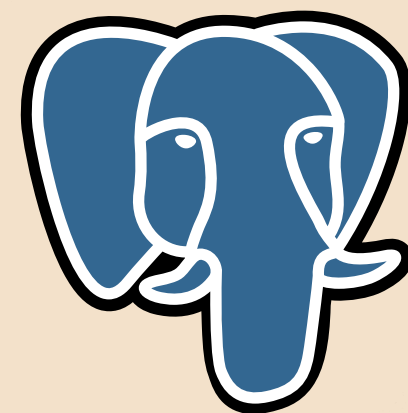
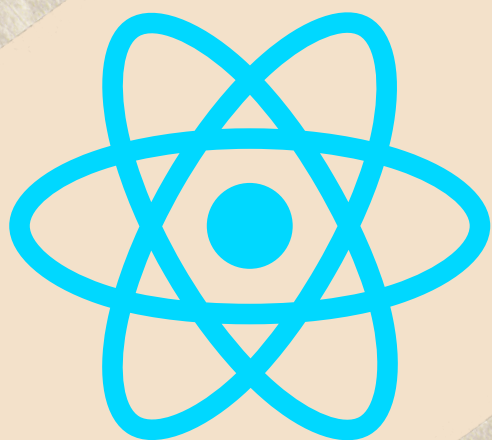


