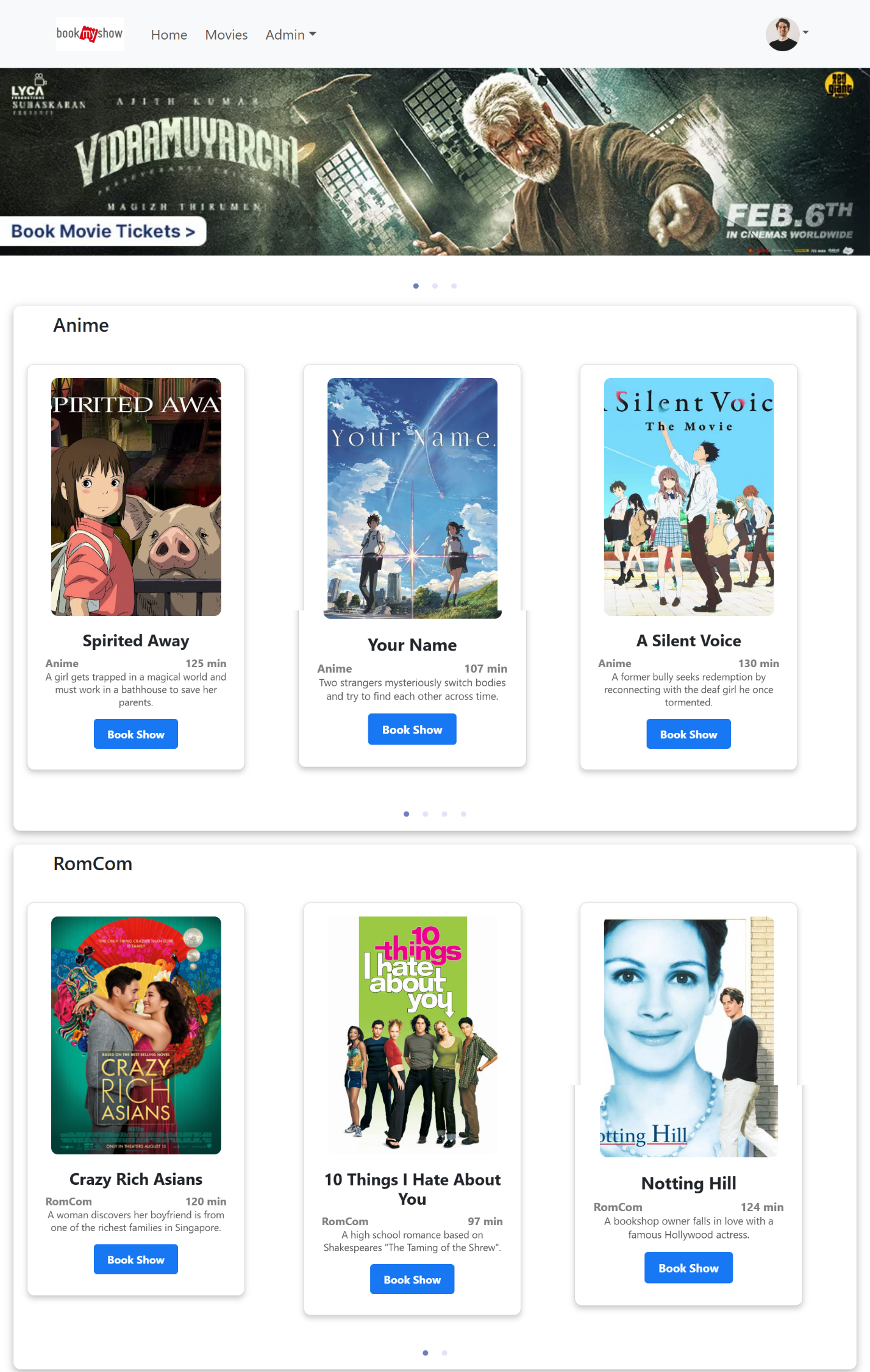
1. Login Page
   * Users must enter valid credentials to proceed to the home page.
   * If credentials are incorrect, an error message is displayed.
   * URL: /api/auth/login



