

NAME : MOHANSUNDAR.K

REG.NO : 20MIS1037

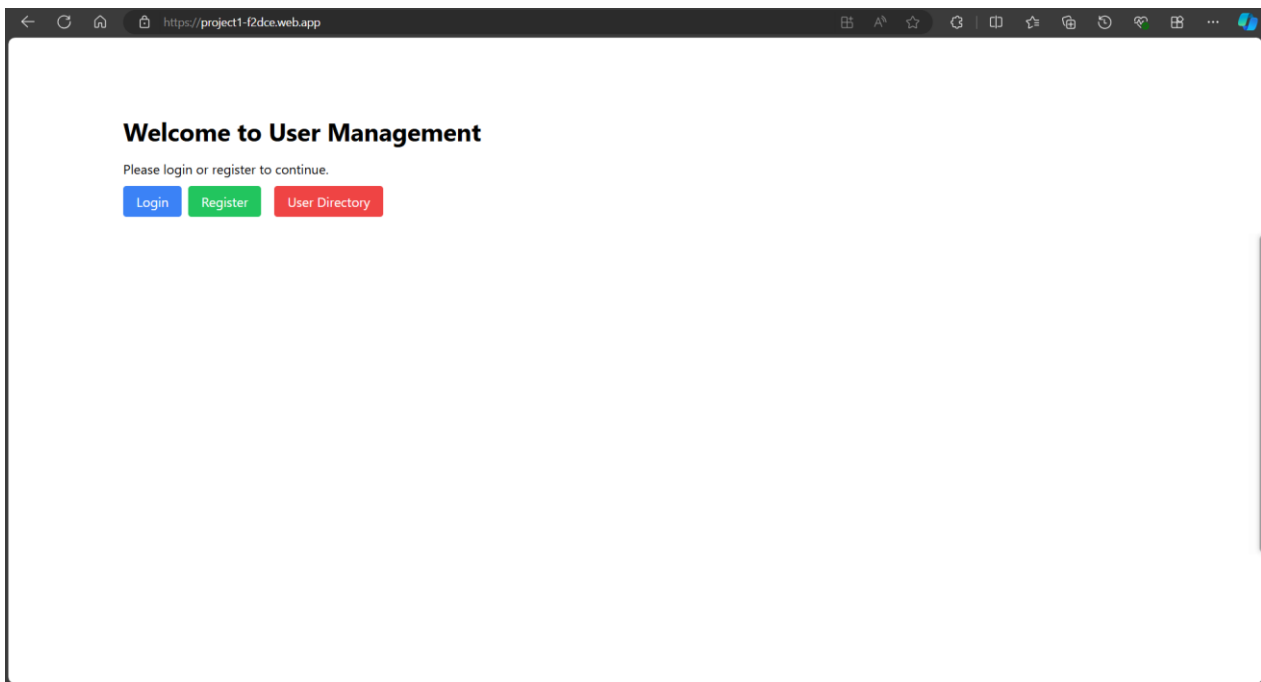
Resume Link : tinyurl.com/20mis1037

1. User Management System Assessment

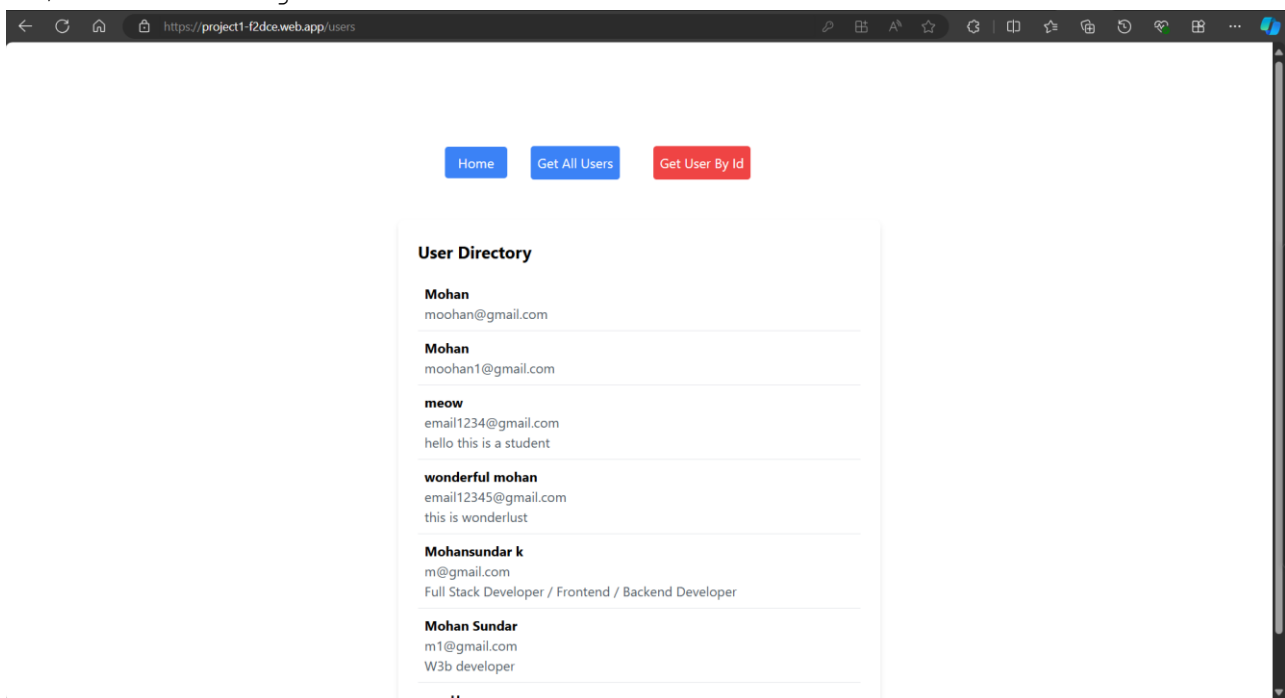
Github Links

- [Frontend](#)
- [Backend](#)
- [Live Demo](#)

