UNIVERSITY OF WESTMINSTER

INFORMATICS INSTITUTE OF TECHNOLOGY

Informatics Institute of Technology (IIT Sri Lanka)

Object-Oriented Programming

5COSC019C.1

Event-Hive Ticket System

Name - M. Jude Maria Bevan
UoW no - w1959430
IIT no - 20223075

1. Table of Contents

2. Introduction

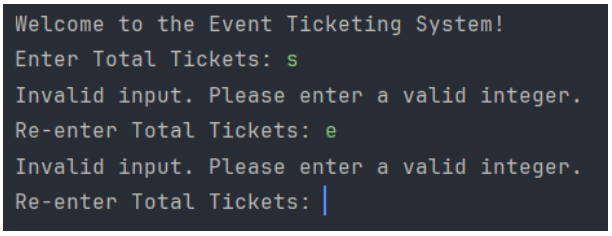
An **Event Ticketing System** is a digital platform enabling efficient event management and ticket sales. It streamlines tasks like online ticketing, and secure access control using threads, ensuring a seamless experience for producers and consumers. This system enhances efficiency, reduces errors, and provides real-time insights, catering to events of all scales.

3. Important Links

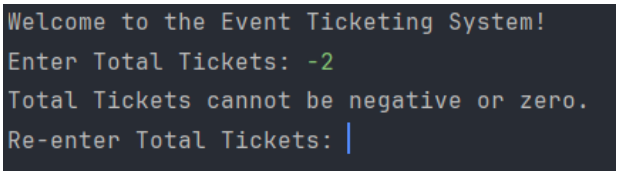
3.1. GitHub repository - https://github.com/judebevan/event-ticketing-system
3.2. API documentation - Event-Hive API collection
3.3. Swagger (localhost) - swagger
3.4. YouTube link - event-system-YouTube

4. Test Cases for Event-System CLI

4.1. String Inputs



4.2. Negative integers

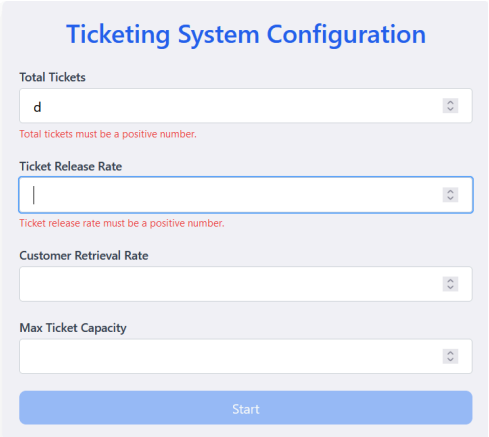


4.3. Below, test cases are tested using my CLI but cannot add screenshot proofs in this document, it contains large amount of logs.

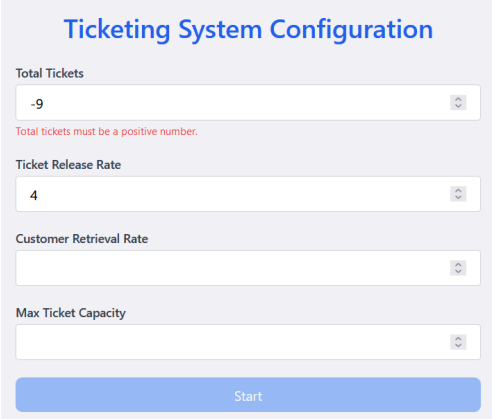
Test Case	Total Tickets	Ticket Release Rate (seconds)	Customer Retrieval Rate (seconds)	Max Ticket Capacity	Vendor Count	Customer Count	Actual Outcome
1	10	1	1	5	2	2	Matches expected. Balanced production and consumption.
2	15	2	1	5	3	1	Matches expected. Customers waited.
3	20	1	3	10	2	2	Matches expected. Vendors paused at full capacity.
4	25	2	2	10	1	1	Matches expected. Handled concurrent operations.
5	0	1	1	10	2	2	Matches expected. Cannot input 0 values
6	10	1	1	10	3	3	Matches expected. No race conditions.
7	100	1	2	20	1	4	Matches expected. Handled high load effectively.