1. Home Page

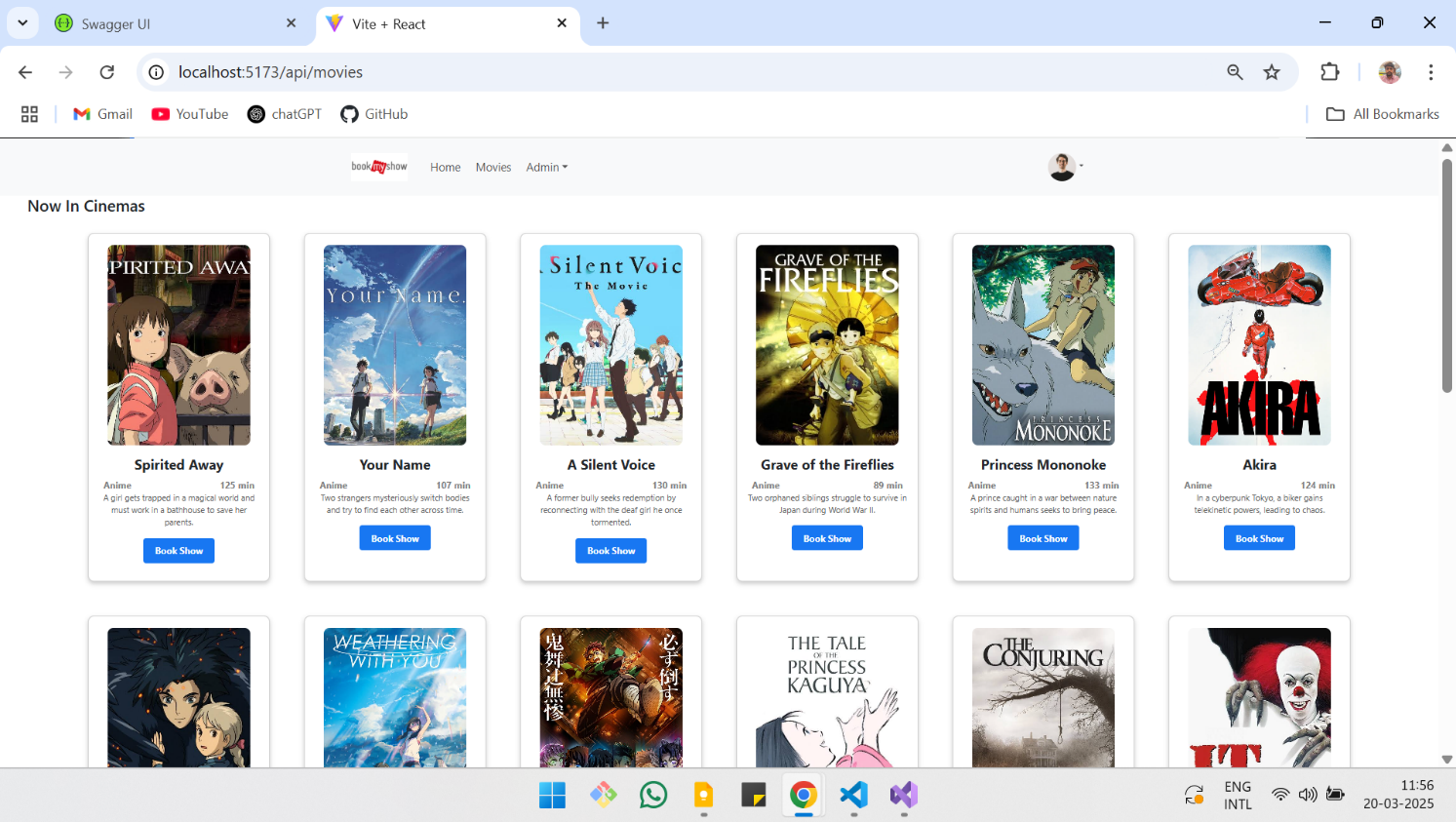
* Displays a carousel with featured movies.
* Shows movies categorized by genre.

