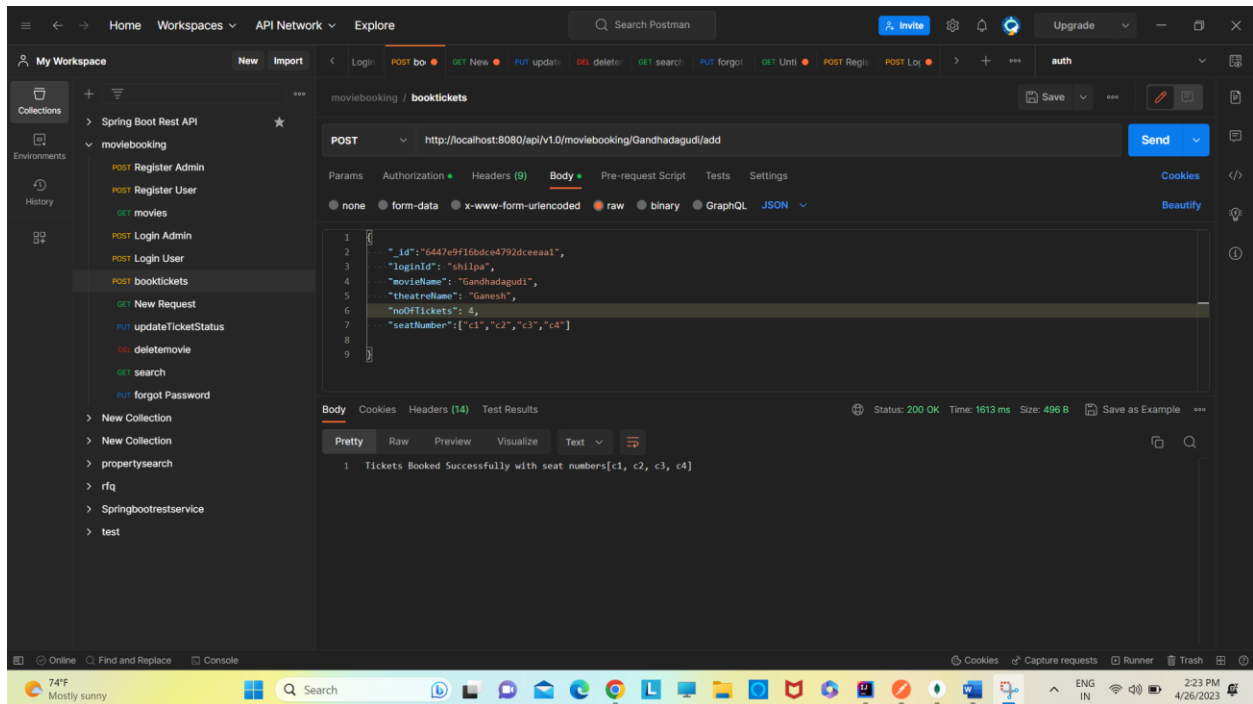


The screenshot displays the VS Code editor with a REST client extension. The left sidebar shows a project structure with folders like 'Spring Boot Rest API' and 'moviebooking'. The main editor area shows a REST client request for 'POST http://localhost:8080/api/v1.0/moviebooking/register'. The request body is a JSON object with fields like 'loginId', 'firstName', 'lastName', 'email', 'contactNumber', 'roles', and 'password'. The response is a 200 OK status with a JSON body containing a success message: 'message: "User registered successfully!"'. The bottom status bar shows the temperature as 74°F and the date as 4/26/2023.

The screenshot shows the VS Code interface with a REST client extension. The left sidebar displays a file explorer with a project named 'moviebooking' and a file named 'Login Admin'. The main editor area shows a REST client request and response. The request is a POST to 'http://localhost:8080/api/v1.0/moviebooking/login' with a JSON body containing 'loginId': 'anusha', 'password': 'anusha@123', and 'email': 'anusha@gmail.com'. The response is a JSON object with fields like 'timestamp', 'date', 'email', 'roles', 'username', 'token', and 'accessToken'.