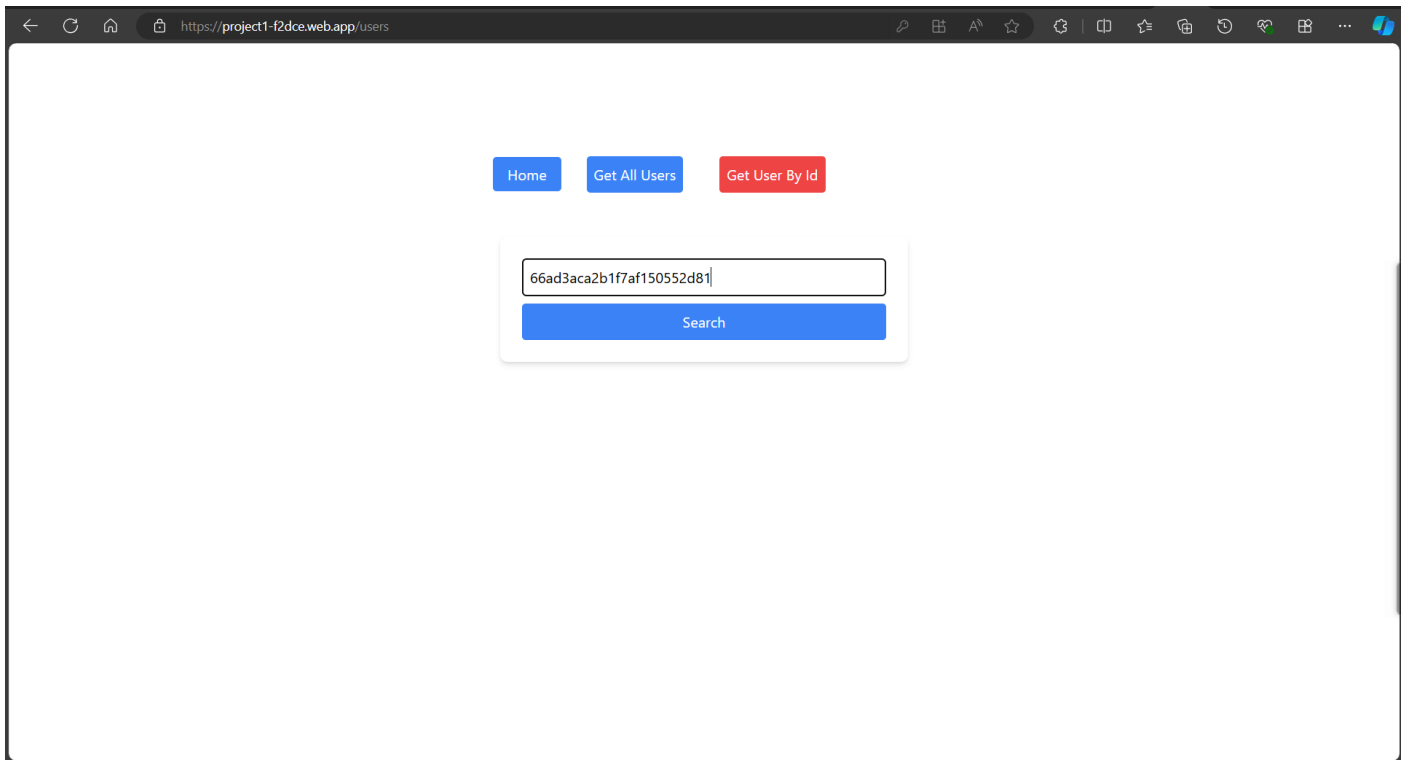
/Home Page



/Get Users Page

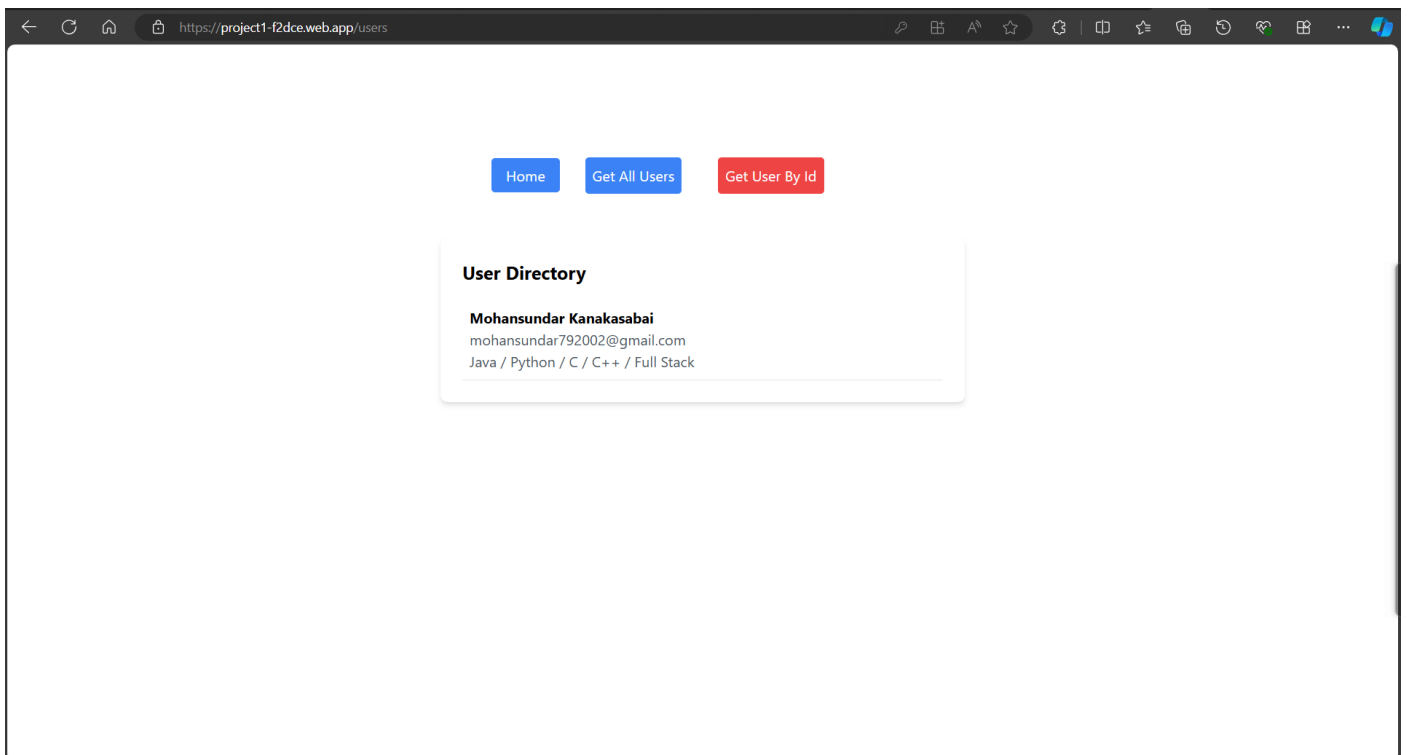


/Get User By ID Page



A screenshot of a web browser showing the URL `https://project1-f2dce.web.app/users`. The page has a dark header bar with navigation icons. Below the header, there are three buttons: "Home" (blue), "Get All Users" (blue), and "Get User By Id" (red). In the center, there is a white search box containing the text "66ad3aca2b1f7af150552d81". Below the search box is a blue "Search" button.

/Get User by Id (After Searching)

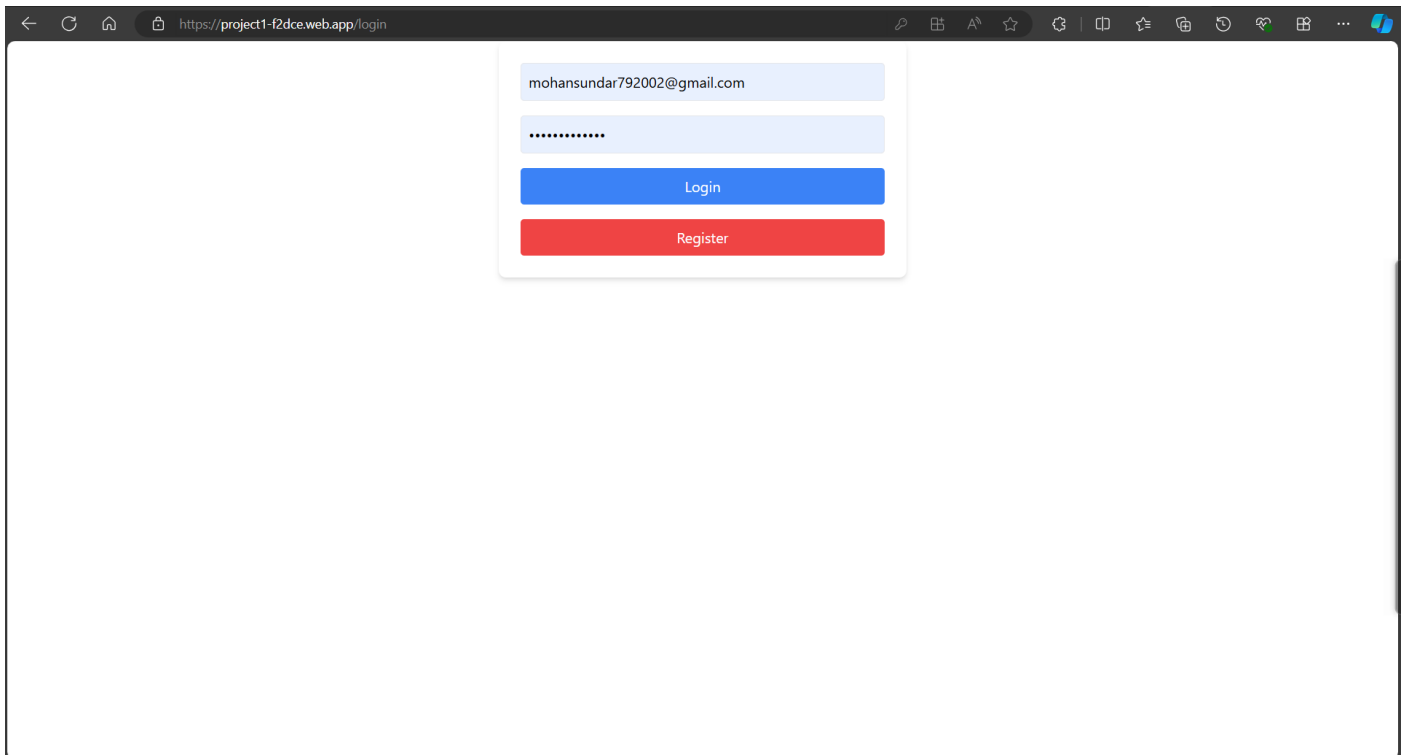


A screenshot of the same web browser showing the same URL. The page layout is identical to the previous screenshot, but the search results are displayed. The "Get User By Id" button is now highlighted in red. Below the search box, a white box titled "User Directory" contains the following information:

User Directory

Mohansundar Kanakasabai
mohansundar792002@gmail.com
Java / Python / C / C++ / Full Stack

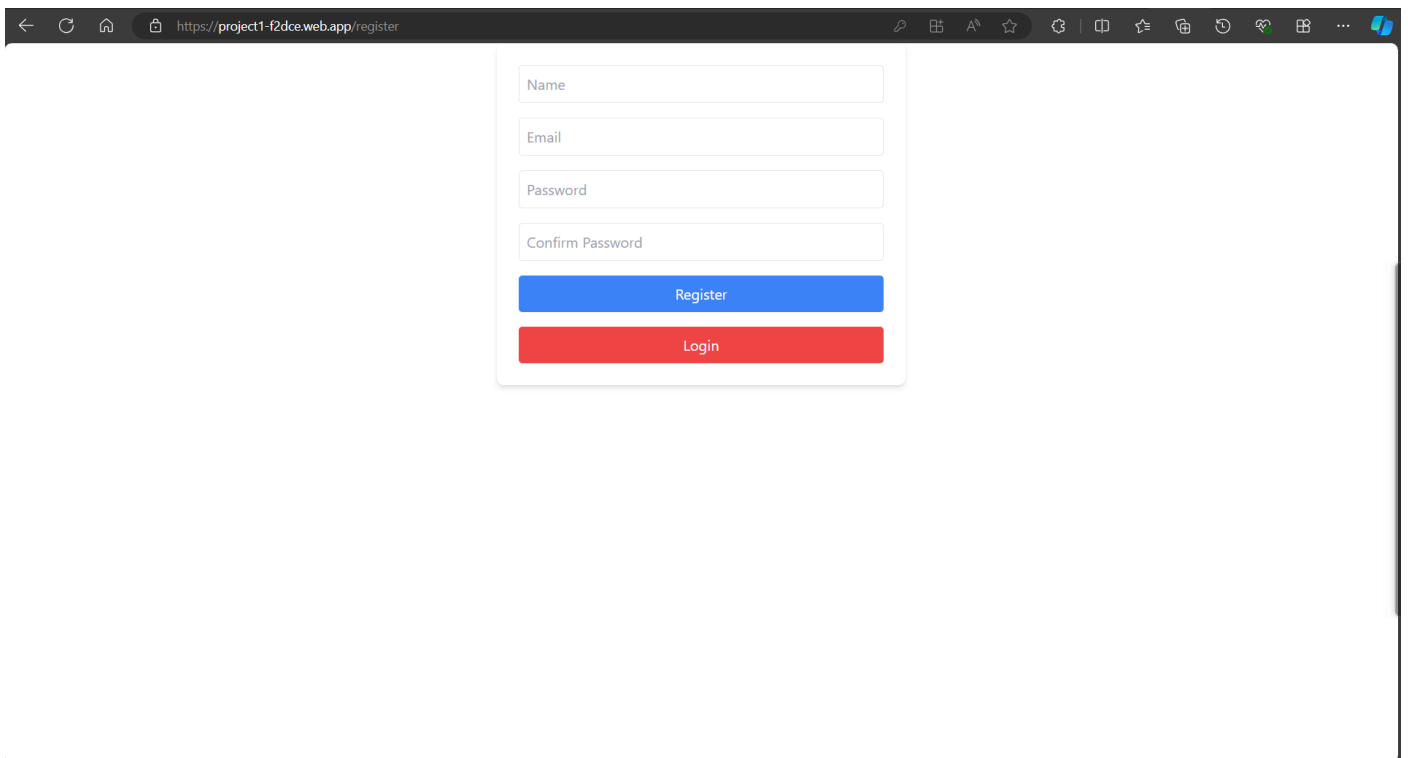
/Login Page



A screenshot of a web browser displaying the login page. The browser's address bar shows the URL `https://project1-f2dce.web.app/login`. The login form is centered on the page and contains the following elements:

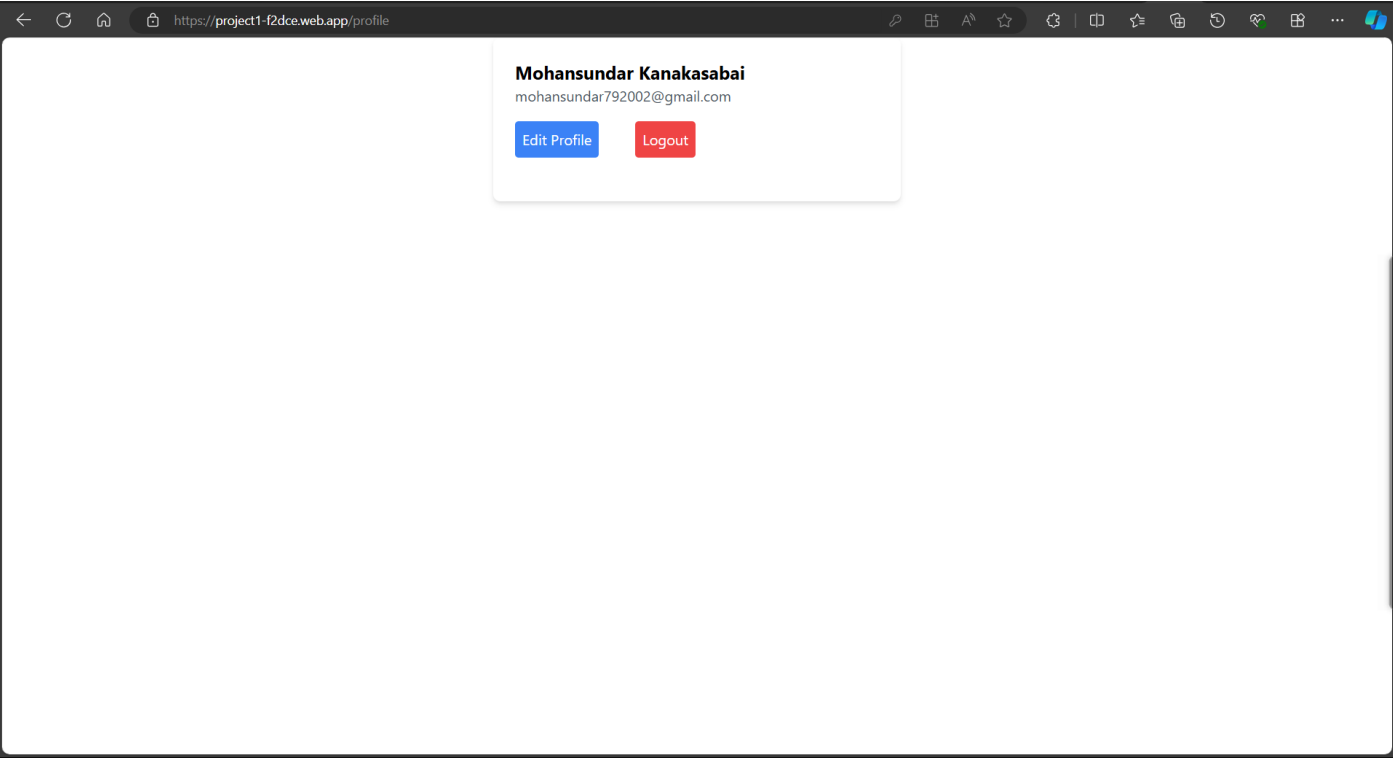
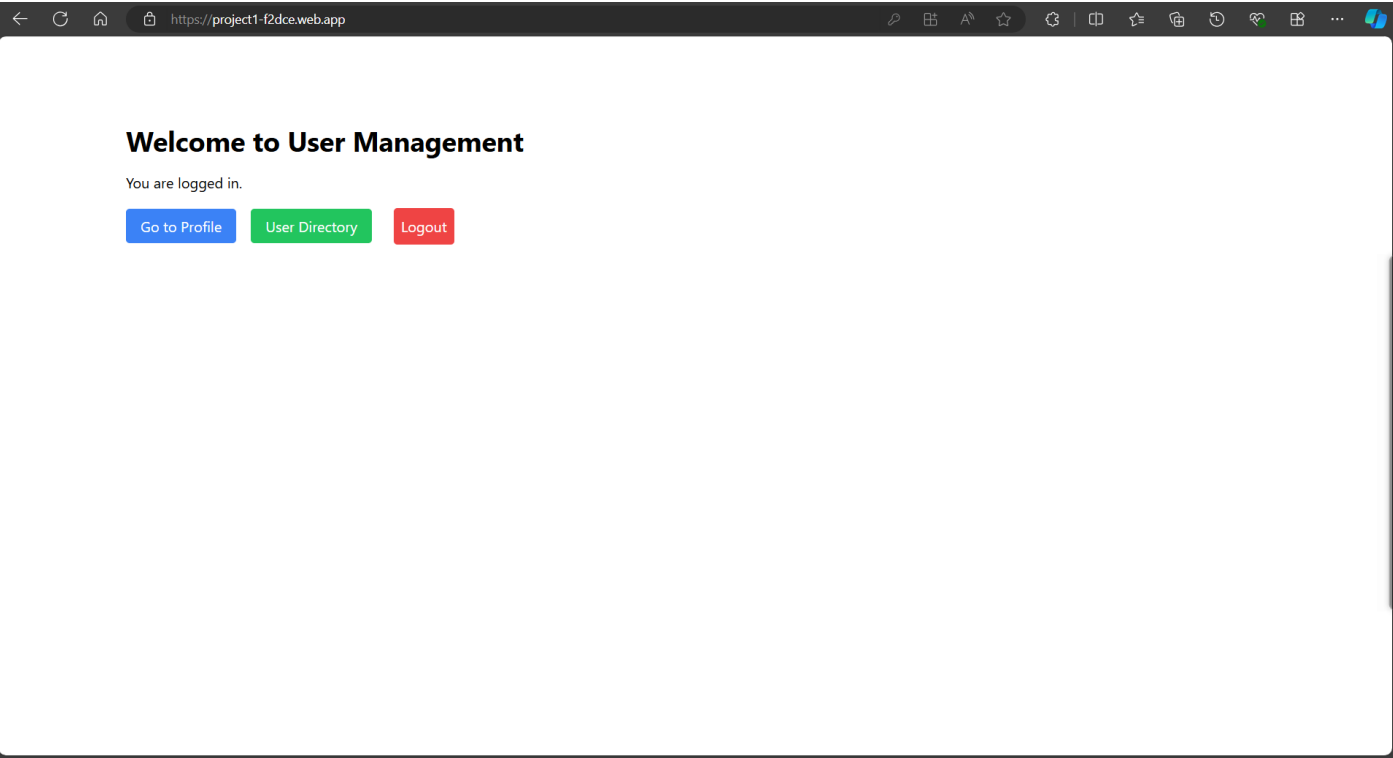
- An email input field with the text `mohansundar792002@gmail.com`.
- A password input field with masked characters (dots).
- A blue button labeled "Login".
- A red button labeled "Register".

/Register Page

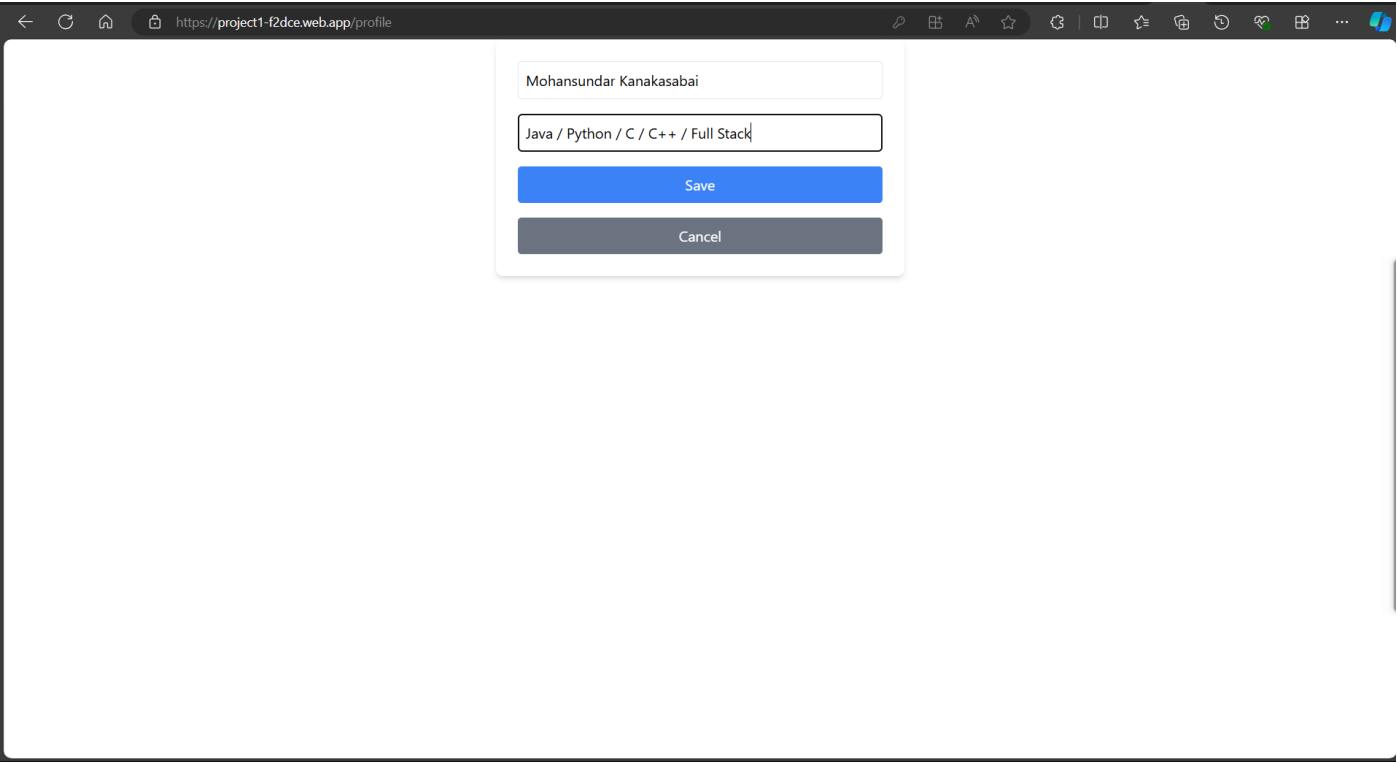


A screenshot of a web browser displaying the register page. The browser's address bar shows the URL `https://project1-f2dce.web.app/register`. The register form is centered on the page and contains the following elements:

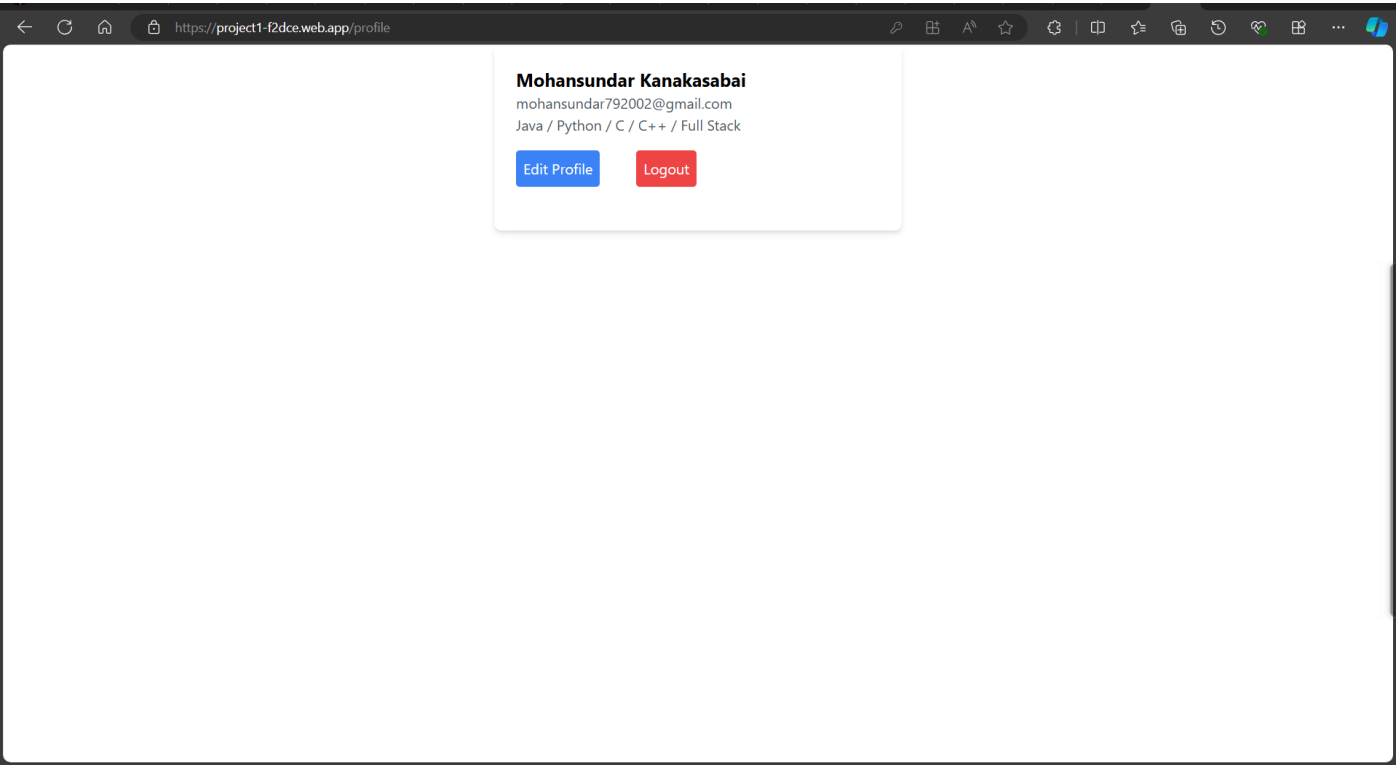
- A text input field labeled "Name".
- A text input field labeled "Email".
- A text input field labeled "Password".
- A text input field labeled "Confirm Password".
- A blue button labeled "Register".
- A red button labeled "Login".



/Edit Profile Page



/Profile Page (After update)



2. (Event Booking System Assessment

Github Link

- [Backend](#)

BASEURL = event-booking-system-pex8.onrender.com/

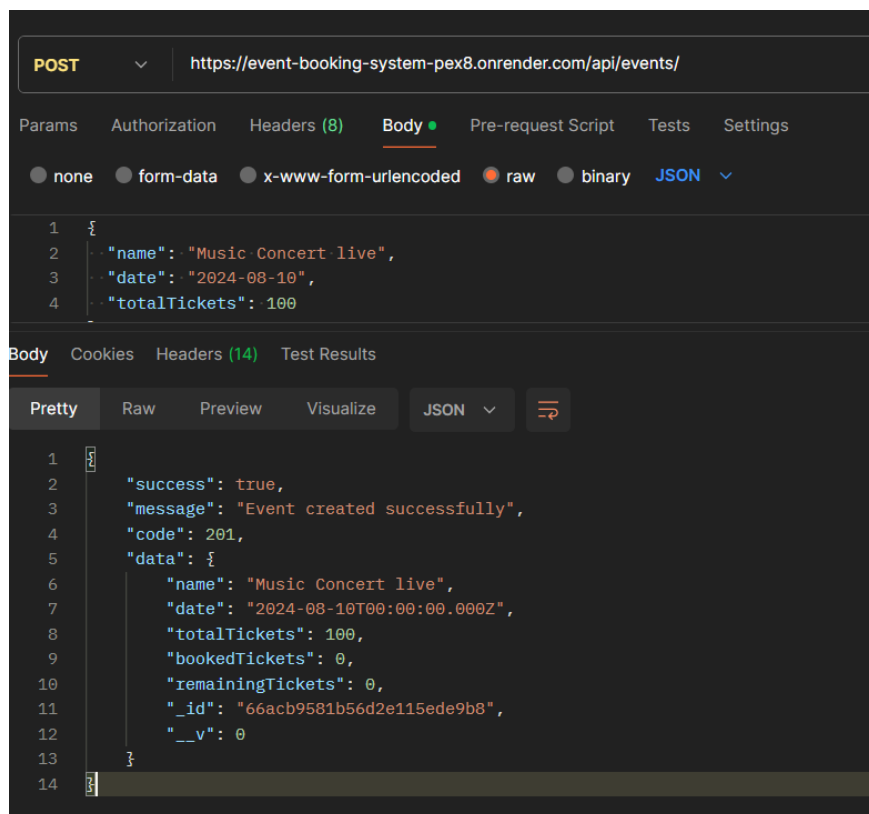
- o POST /events: Create a new event. The request body should contain the event name, date, and total number of tickets.

API - event-booking-system-pex8.onrender.com/api/events

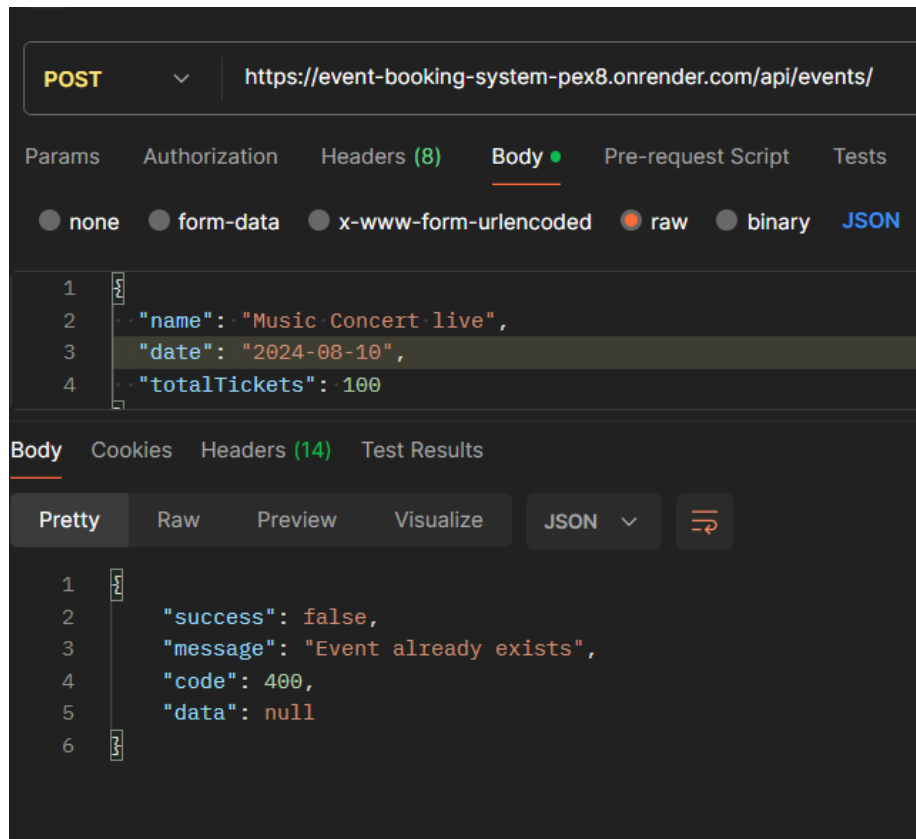
Body :