* Clicking on a movie card navigates to its booking page.
* URL: /api/HomePage

****

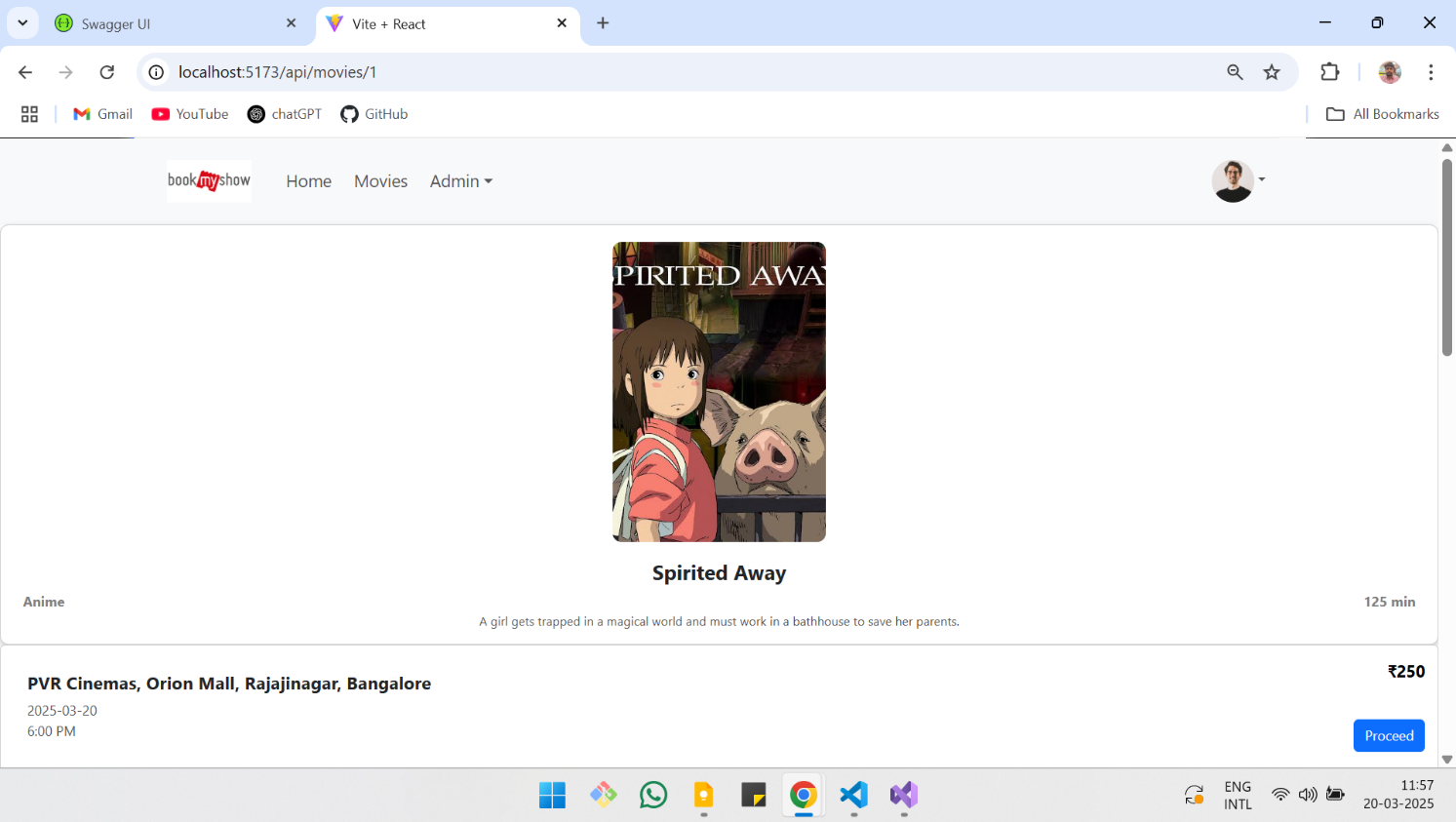
1. Movie Page

* Shows all the movie available.
* Click on Book Now button to book a ticket.
* URL: /api/movies



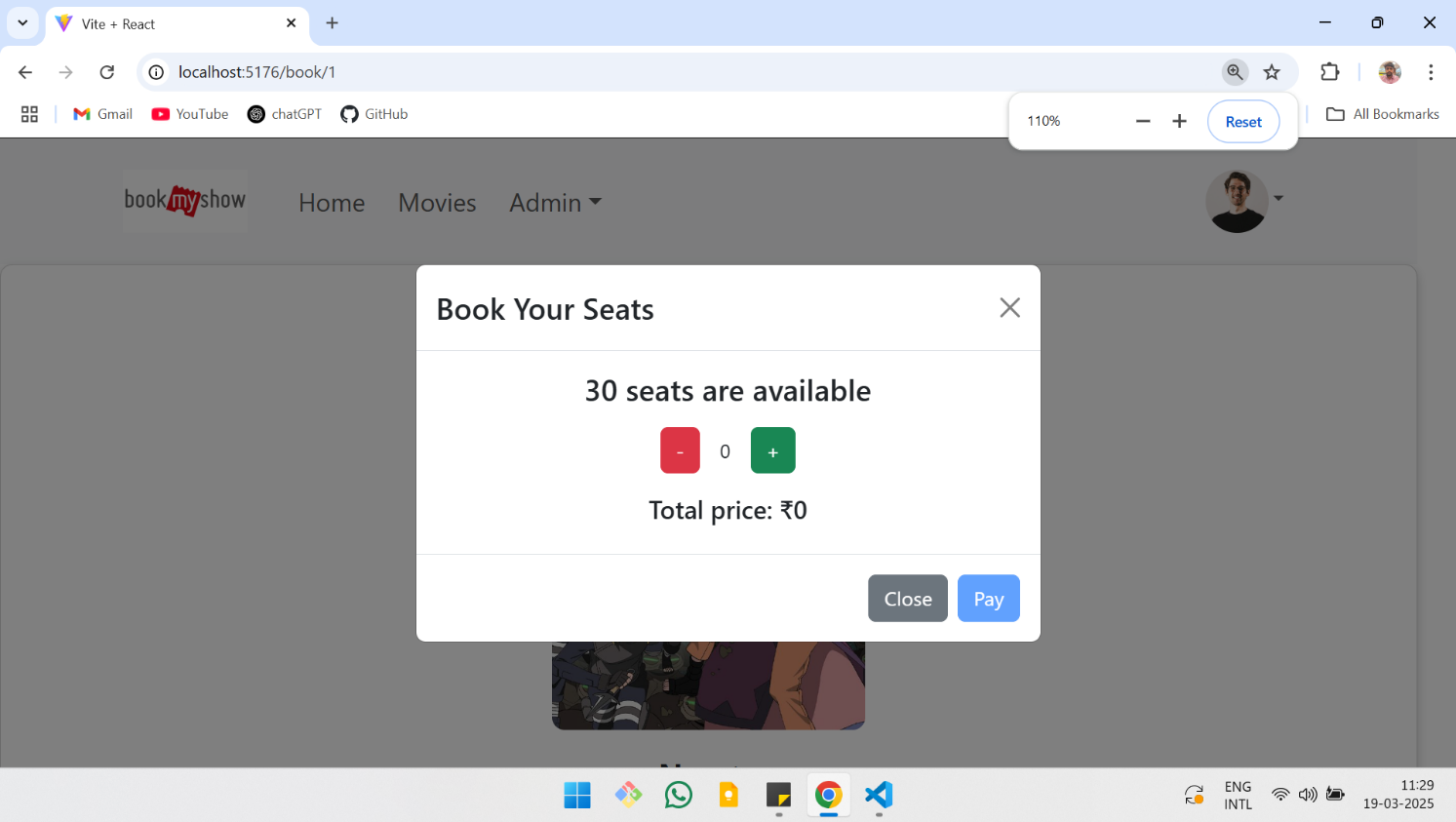
1. Show Booking Page

* Show the details about the movie.
* Shows all the available slots based on seat availability.
* Proceed Button to continue.
* URL: /api /book/{movieId}