SWIGGY I++

Swiggi food app

Yash Mantri

TECHNOLOGY STACK



Content



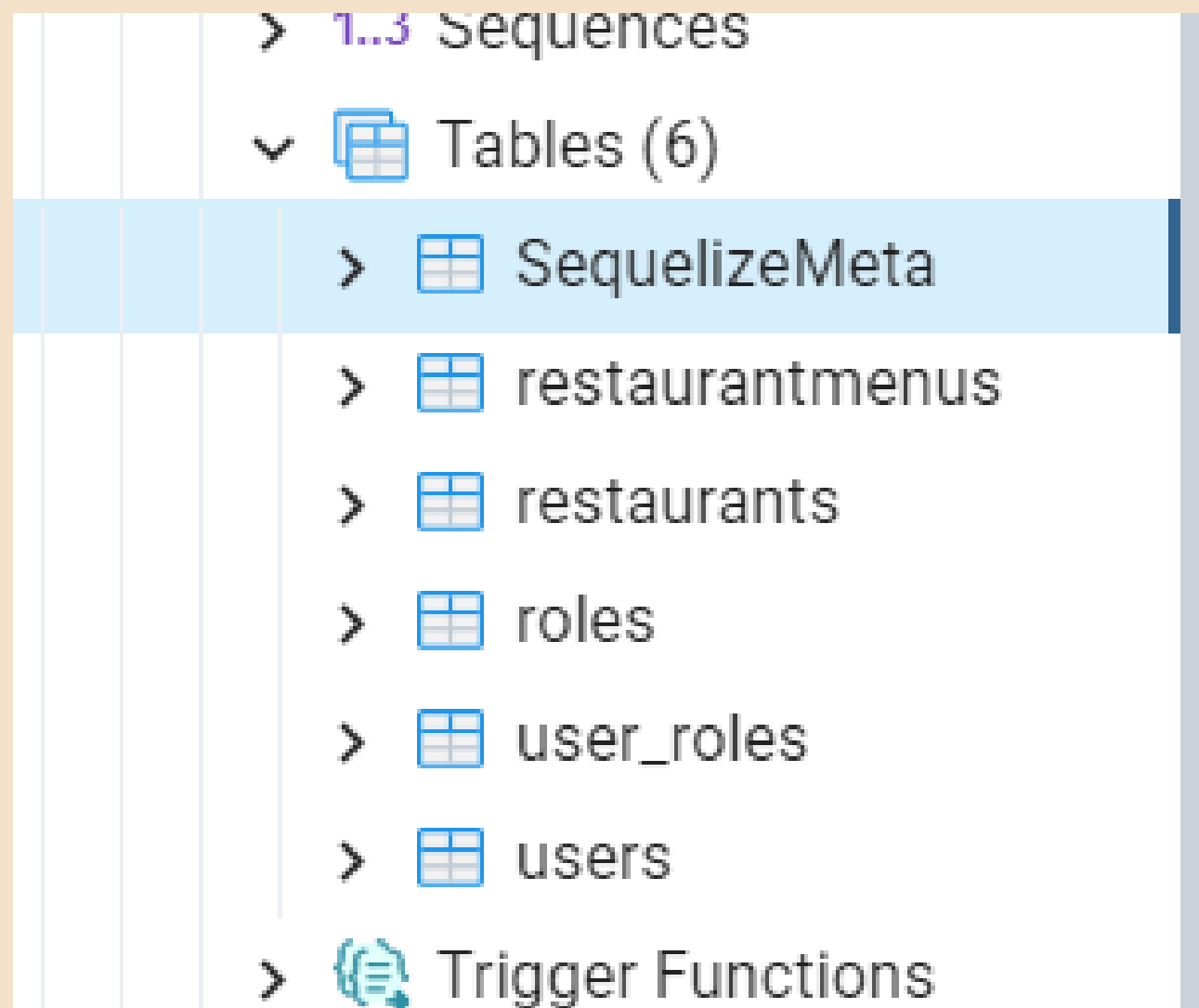
BACKEND

FRONTEND

Database

4 Tables:

1. Users : Store username,email-id,password and roles
2. Roles: Store user_role(Moderator, User, admin)
3. Restaurant : Restaurant_id, name
4. Menu : Price, Name



Routes

1. APP.GET('/API') : WELCOME MESSAGE
2. APP.POST('/API/RESTUARANT') : TO CREATE RESTAURANT WITH GIVEN TITLE

http://127.0.0.1:8000/api

GET

http://127.0.0.1:8000/api

Params

Authorization

Headers (9)

Body ●

Pre-request Script

Tests

Body

Cookies

Headers (9)

Test Results

Pretty

Raw

Preview

Visualize

JSON

▼



```
1 {  
2   "message": "Welcome to the restuarant API!"  
3 }
```

POST

http://127.0.0.1:8000/api/restuarant

Params

Authorization

Headers (9)

Body ●

Pre-request Script

Tests

● none

● form-data

● x-www-form-urlencoded

● raw

● binary

● GraphQL

```
1 {  
2   "id": 3,  
3   "title": "new restaurant"
```

Body

Cookies

Headers (9)