Scenario - 1 - New Event Creation

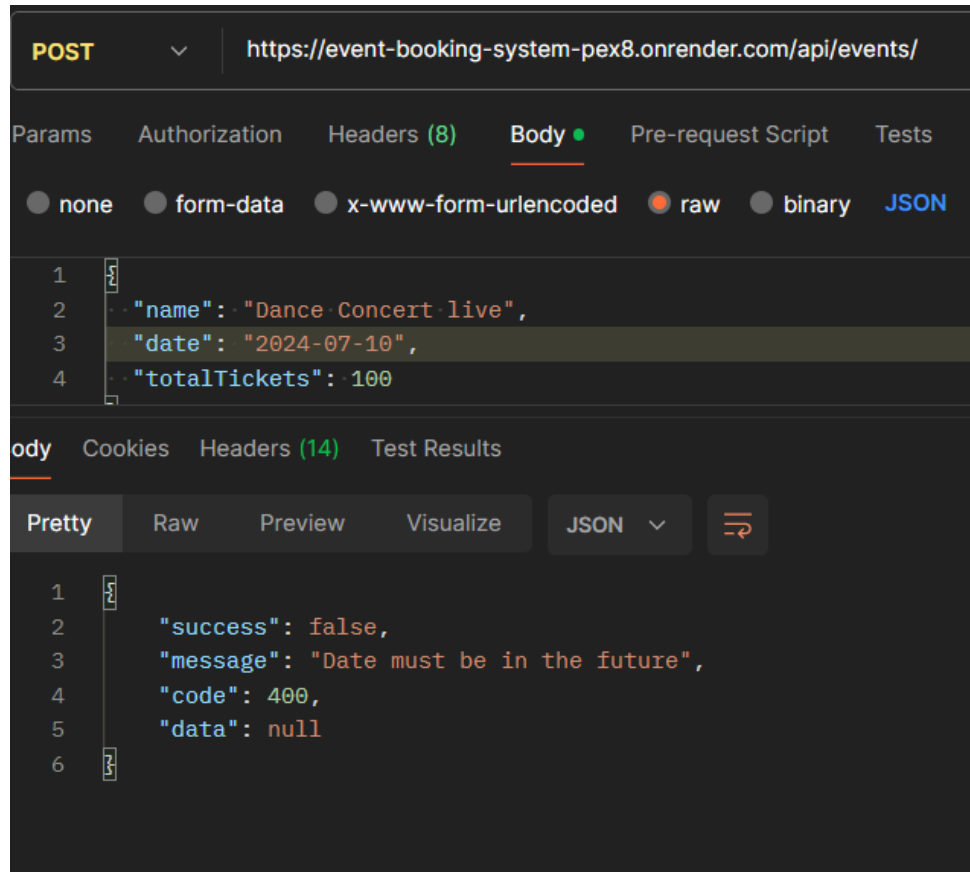
```
{  
"name": "Music Concert ",  
"date": "2024-08-10",  
"totalTickets": 100  
}
```



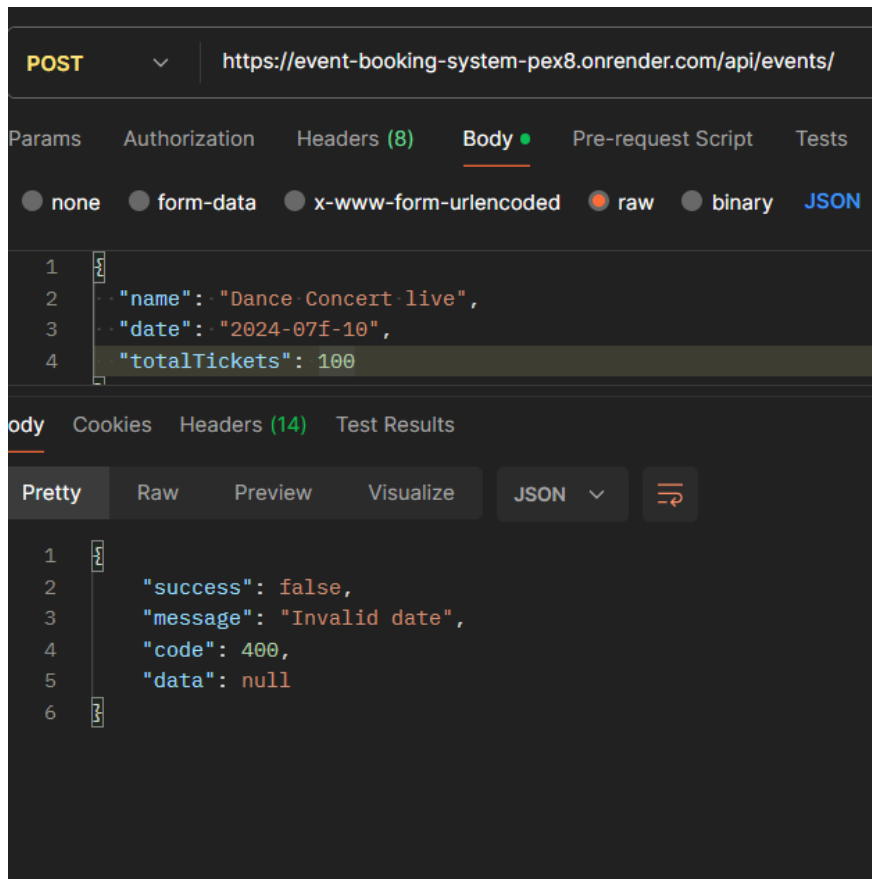
Scenario - 2 - Event Already Exists



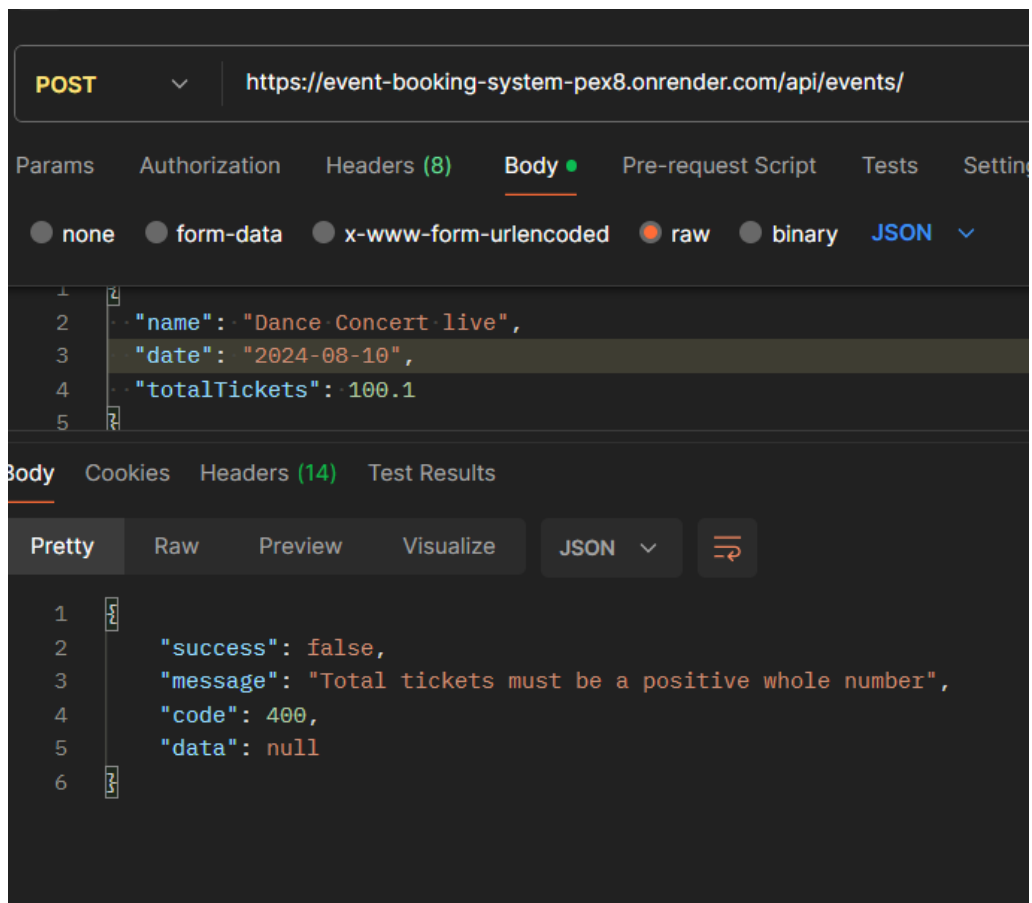
Scenario - 3 - Using Past date



Scenario - 4 - Invalid Date Format



Scenario - 5 - Negative / Decimal totalTickets



Event id = "66acb9581b56d2e115ede9b8"

POST /bookings: Book tickets for an event. The request body should contain the user ID, quantity and event ID

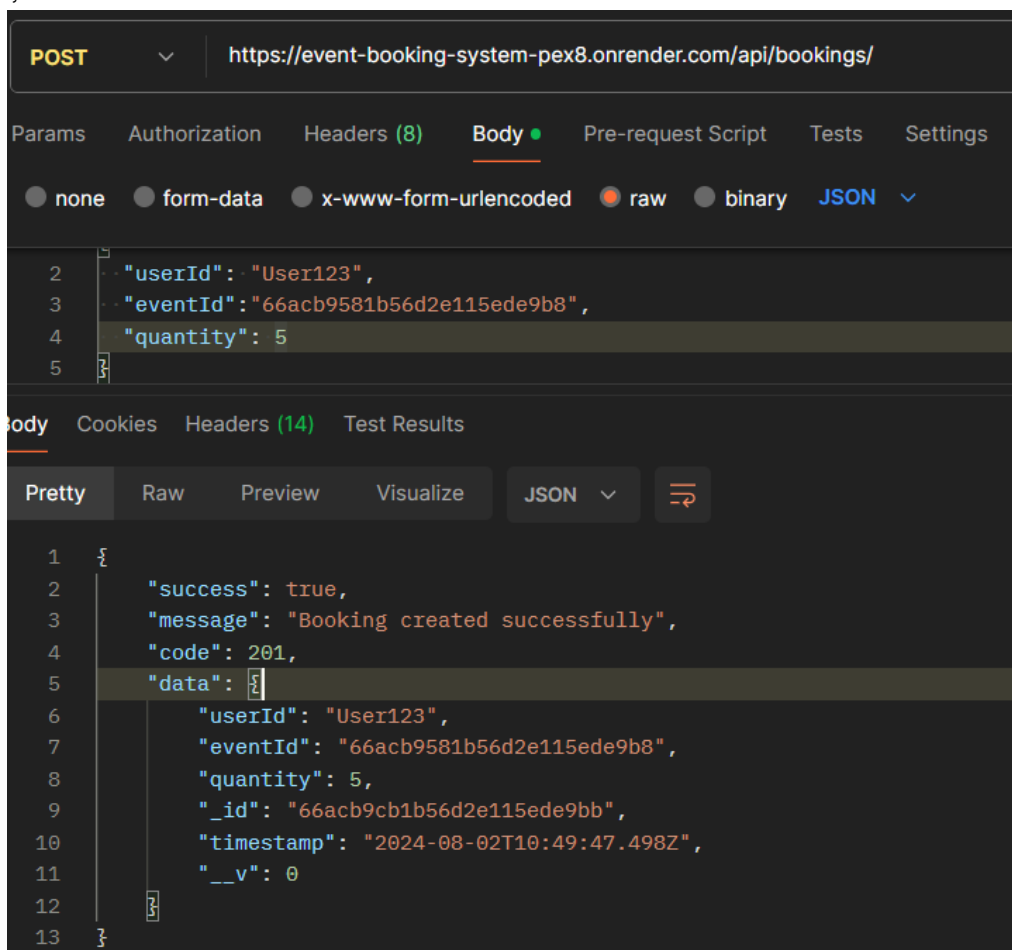
i.Booking Limit:

- Each user can book a maximum of 15 tickets per booking request. This means that if a user tries to book more than 15 tickets in a single request, the system should only process up to 15 tickets and reject any additional requests in that transaction.

ii.Availability Check:

- Before confirming the booking, the system must check if there are enough tickets available for the event. If the event has reached its maximum capacity (i.e., no more tickets are available), the system should prevent further bookings and inform the user that the event is fully booked.

```
{
  "userId": "64c5f6d4e8a84e30c8b5fcd1",
  "eventId": "64c5f6d4e8a84e30c8b5fcd0",
  "quantity": 5
}
```



After booking ticket remaining tickets decreased to 95.

```
← × 🏠 🔒 https://event-booking-system-pex8.onrender.com/api/events/

1 {
2   "success": true,
3   "message": "Events retrieved successfully",
4   "code": 200,
5   "data": [
6     {
7       "_id": "66acb9581b56d2e115ede9b8",
8       "name": "Music Concert live",
9       "date": "2024-08-10T00:00:00.000Z",
10      "totalTickets": 100,
11      "bookedTickets": 5,
12      "remainingTickets": 95,
13      "__v": 0
14    }
15  ]
16 }
```

Scenario - 2 - Booking more than 15 tickets

```
POST http://localhost:5000/api/bookings/

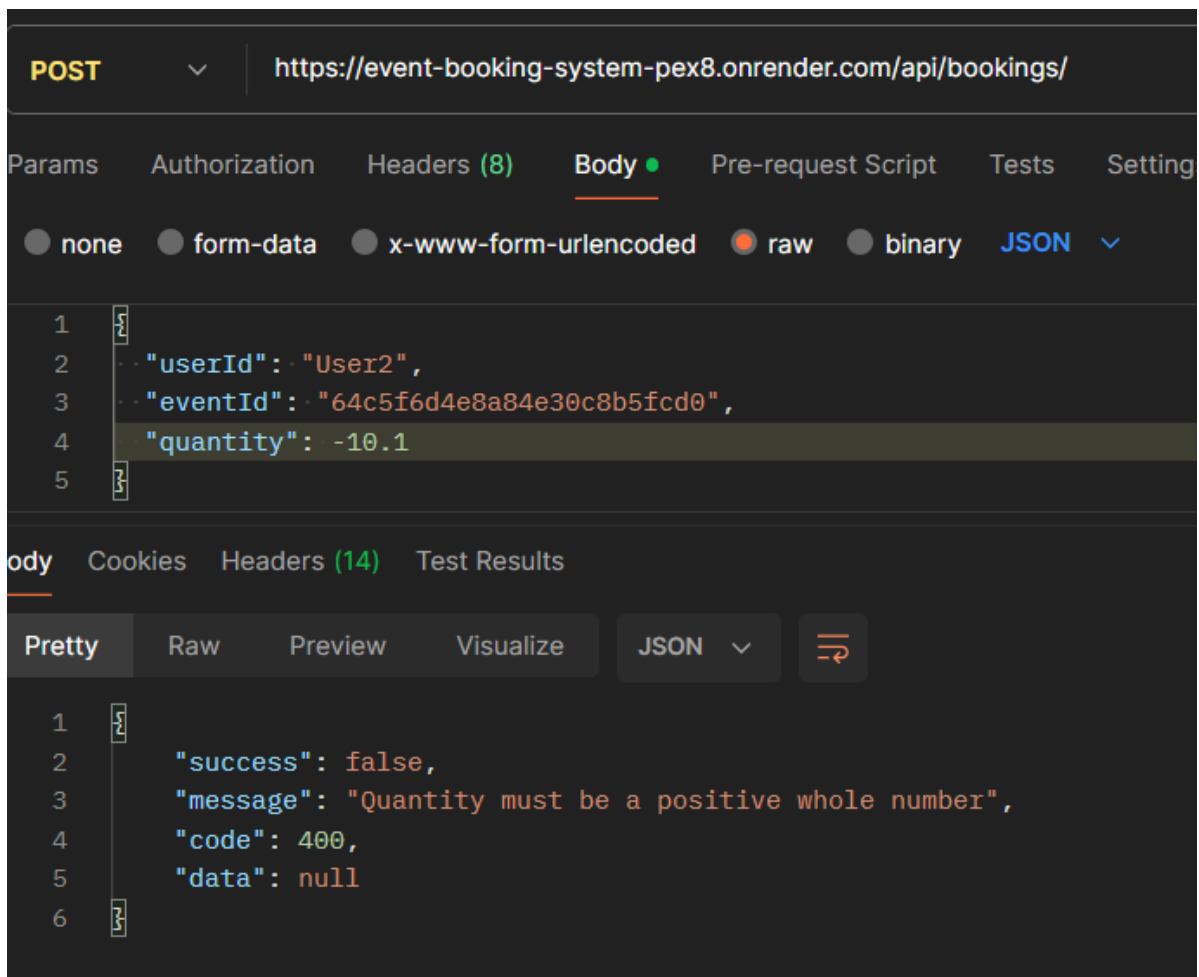
Params Authorization Headers (8) Body Pre-request Script Tests Set
● none ● form-data ● x-www-form-urlencoded ● raw ● binary JSON ▼

1 {
2   "userId": "User3",
3   "eventId": "66acb9581b56d2e115ede9b8",
4   "quantity": 16
5 }

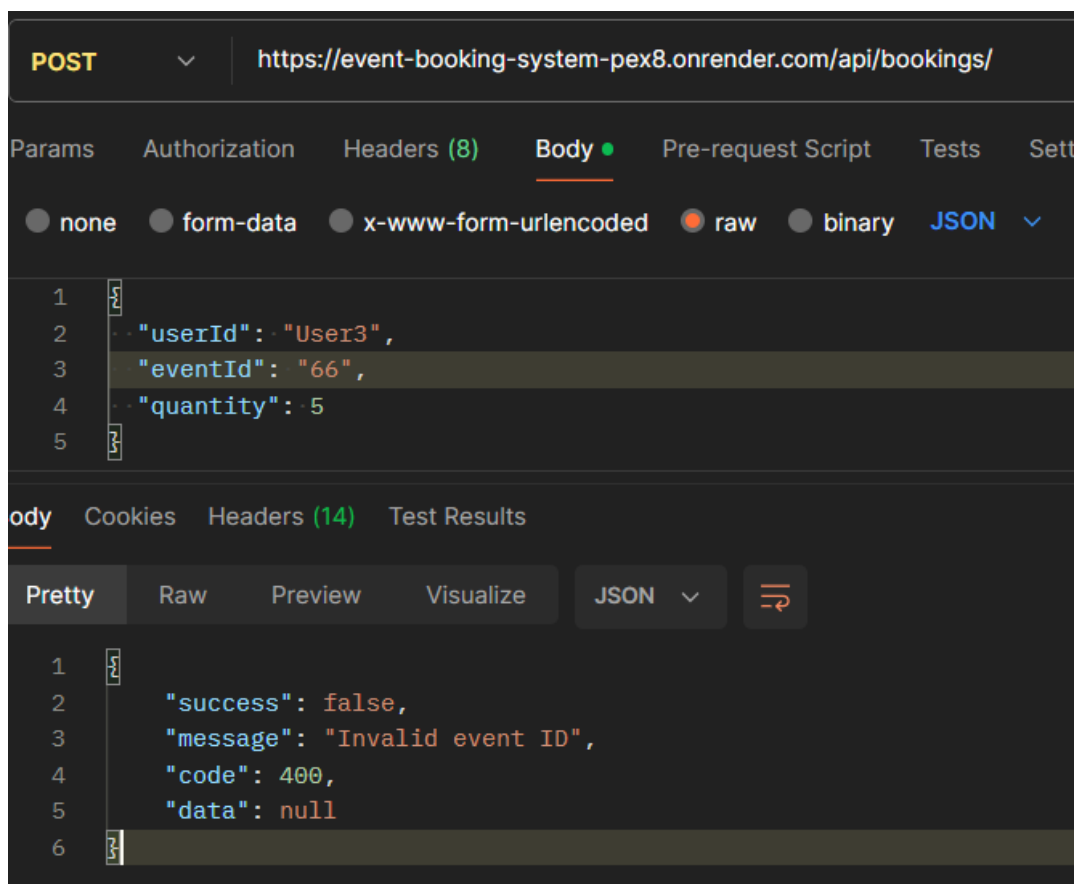
Body Cookies Headers (7) Test Results
Pretty Raw Preview Visualize JSON ↺

1 {
2   "success": true,
3   "message": "Booking created successfully",
4   "code": 201,
5   "data": {
6     "userId": "User3",
7     "eventId": "66acb9581b56d2e115ede9b8",
8     "quantity": 15,
9     "_id": "66acbea50e457c34ec988a7b",
10    "timestamp": "2024-08-02T11:10:29.219Z",
11    "__v": 0
12  }
```