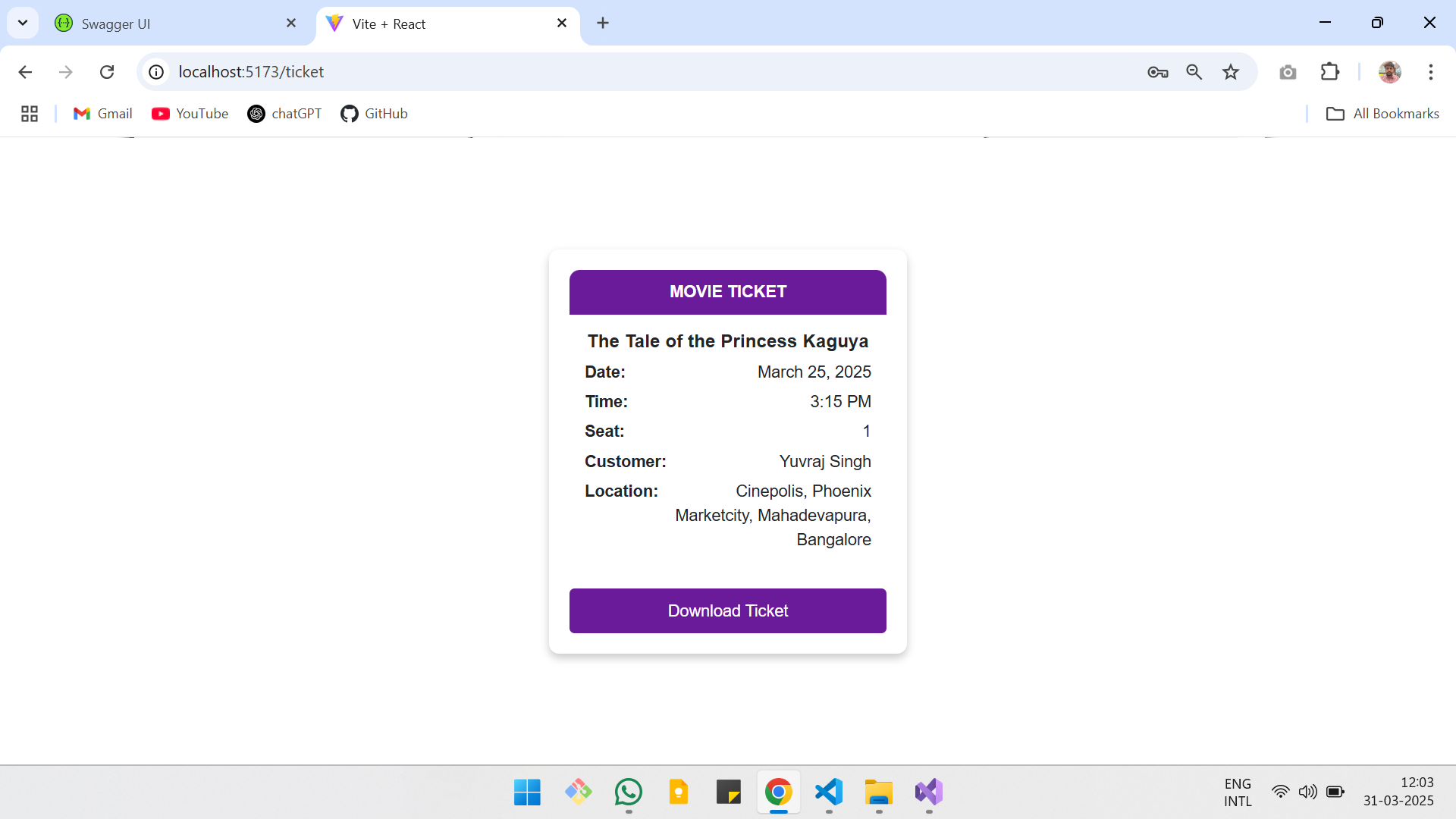
1. Seat Booking page

* A pop up to select number of tickets.
* User can’t select more than available number of tickets.
* Button to submit.
* URL: /api /book/{movieId}