Test Results

Pretty

Raw

Preview

Visualize

JSON

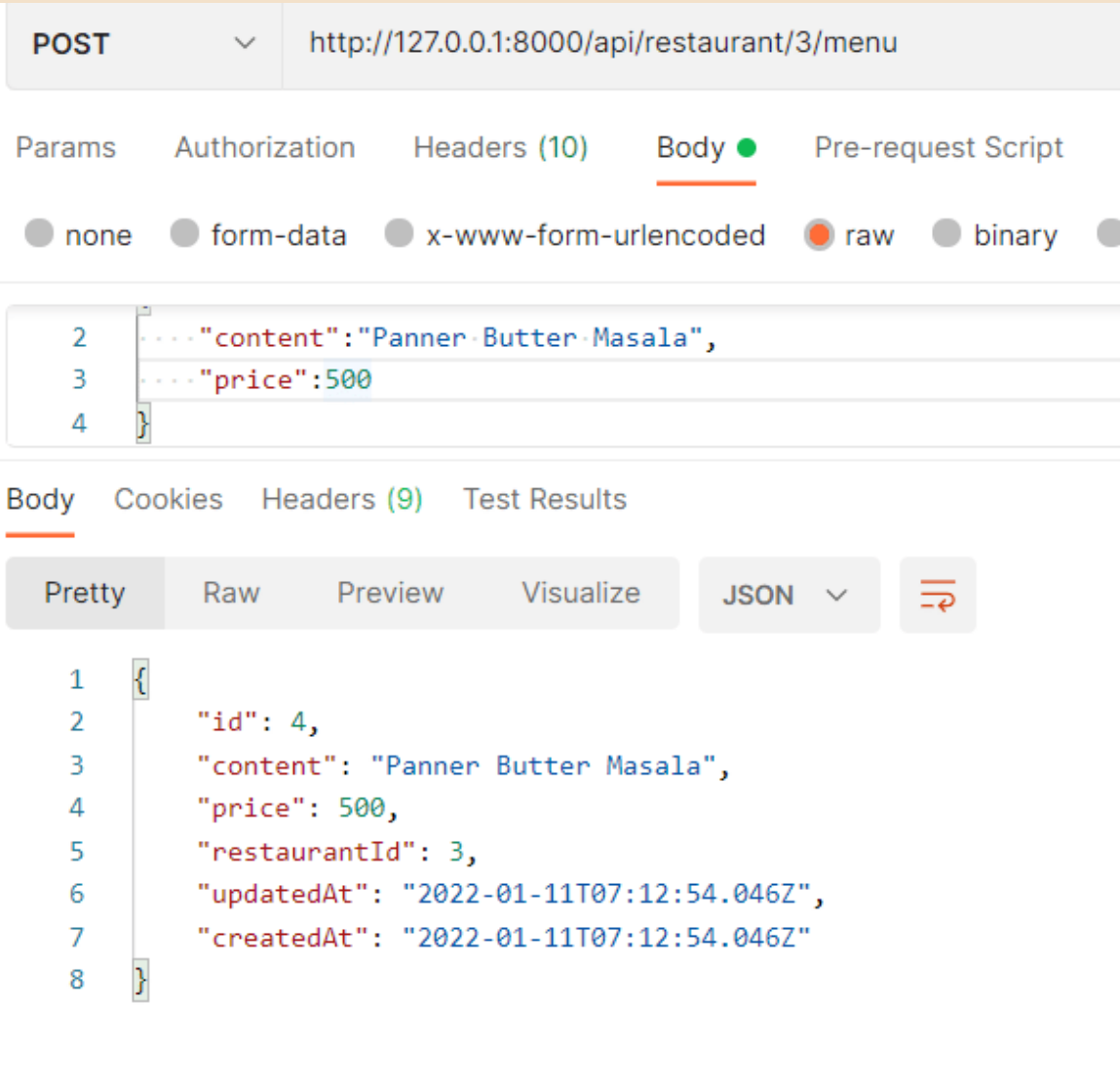
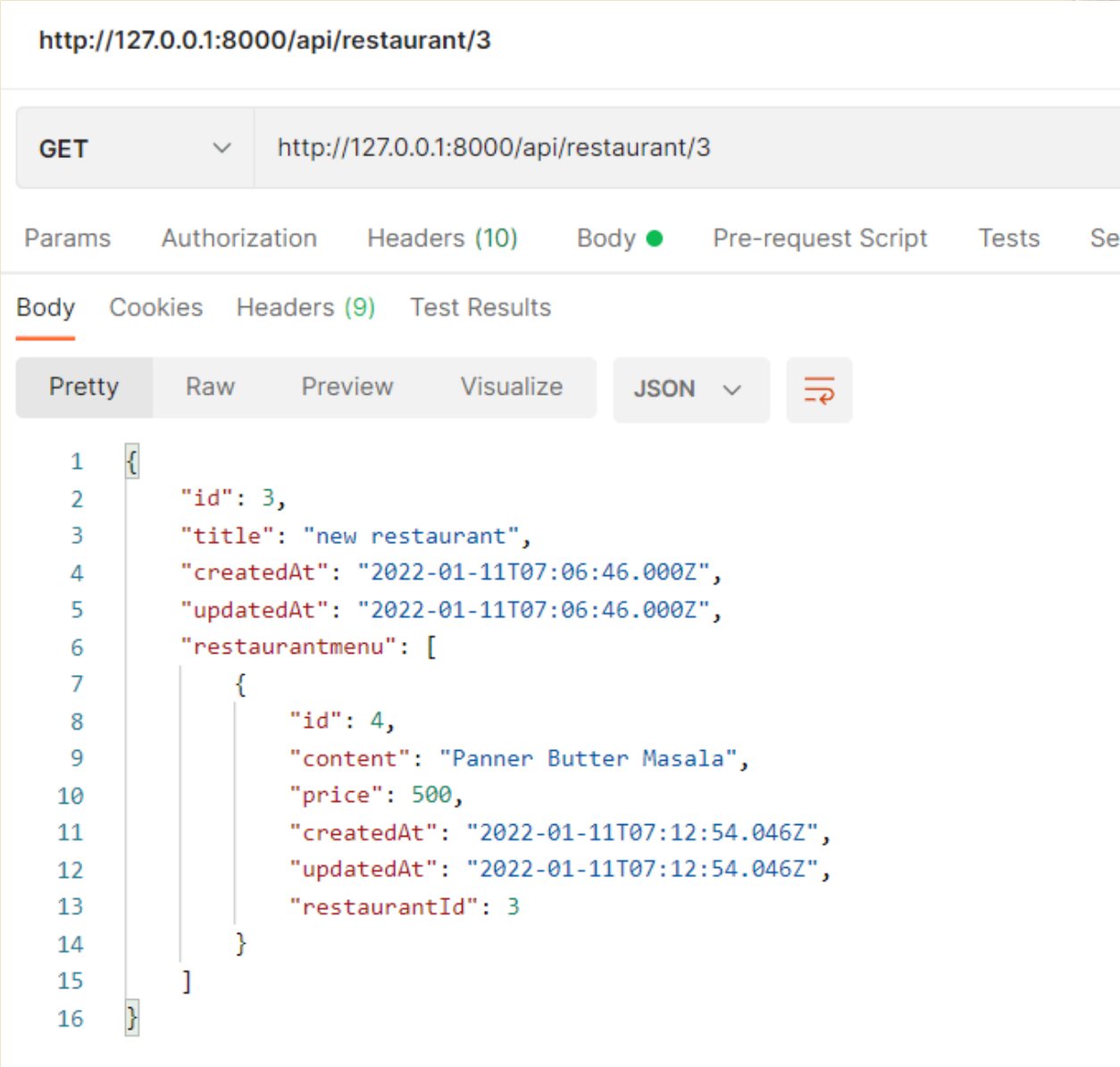
▼



```
1 {  
2   "id": 3,  
3   "title": "new restaurant",  
4   "updatedAt": "2022-01-11T07:06:46.000Z",  
5   "createdAt": "2022-01-11T07:06:46.000Z"  
6 }
```