5. Test Cases for Event-Hive configurations

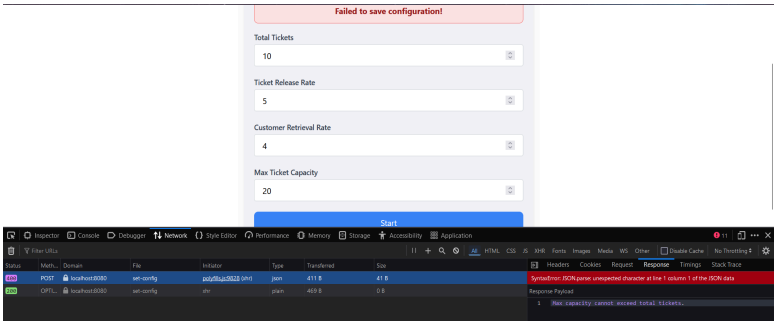
5.1. String inputs



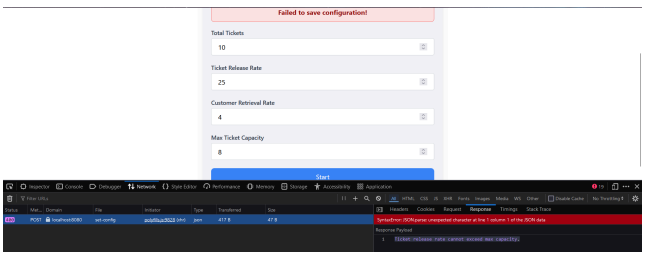
4.2. negative integer inputs



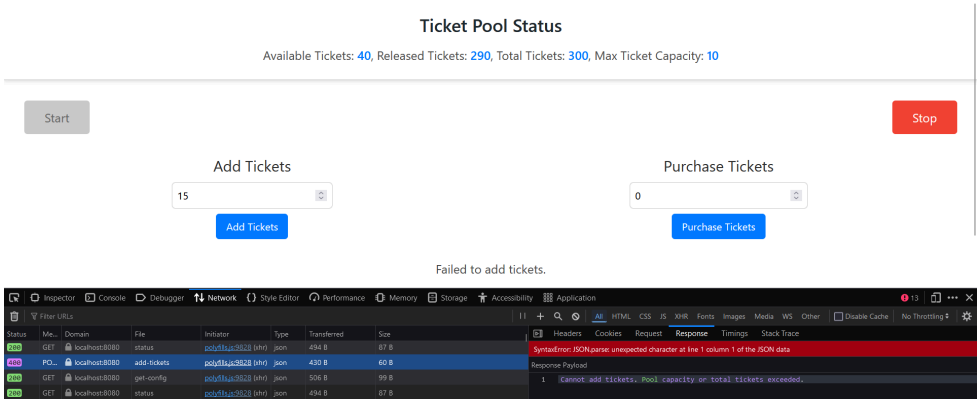
4.3. Max Ticket Capacity should not be grater than Total Tickets



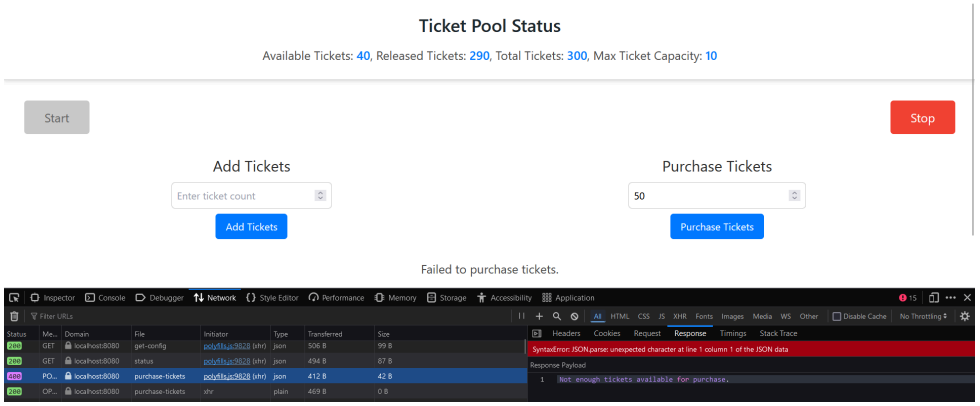
4.4. Ticket release rate should not exceed max capacity.