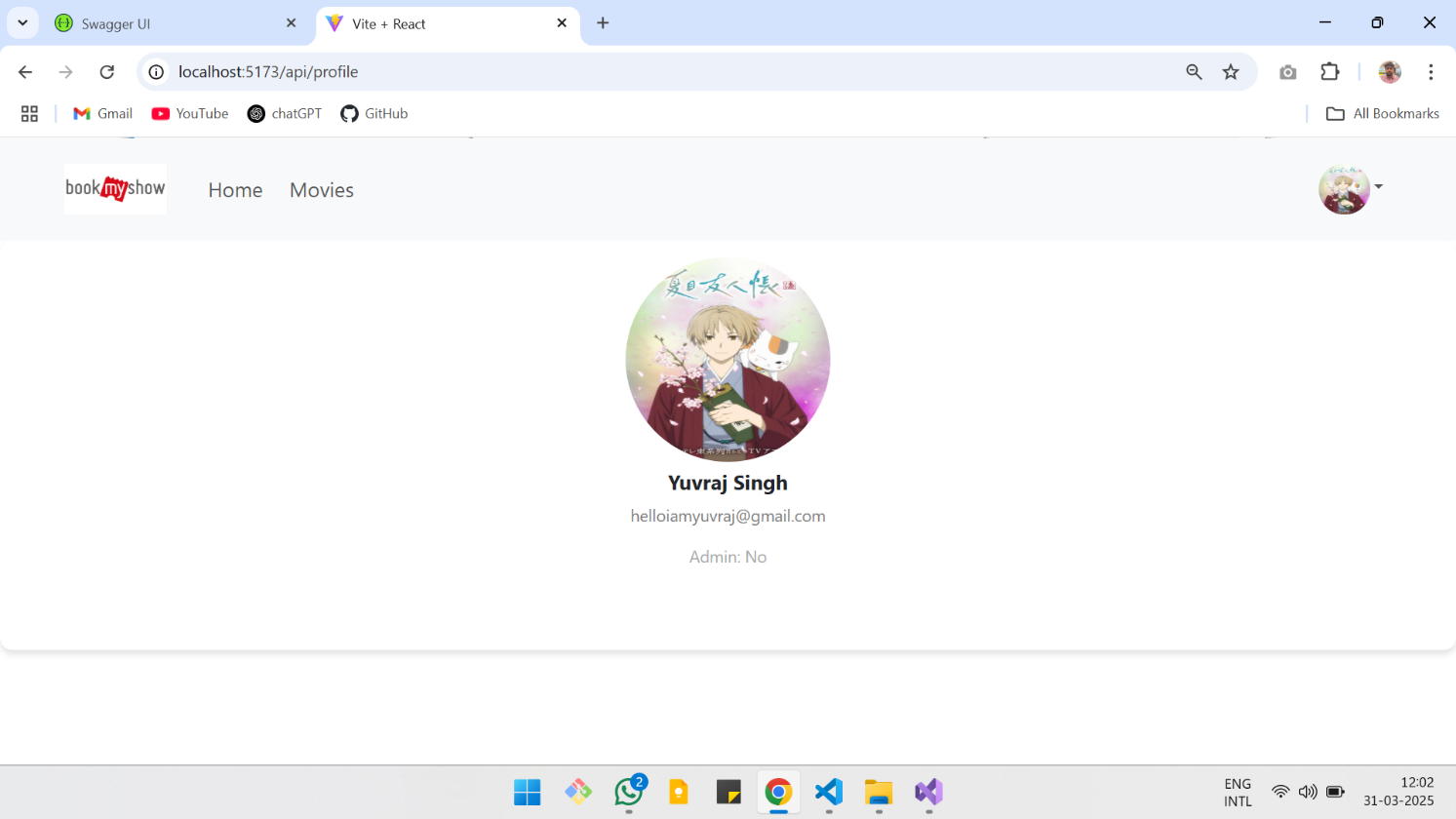
1. Ticket Confirmation

* Ticket Generation.
* URL: /api/ticket

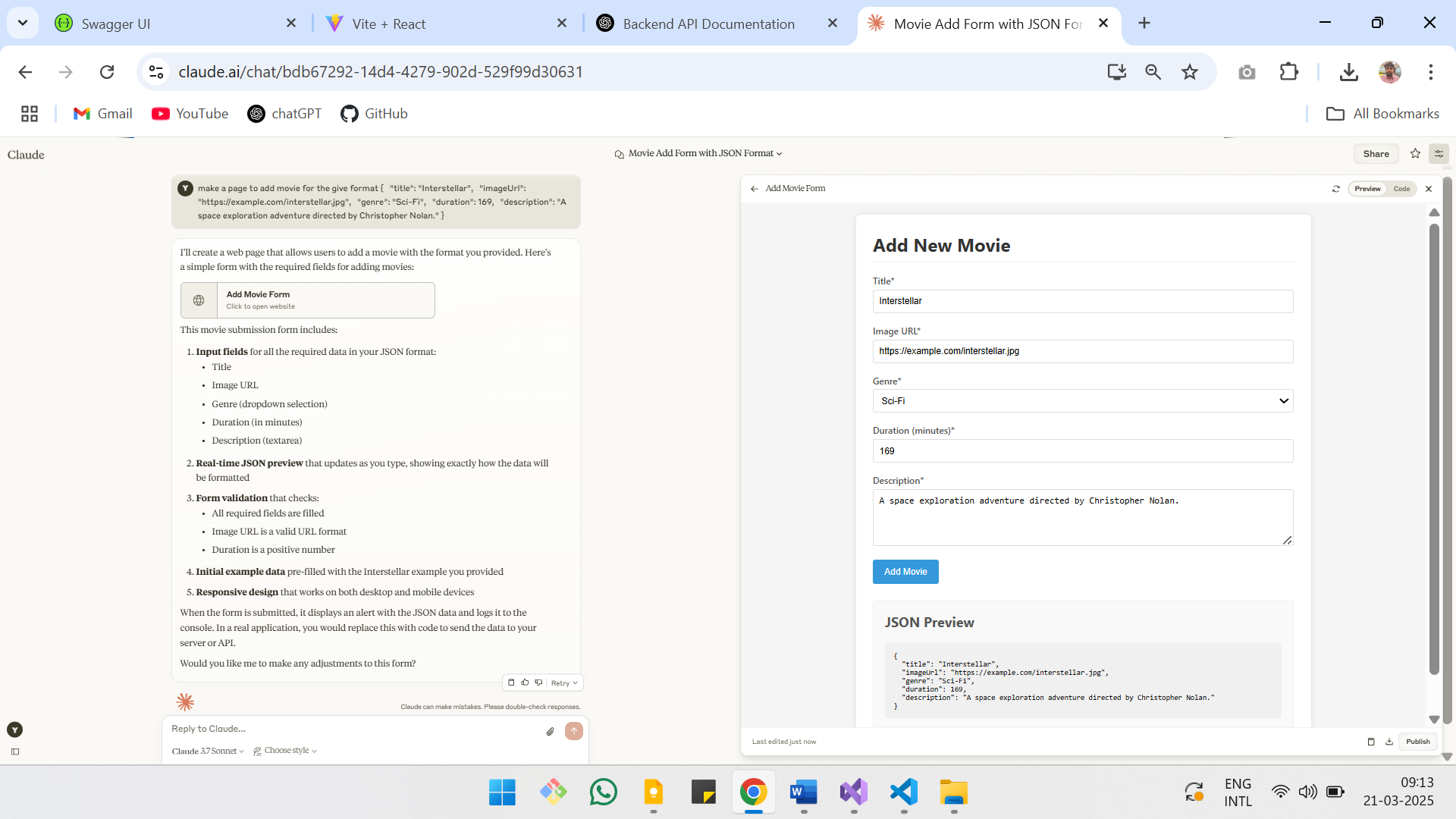


1. Profile page