3. APP.POST('/API/RESTAURANT/:RESTAURANTID/MENU')

To add the item in menu of the restaurant

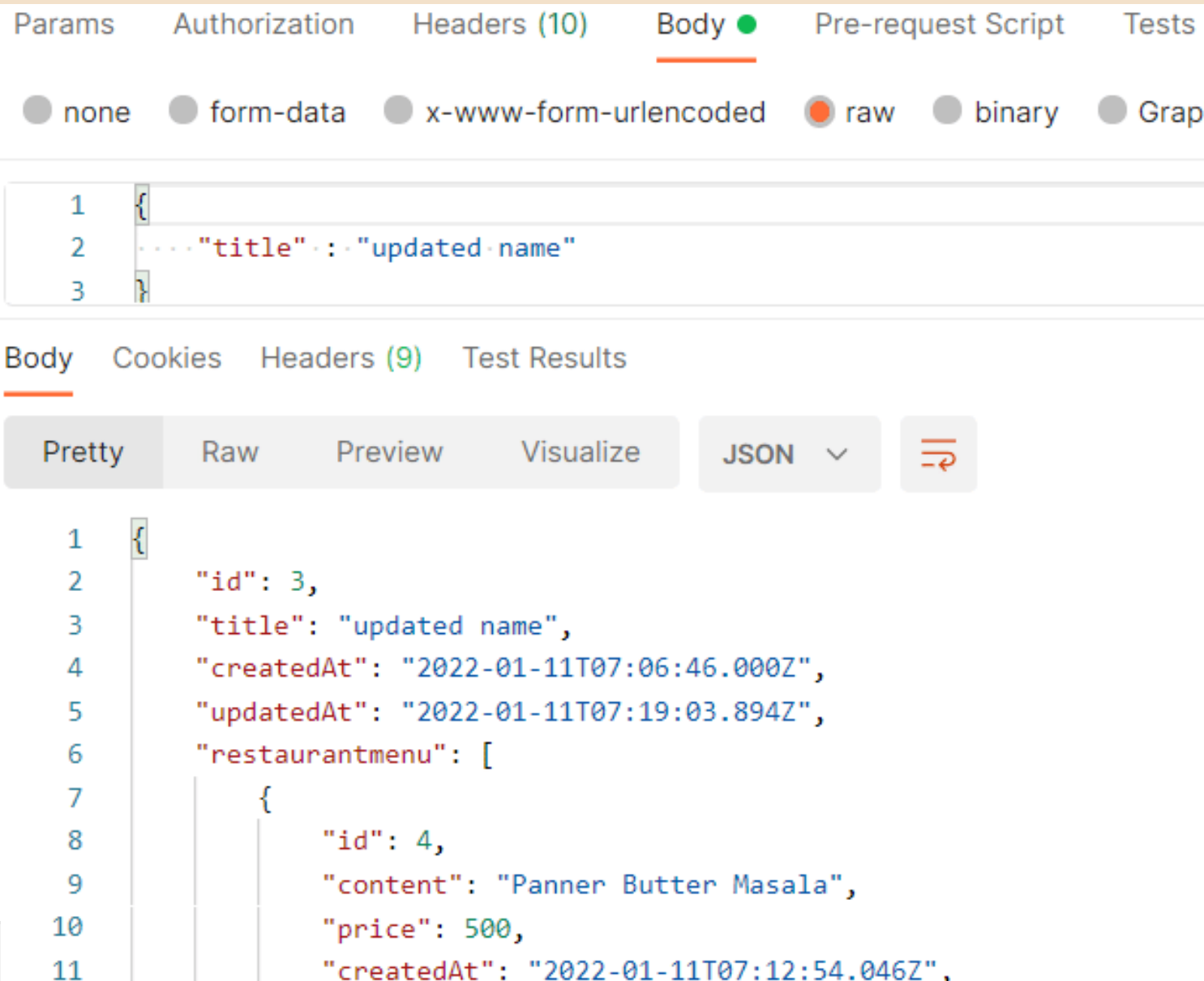


4.APP.GET('/API/RESTAURANT/:RESTAURANTID')

To get the information about the restaurant

5. APP.PUT ('/API/RESTAURANT/:RESTAURANTID')

To update the restaurant name. The restaurant with id:3 name is being updated in the example