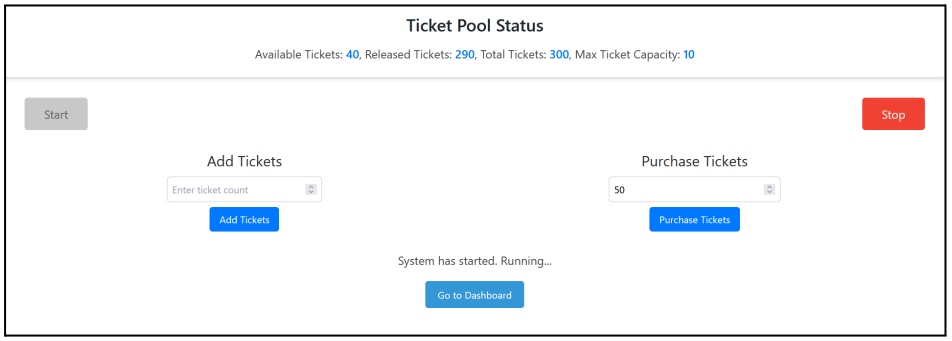
4.5. Vendor cannot add tickets more than the Max Ticket Capacity, Total Tickets.

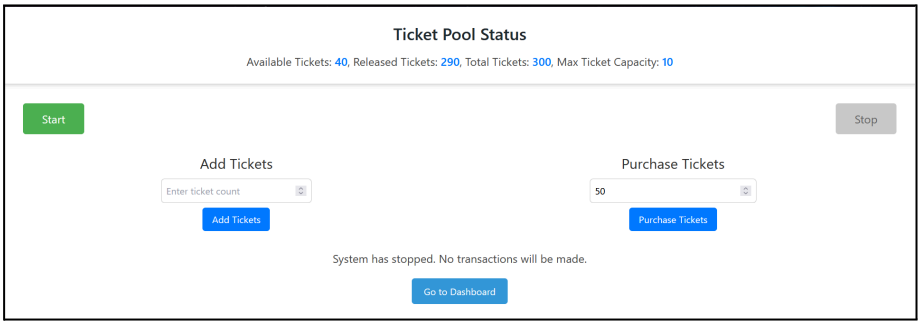


4.6. Customer cannot purchase tickets more than the available tickets in the pool.



4.7. Once event started, stop button will be enabled and start button will remain disabled, and when we click stop button the start button will be disabled.