* Check User details.
* URL: /api/profile

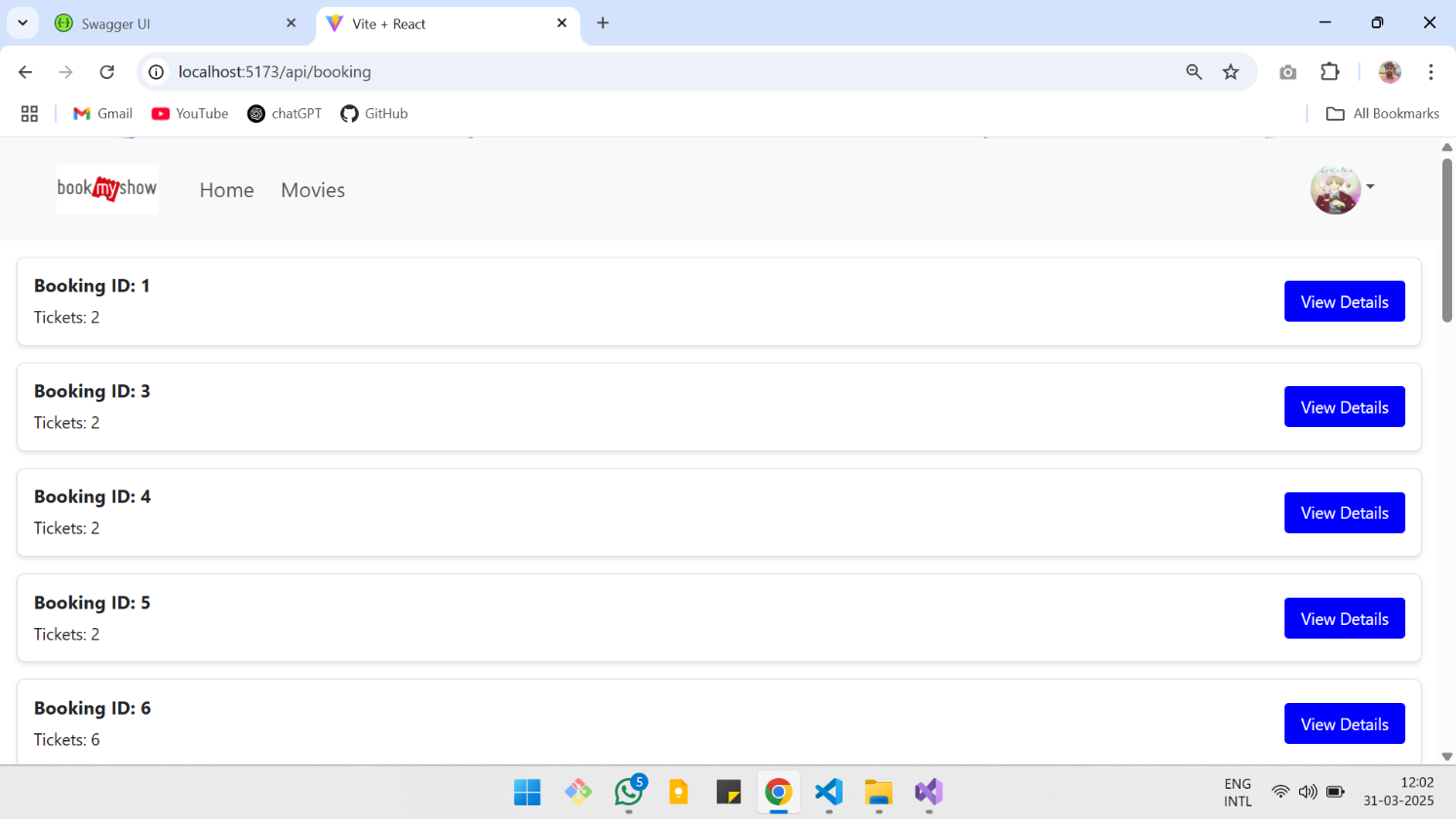


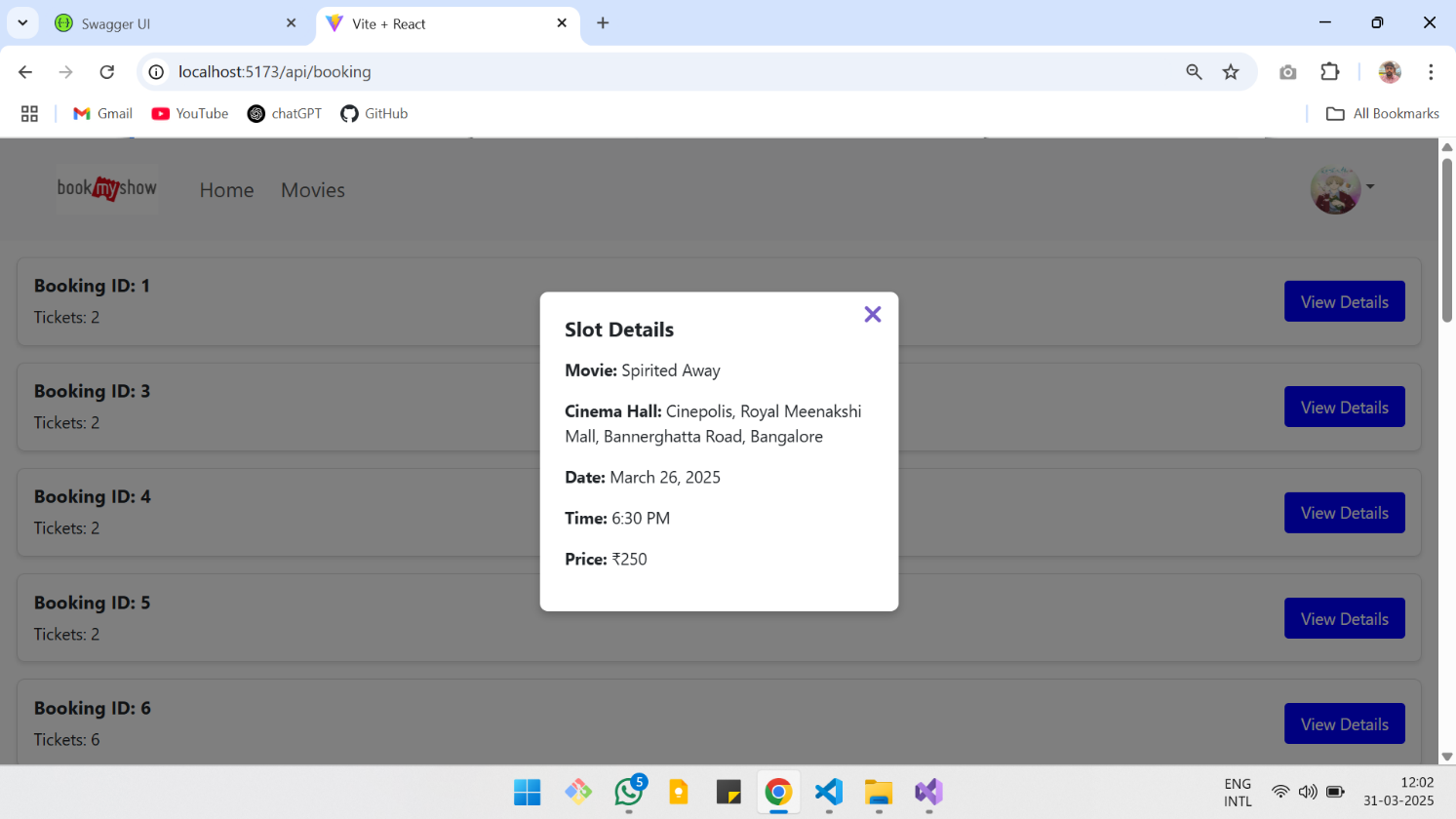
1. Add Movie
   * Add a movie.
   * Only admin has access
   * URL: /api/movie/add



1. Booking History

* Check booking history of user.
* When more details are required. Press the button to open more details.