6. APP.DELETE ('/API/RESTAURANT/:RESTAURANTID')

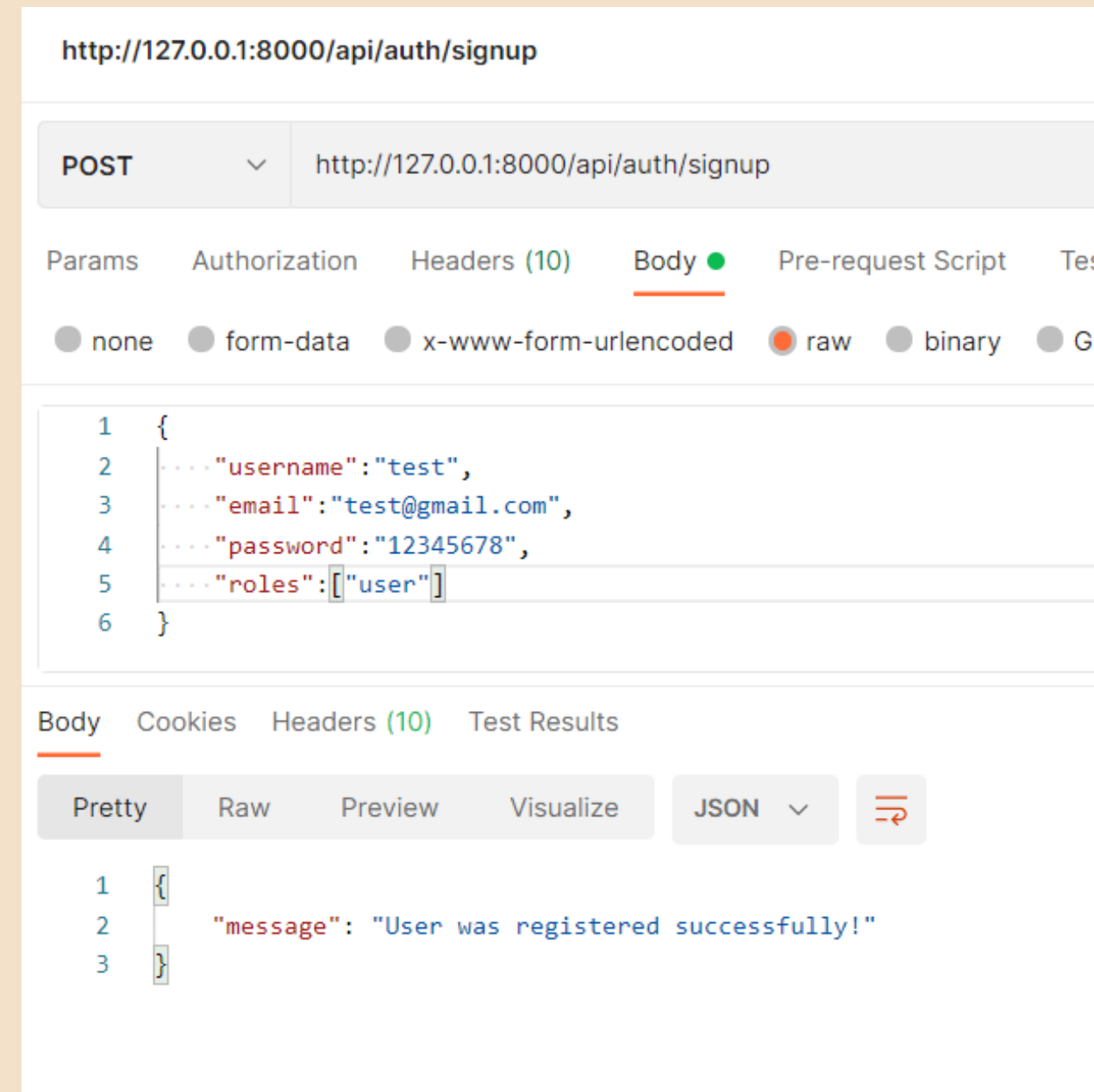
To completely delete the restaurant from the data. In the example after deleting restaurant with id : 3 the output in databse is

DELETE		http://127.0.0.1:8000/api/restaurant/3			
Data Output		Explain	Messages	Notifications	
	id [PK] integer	title character varying (255)	createdAt timestamp with time zone	updatedAt timestamp with time zone	
1	1	New eagle	2022-01-08 12:10:48.789+05:30	2022-01-08 12:10:48.789+05:30	
2	2	Vandu International	2022-01-09 13:20:57.927+05:30	2022-01-09 13:20:57.927+05:30	