6. API Endpoints

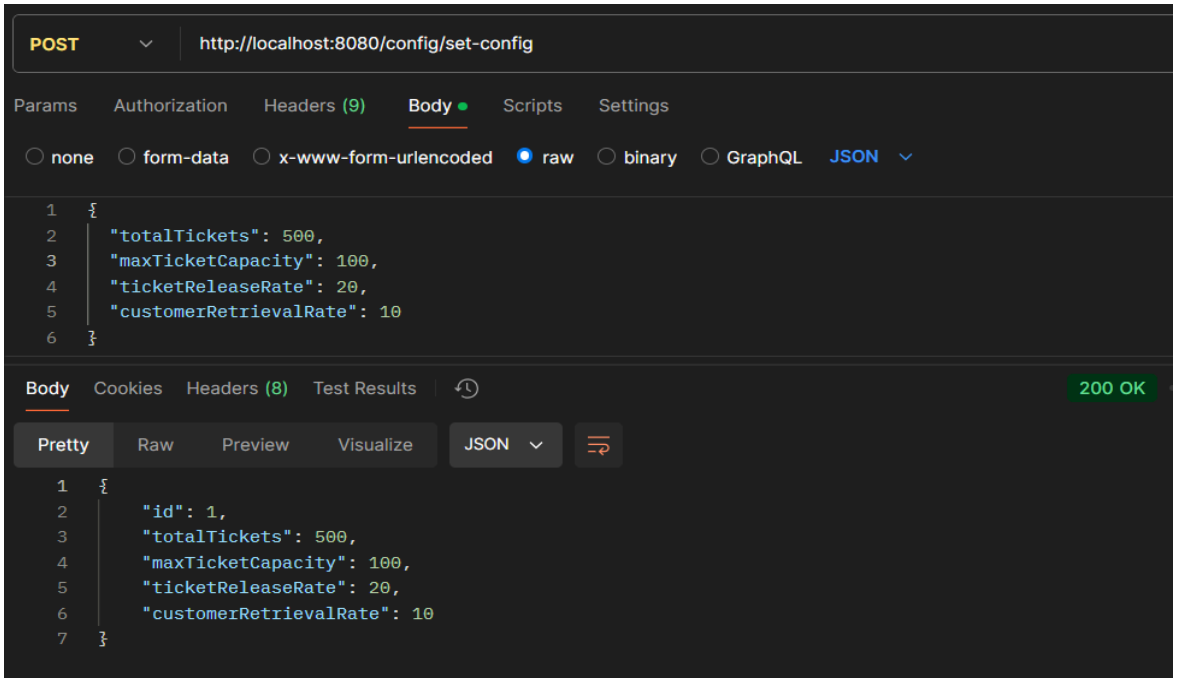
5.1 Configs

Method - POST set-config

Endpoint - http://localhost:8080/config/set-config

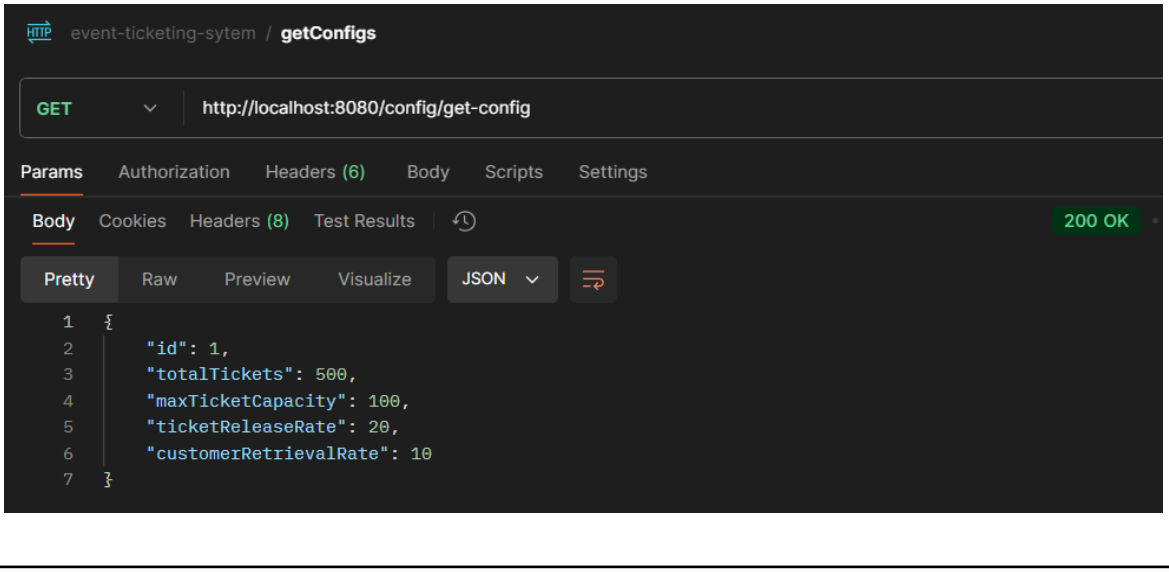
Request body:

```
{
 "totalTickets": 500,
 "maxTicketCapacity": 100,
 "ticketReleaseRate": 20,
 "customerRetrievalRate": 10
}
```