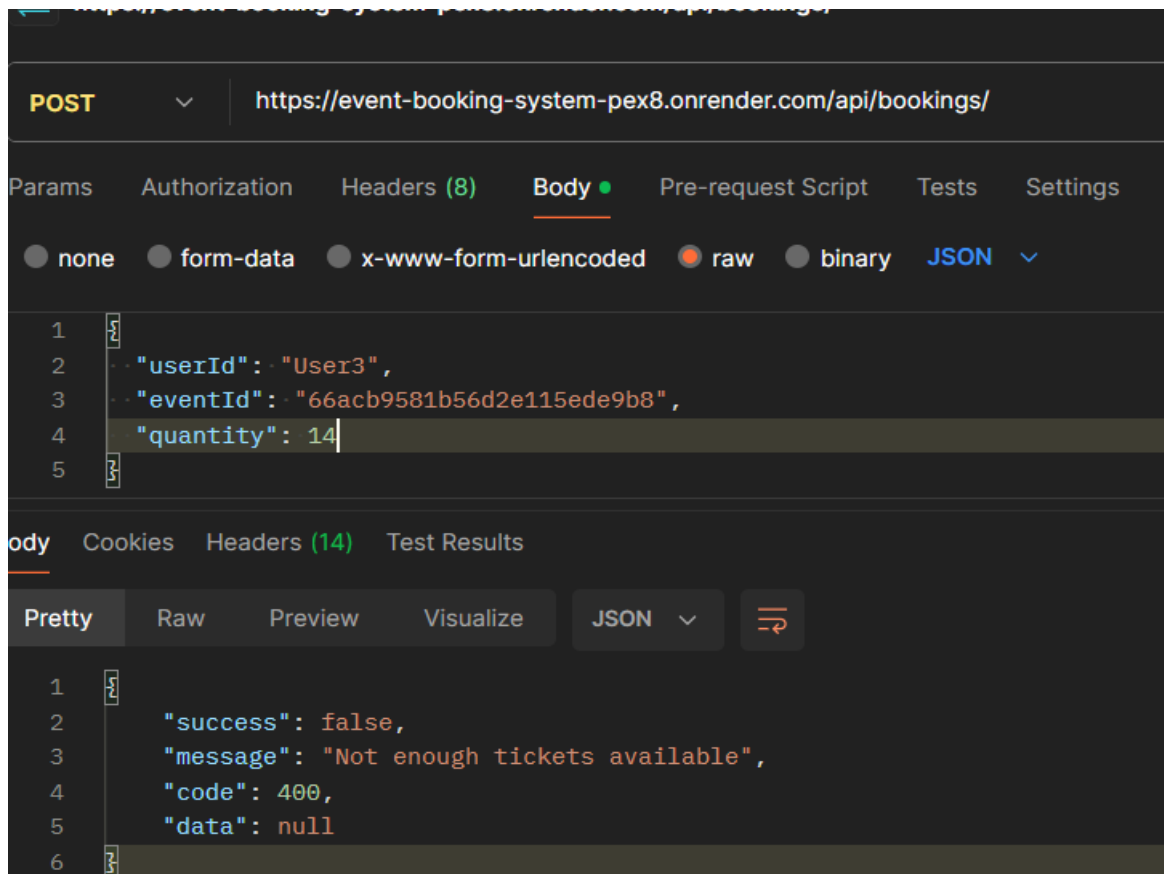
Scenario - 3 - Quantity in decimal/negative



Scenario - 4 - Invalid Event ID

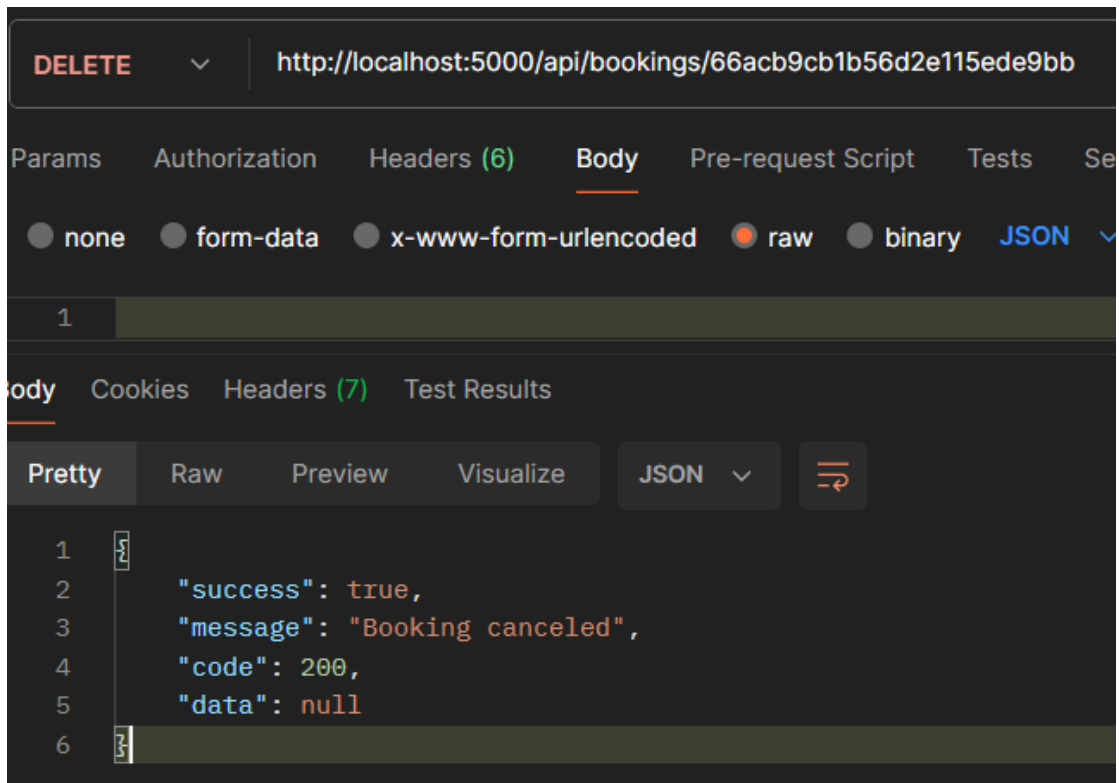


Scenario - 5 - Not enough tickets available

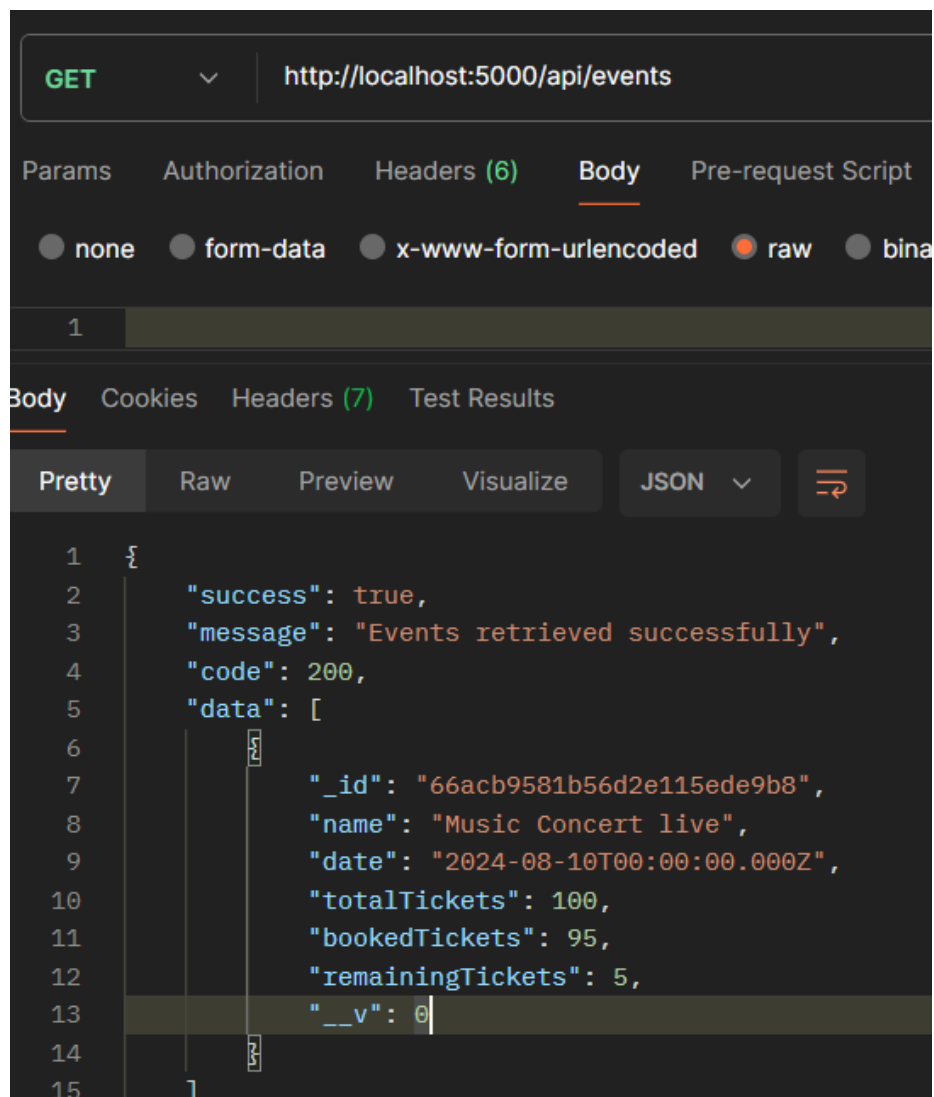


```
localhost:5000/api/bookings
1 {
2   "success": true,
3   "message": "Bookings retrieved successfully",
4   "code": 200,
5   "data": [
6     {
7       "_id": "66acb9cb1b56d2e115ede9bb",
8       "userId": "User123",
9       "eventId": "66acb9581b56d2e115ede9b8",
10      "quantity": 5,
11      "timestamp": "2024-08-02T10:49:47.498Z",
12      "__v": 0
13    },
14    {
15      "_id": "66acbcc61b56d2e115ede9c4",
16      "userId": "User3",
17      "eventId": "66acb9581b56d2e115ede9b8",
18      "quantity": 5,
19      "timestamp": "2024-08-02T11:02:30.749Z",
20      "__v": 0
21    },
22    {
23      "_id": "66acbcef1b56d2e115ede9c8",
24      "userId": "User3",
25      "eventId": "66acb9581b56d2e115ede9b8",
26      "quantity": 5,
27      "timestamp": "2024-08-02T11:03:11.548Z",
28      "__v": 0
29    },
30    {
31      "_id": "66acbcf41b56d2e115ede9cc",
32      "userId": "User3",
33      "eventId": "66acb9581b56d2e115ede9b8",
34      "quantity": 14,
35      "timestamp": "2024-08-02T11:03:16.196Z",
36      "__v": 0
37    },
38    {
39      "_id": "66acbcf51b56d2e115ede9d0",
40      "userId": "User3",
41      "eventId": "66acb9581b56d2e115ede9b8",
42      "quantity": 14,
43      "timestamp": "2024-08-02T11:03:17.850Z",
44      "__v": 0
45    },
46    {
47      "_id": "66acbcf71b56d2e115ede9d4",
48      "userId": "User3",
49      "eventId": "66acb9581b56d2e115ede9b8",
50      "quantity": 14,
51      "timestamp": "2024-08-02T11:03:19.510Z",
52      "__v": 0
53    },
54    {
55      "_id": "66acbcfb1b56d2e115ede9dc",
56      "userId": "User3",
57      "eventId": "66acb9581b56d2e115ede9b8",
58      "quantity": 14,
59      "timestamp": "2024-08-02T11:03:23.012Z",
60      "__v": 0
61    }
62  ]
63 }
```

- DELETE /bookings/:id: Cancel a booking by ID.