**API Documentation**

(All endopints are secured with JWT)

1. Get All Movies

* Endpoint: GET /api/movies/
* Description: Returns a list of all available movies.
* Response Example:

[

{

"movieId": 1,

"title": "Inception",

"imageUrl": "https://example.com/inception.jpg",

"genre": "Sci-Fi",

"duration": 148,

"description": "A mind-bending thriller by Christopher Nolan."

},

{

"movieId": 2,

"title": "Spirited Away",

"imageUrl": "https://example.com/spiritedaway.jpg",

"genre": "Anime",

"duration": 125,

"description": "A Studio Ghibli masterpiece."

}

]

1. Get Movies by Genre

* Endpoint: GET /api/movie/?genre={genre}
* Description: Returns movies filtered by genre.
* Query Parameters:

genre (string) – Required. One of Anime, Horror, Action, RomCom.

* Example Request:  
  GET /api/movie/?genre=Sci-Fi
* Response Example:

[

{

"movieId": 1,

"title": "Inception",

"imageUrl": "https://example.com/inception.jpg",

"genre": "Sci-Fi",

"duration": 148,

"description": "A mind-bending thriller by Christopher Nolan."

}

]

1. Get Movie Details by ID

* Endpoint: GET /api/movie/:id
* Description: Fetches details of a specific movie by ID.
* Example Request:  
  GET /api/movie/1
* Response Example:

{

"movieId": 1,

"title": "Inception",

"imageUrl": "https://example.com/inception.jpg",

"genre": "Sci-Fi",

"duration": 148,

"description": "A mind-bending thriller by Christopher Nolan."

}

4. Get Available Slots for a Movie

* Endpoint: GET /api/movie/:id/slot/
* Description: Returns all available show slots for a given movie.
* Example Request:  
  GET /api/movie/1/slot/
* Response Example:

[

{

"showId": 101,

"movieId": 1,

"cinemaHall": "IMAX",

"date": "2025-03-25",

"timing": "18:30",

"price": 250,

"availableSeats": 50

},

{

"showId": 102,

"movieId": 1,

"cinemaHall": "PVR",

"date": "2025-03-26",

"timing": "20:00",

"price": 200,

"availableSeats": 30

}

]

5. Get Slot Details

* Endpoint: GET /api/slot/:id
* Description: Fetches details of a specific show slot by ID.
* Example Request:  
  GET /api/slot/101
* Response Example:

{

"showId": 101,

"movieId": 1,

"cinemaHall": "IMAX",

"date": "2025-03-25",

"timing": "18:30",

"price": 250,

"availableSeats": 50

}

6. Book a Ticket

* Endpoint: POST /api/slot/:id/book
* Description: Books tickets for a specific show and returns booking details.
* Request Body:

{

"userId": 5,

"tickets": 2

}

* Example Request:  
  POST /api/slot/101/book
* Response Example:

{

"bookingId": 1001,

"userId": 5,

"showId": 101,

"tickets": 2,

}

7. Add a Movie

* Endpoint**:** POST api/movie/add
* Description: Adds a new movie to the database.
* Request Body

{

"title": "Interstellar",

"imageUrl": "https://example.com/interstellar.jpg",

"genre": "Sci-Fi",

"duration": 169,

"description": "A space exploration adventure directed by Christopher Nolan."

}

* Response

{

"movieId": 3,

"title": "Interstellar",

"imageUrl": "https://example.com/interstellar.jpg",

"genre": "Sci-Fi",

"duration": 169,

"description": "A space exploration adventure directed by Christopher Nolan.",

"message": "Movie added successfully"

}

**Database Schema**

**Movies**

|  |  |  |
| --- | --- | --- |
| **Column** | **Type** | **Constraints** |
| movieId | INT | PRIMARY KEY |
| title | VARCHAR | NOT NULL |
| imageUrl | VARCHAR | NOT NULL |
| genre | VARCHAR | CHECK IN (Anime, Horror, Sci-Fi, RomCom) |
| duration | INT | NOT NULL |
| description | TEXT | NOT NULL |

**Shows**

|  |  |  |
| --- | --- | --- |
| **Column** | **Type** | **Constraints** |
| showId | INT | PRIMARY KEY |
| movieId | INT | FOREIGN KEY -> Movies |
| cinemaHall | VARCHAR | NOT NULL |
| date | DATE | NOT NULL |
| timing | VARCHAR | NOT NULL |
| price | INT | NOT NULL |
| availableSeats | INT | NOT NULL |

**Bookings**

|  |  |  |
| --- | --- | --- |
| **Column** | **Type** | **Constraints** |
| bookingId | INT | PRIMARY KEY |
| userId | INT | FOREIGN KEY -> Users |
| showId | INT | FOREIGN KEY -> Shows |
| tickets | INT | NOT NULL |

**User**

|  |  |  |
| --- | --- | --- |
| **Column** | **Type** | **Constraints** |
| userId | INT | PRIMARY KEY |
| name | VARCHAR | NOT NULL |
| email | VARCHAR | NOT NULL |
| password | VARCHAR | NOT NULL |
| picture | VARCHAR | NOT REQUIRED |
| isAdmin | BOOL | NOT NULL |

**ER Diagram**