Method - GET get-config

Endpoint - http://localhost:8080/config/get-config



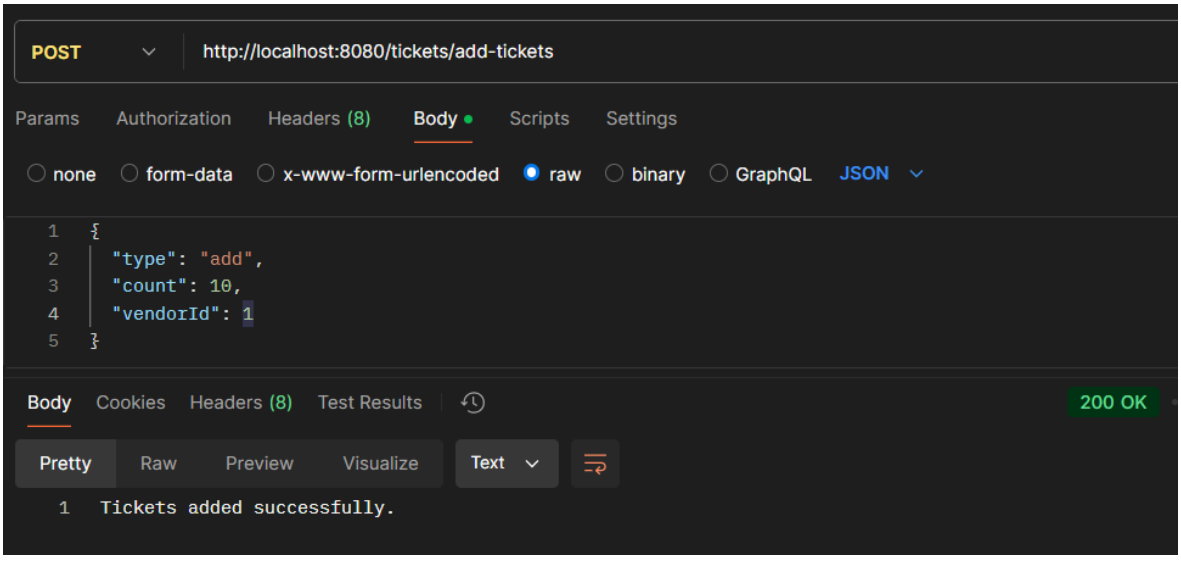
5.2 Tickets

Method - POST add-tickets

Endpoint - http://localhost:8080/tickets/add-tickets

Request body:

```
{
 "type": "add",
 "count": 10,
 "vendorId": 1
}
```

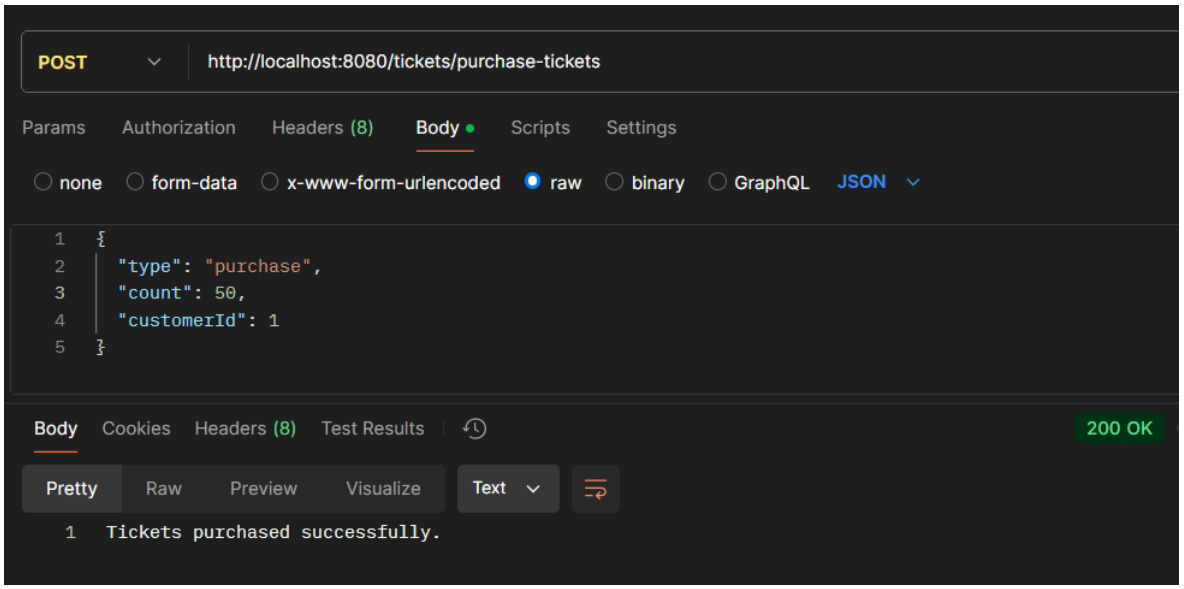


Method - POST purchase-tickets

Endpoint - http://localhost:8080/tickets/purchase-tickets

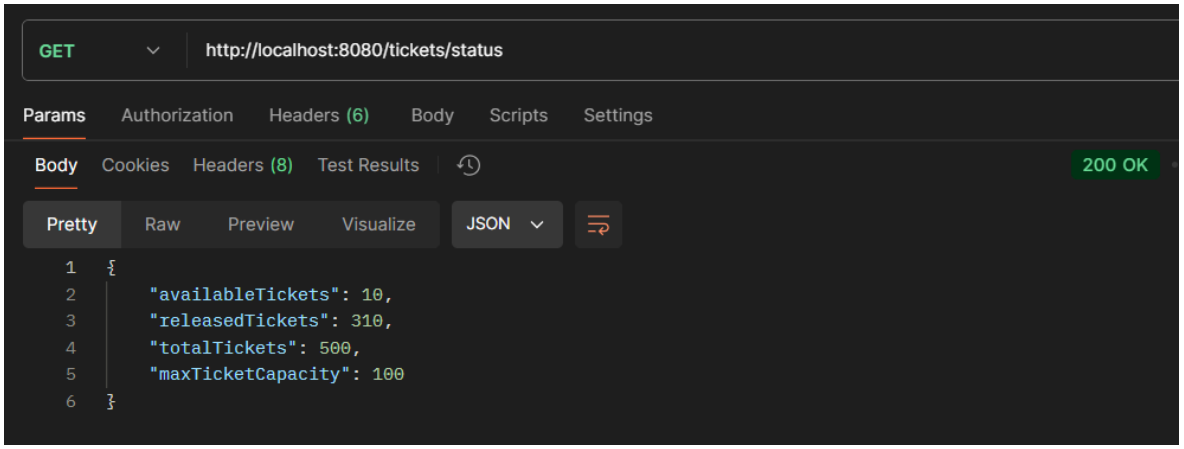