User Booked tickets successfully



```
_id: ObjectId('6447e9f16bdce4792dceaa1')
movieName: "Gandhadagudi"
theatreName: "Ganesh"
noOfTicketsAvailable: 4
_class: "com.rbp.movieapp.models.Movie"
ticketsStatus: "BOOK ASAP"
```

Before Booking the tickets the noOfTicketsAvailable are 4

```
_id: ObjectId('6447e9f16bdce4792dceaa1')
movieName: "Gandhadagudi"
theatreName: "Ganesh"
noOfTicketsAvailable: 0
_class: "com.rbp.movieapp.models.Movie"
```

After Booking the tickets the noOfTicketsAvailable gets updates to 0

Booked tickets detail

Postman interface showing a GET request to `http://localhost:8080/api/v1.0/moviebooking/getallbookedtickets/Gandhadagudi` with a Bearer Token authorization. The response is a JSON object with the following structure:

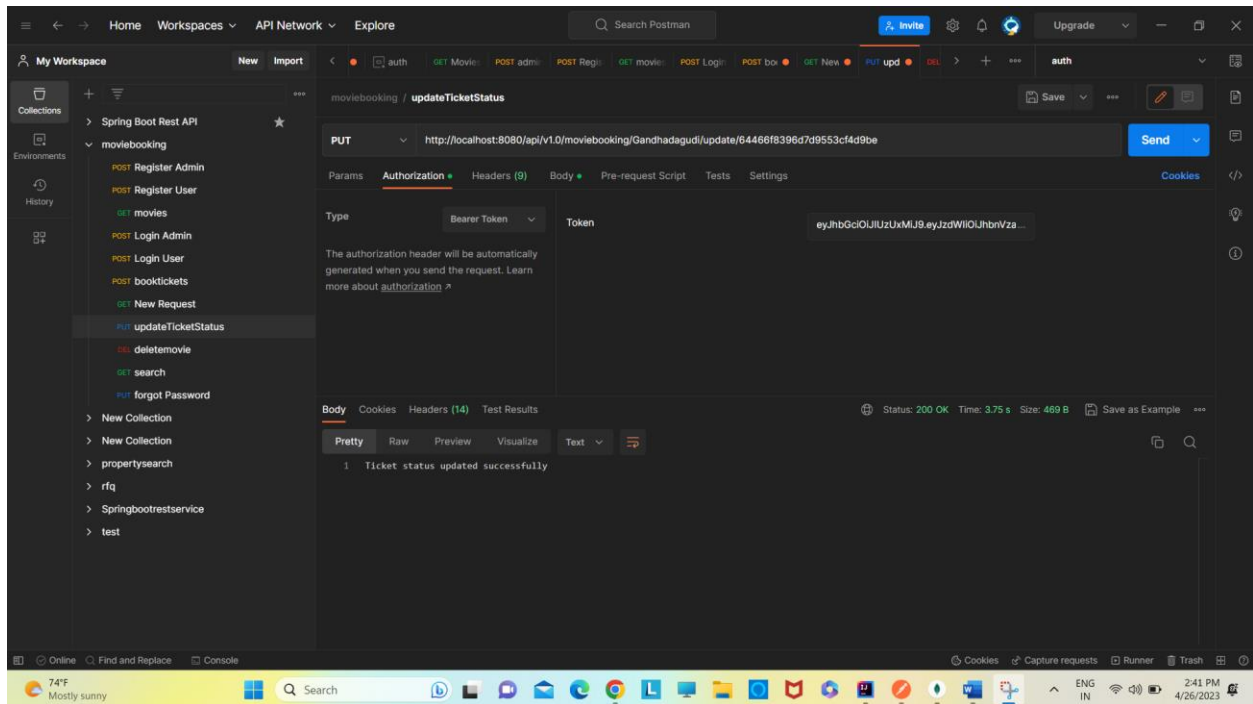
```
{  "id": 1,  "timestamp": 1682434545,  "date": "2023-04-25T14:55:45.000+00:00",  "loginId": "shilpa",  "movieName": "Gandhadagudi",  "theatreName": "Ganesh",  "noOfTickets": 4,  "seatNumber": [    "c1",    "c2",    "c3",    "c4"  ]}
```