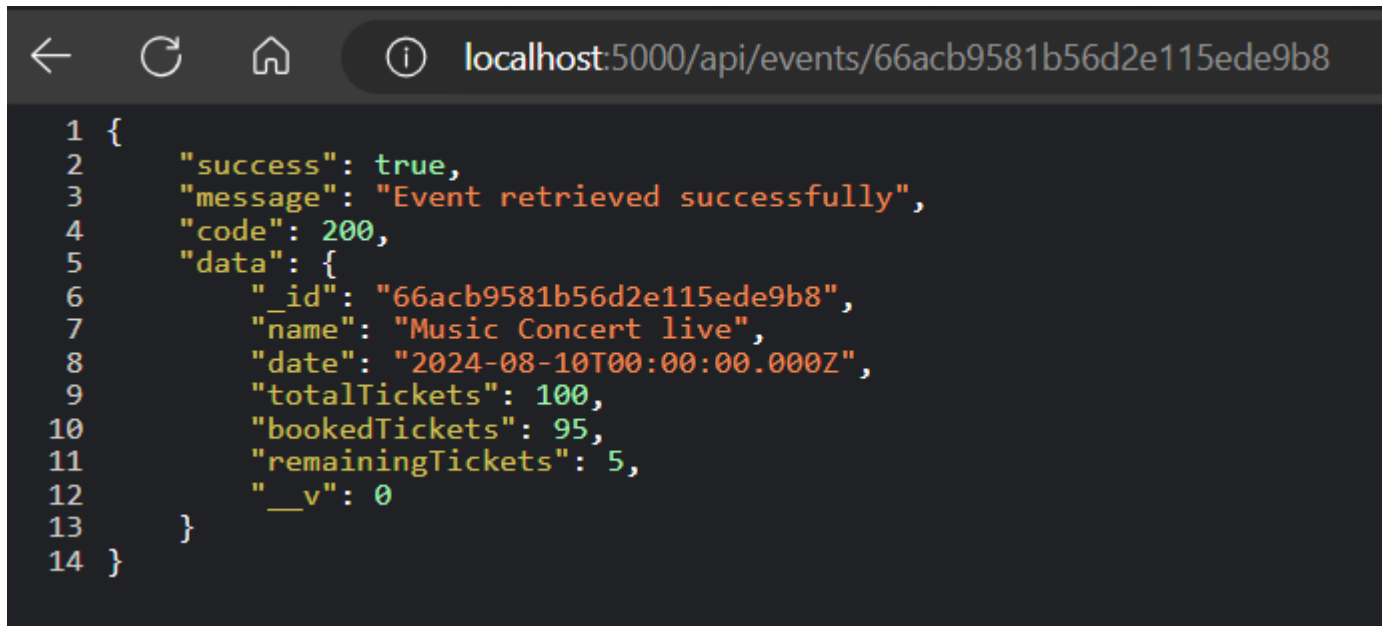
After deleting booking with 5 tickets



- o GET /events: Retrieve a list of events with available tickets.

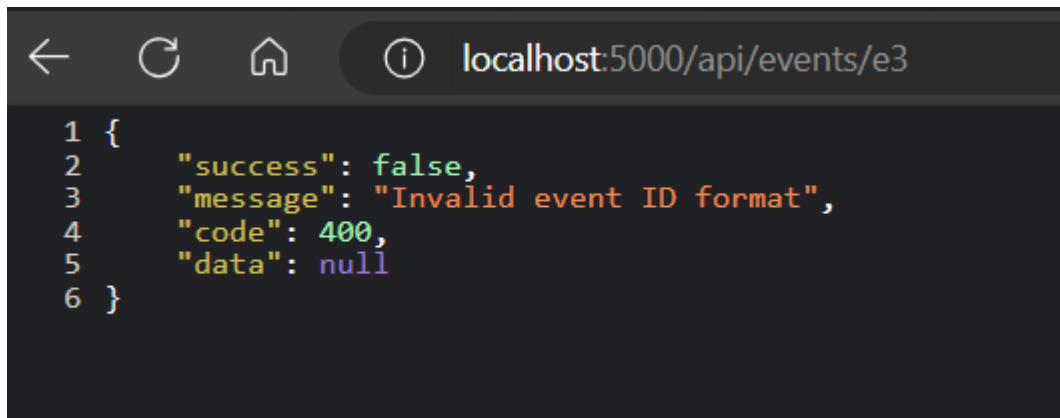
- o GET /events/:id: Retrieve details of a specific event, including booked tickets and remaining tickets.

Scenario - 1 - Successful finding of event



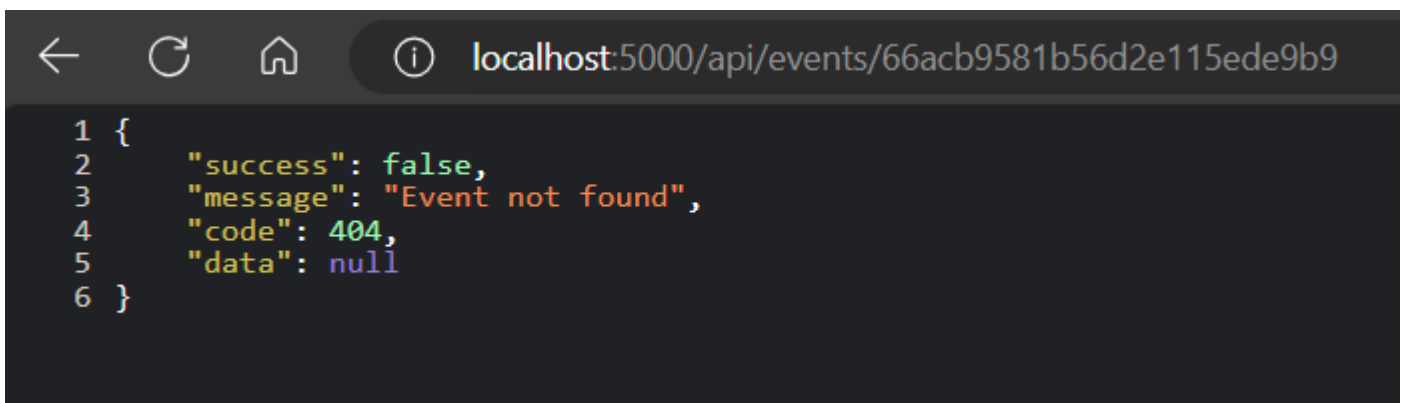
```
1 {
2   "success": true,
3   "message": "Event retrieved successfully",
4   "code": 200,
5   "data": {
6     "_id": "66acb9581b56d2e115ede9b8",
7     "name": "Music Concert live",
8     "date": "2024-08-10T00:00:00.000Z",
9     "totalTickets": 100,
10    "bookedTickets": 95,
11    "remainingTickets": 5,
12    "__v": 0
13  }
14 }
```

Scenario - 2 - Invalid Event id



```
1 {
2   "success": false,
3   "message": "Invalid event ID format",
4   "code": 400,
5   "data": null
6 }
```

Scenario - 3 - Event Not Found



```
1 {
2   "success": false,
3   "message": "Event not found",
4   "code": 404,
5   "data": null
6 }
```

Scenario - 4 - Getting all events

```
1 {
2   "success": true,
3   "message": "Events retrieved successfully",
4   "code": 200,
5   "data": [
6     {
7       "_id": "66acb9581b56d2e115ede9b8",
8       "name": "Music Concert live",
9       "date": "2024-08-10T00:00:00.000Z",
10      "totalTickets": 100,
11      "bookedTickets": 95,
12      "remainingTickets": 5,
13      "__v": 0
14    },
15    {
16      "_id": "66acc1010e457c34ec988a8b",
17      "name": "Standup Comedy",
18      "date": "2024-09-07T00:00:00.000Z",
19      "totalTickets": 120,
20      "bookedTickets": 0,
21      "remainingTickets": 0,
22      "__v": 0
23    },
24    {
25      "_id": "66acc1100e457c34ec988a8e",
26      "name": "Memicry Schow",
27      "date": "2024-09-08T00:00:00.000Z",
28      "totalTickets": 120,
29      "bookedTickets": 0,
30      "remainingTickets": 0,
31      "__v": 0
32    }
33  ]
34 }
```


POST /print-ticket: This endpoint allows users to print a ticket for a specific booking. It generates a printable format of the ticket, which includes details of the event and booking.

POST

http://localhost:5000/api/bookings/print-ticket

Send

Params

Authorization

Headers (8)

Body

Pre-request Script

Tests

Settings

none

form-data

x-www-form-urlencoded

raw

binary

JSON

1

2

3

["bid": "66acbcef1b56d2e115ede9c8"]

Body

Cookies

Headers (7)

Test Results

Status: 200 OK

Time: 540 ms

Size: 1.68 KB

Save Response

Event Ticket

Event: Music Concert live
Date: Sat Aug 10 2024 05:30:00 GMT+0530 (India Standard Time)
User ID: User3
Quantity: 5
Ticket Issued Time: Fri Aug 02 2024 16:33:11 GMT+0530 (India Standard Time)

http://localhost:5000/api/bookings/print-ticket

Save

</>

POST

http://localhost:5000/api/bookings/print-ticket

Send

Params

Authorization

Headers (8)

Body

Pre-request Script

Tests

Settings

none

form-data

x-www-form-urlencoded

raw

binary

JSON

1

2

3

["bid": "66acc1e29e457c34ec988a97"]

Body

Cookies

Headers (7)

Test Results

Status: 200 OK

Time: 495 ms

Size: 1.68 KB

Save Response

Event Ticket

Event: Standup Comedy
Date: Sat Sep 07 2024 05:30:00 GMT+0530 (India Standard Time)
User ID: Mohan123
Quantity: 10
Ticket Issued Time: Fri Aug 02 2024 16:54:18 GMT+0530 (India Standard Time)