Request body:

```
{
 "type": "purchase",
 "count": 50,
 "customerId": 1
}
```



Method - GET status

Endpoint - http://localhost:8080/tickets/status



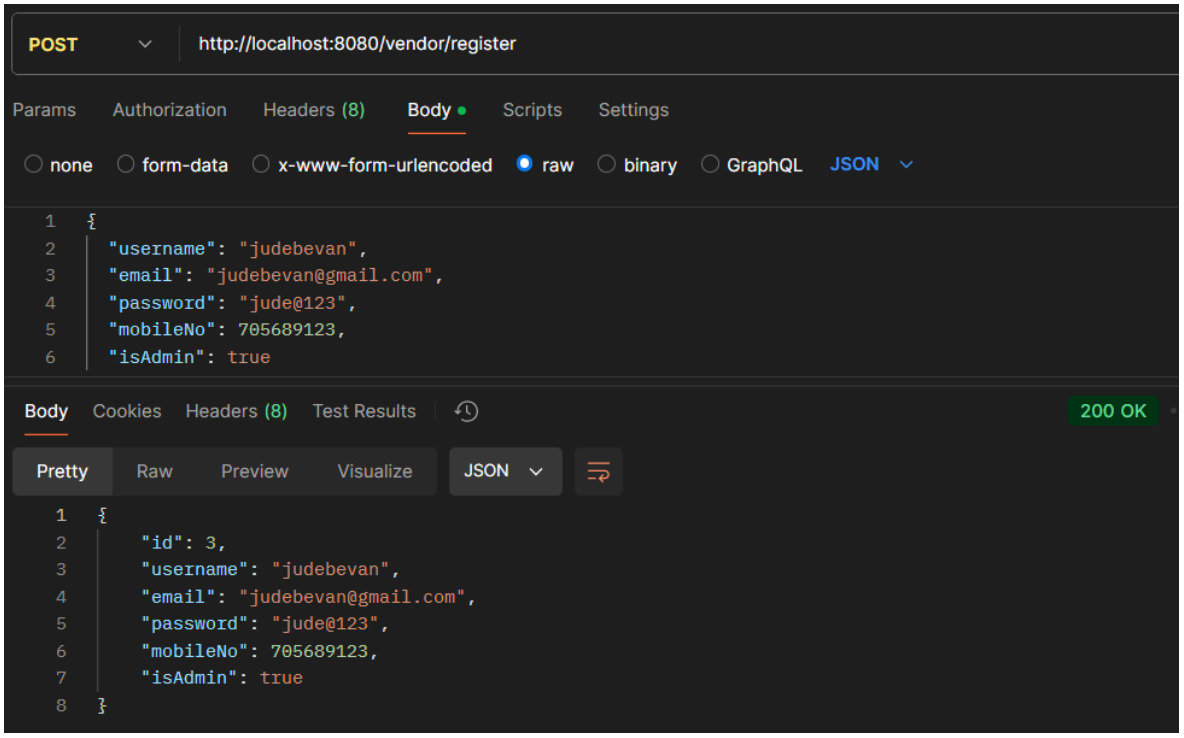
5.3 Login, Registration

Method - POST vendor-register

Endpoint - http://localhost:8080/vendor/register

Request body:

```
{
 "username": "judebevan",
 "email": "judebevan@gmail.com",
 "password": "jude@123",
 "mobileNo": 705689123,
 "isAdmin": true
}
```