All Released movies

Postman interface showing a GET request to `http://localhost:8080/api/v1.0/moviebooking/all` with No Auth authorization. The response is a JSON array of movie details:

```
[  {    "movieName": "Kantara",    "theatreName": "Vaibhav",    "noOfTicketsAvailable": 118,    "ticketsStatus": "BOOK ASAP"  },  {    "id": 2,    "timestamp": 1682337667,    "date": "2023-04-24T12:01:07.000+00:00",    "movieName": "Dada",    "theatreName": "PVR",    "noOfTicketsAvailable": 122,    "ticketsStatus": "Book ASAP"  },  {    "id": 3,    "timestamp": 1682434545,    "date": "2023-04-25T14:55:45.000+00:00",    "movieName": "Gandhadagudi",    "theatreName": "Ganesh",  }]
```

Admin has updated the ticketStatus



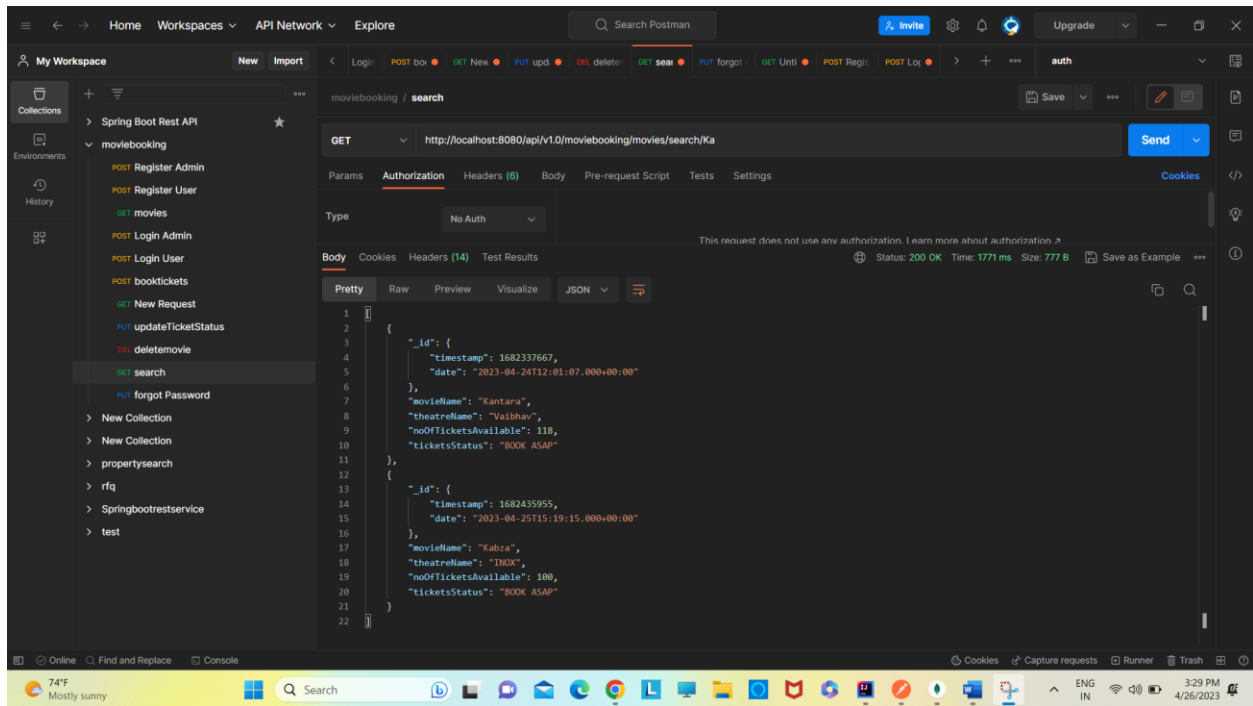
```
_id: ObjectId('6447e9f16bdce4792dceaa1')
movieName: "Gandhadagudi"
theatreName: "Ganesh"
noOfTicketsAvailable: 0
_class: "com.rbp.movieapp.models.Movie"
```

Before updating Ticket Status

```
_id: ObjectId('6447e9f16bdce4792dceaa1')
movieName: "Gandhadagudi"
theatreName: "Ganesh"
noOfTicketsAvailable: 0
ticketsStatus: "SOLD OUT"
_class: "com.rbp.movieapp.models.Movie"
```

After updating Ticket Status

Search movieName also supports partial movie name search



Complete Name Search