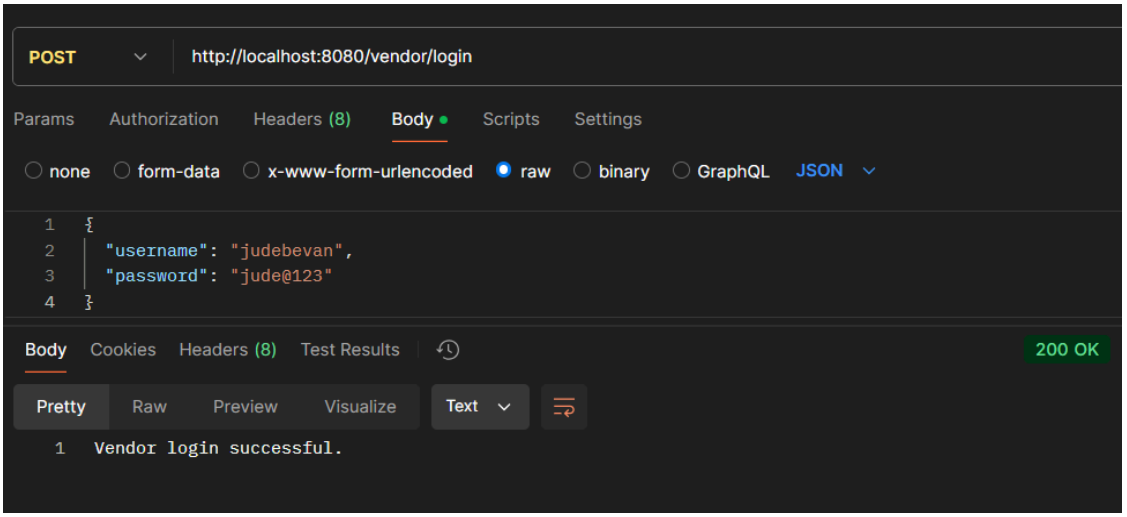


Method - POST vendor-login

Endpoint - http://localhost:8080/vendor/login

Request body:

```
{
 "username": "judebevan",
 "password": "jude@123"
}
```

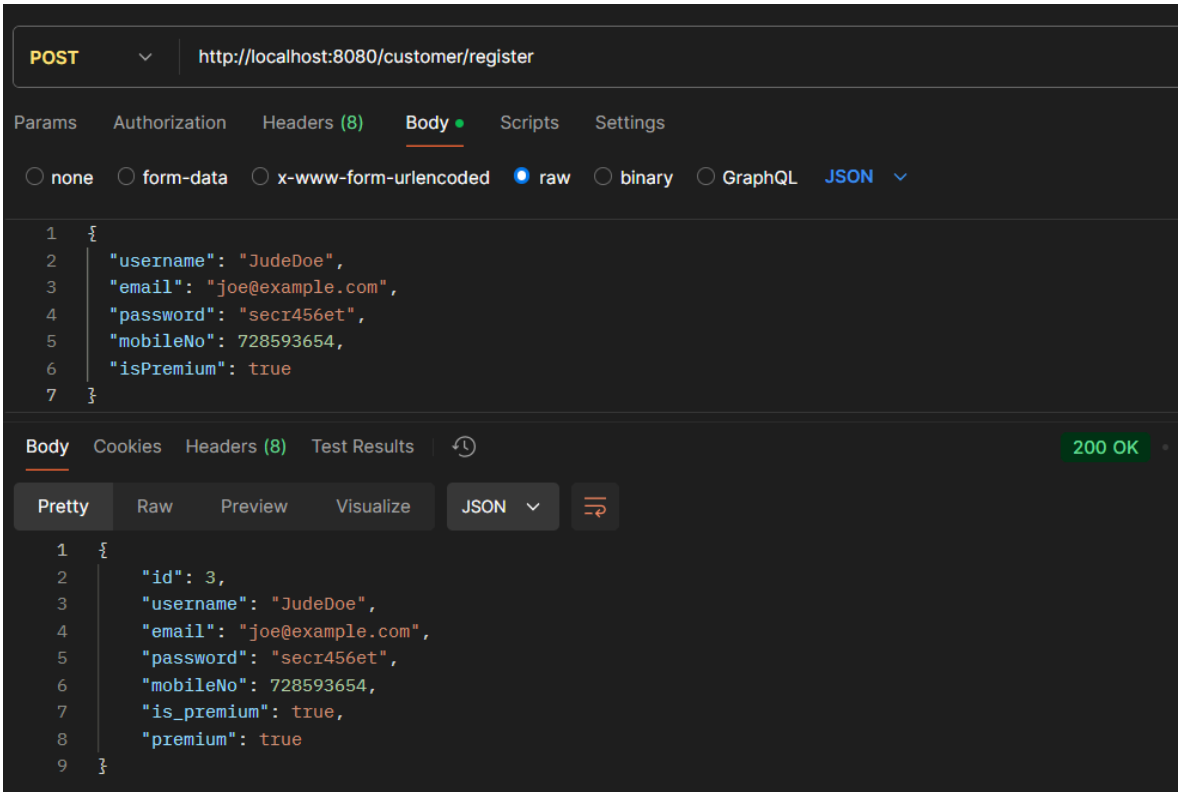


Method - POST customer-register

Endpoint - http://localhost:8080/customer/register

Request body:

```
{
 "username": "JudeDoe",
 "email": "joe@example.com",
 "password": "secr456et",
 "mobileNo": 728593654,
 "isPremium": true
}
```

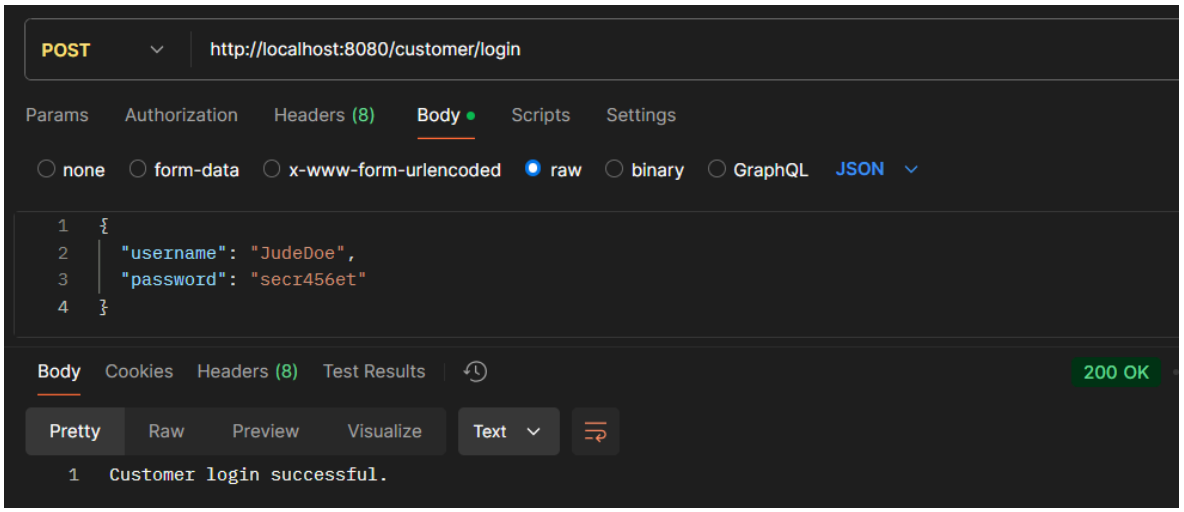


Method - POST customer-login

Endpoint - http://localhost:8080/customer/login

Request body:

```
{
 "username": "JudeDoe",
 "password": "secr456et"
}
```



7. Conclusion

The Event Ticketing System successfully demonstrates its ability to streamline event management and ticket sales through efficient automation and robust handling of various scenarios. The system's key features, such as real-time ticket production and consumption, concurrent vendor-customer operations, and edge case handling, have been rigorously tested and validated.

By providing a user-friendly and scalable solution, the system eliminates traditional inefficiencies and enhances the overall event management experience. It is capable of handling high loads, ensuring data consistency, and meeting modern event ticketing requirements. This project highlights the importance of integrating technology to optimize event workflows, making it a valuable tool for organizers and attendees alike.