SignUp

API : <http://127.0.0.1:8000/api/auth/signup>

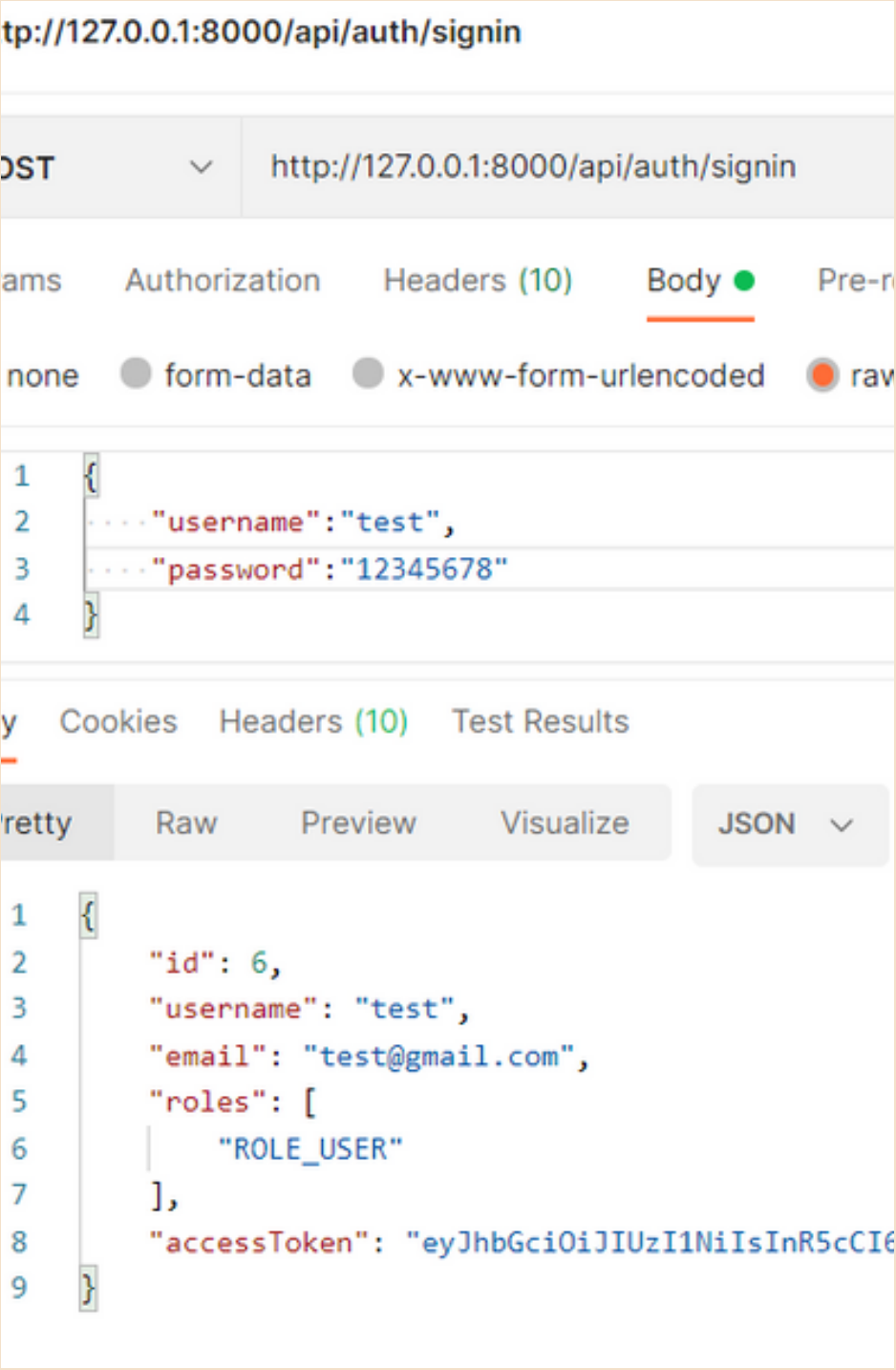
During a SingUp a token is generated using the secret key which helps in the authentication. It also checks wheter the email or username already exists in the database or not



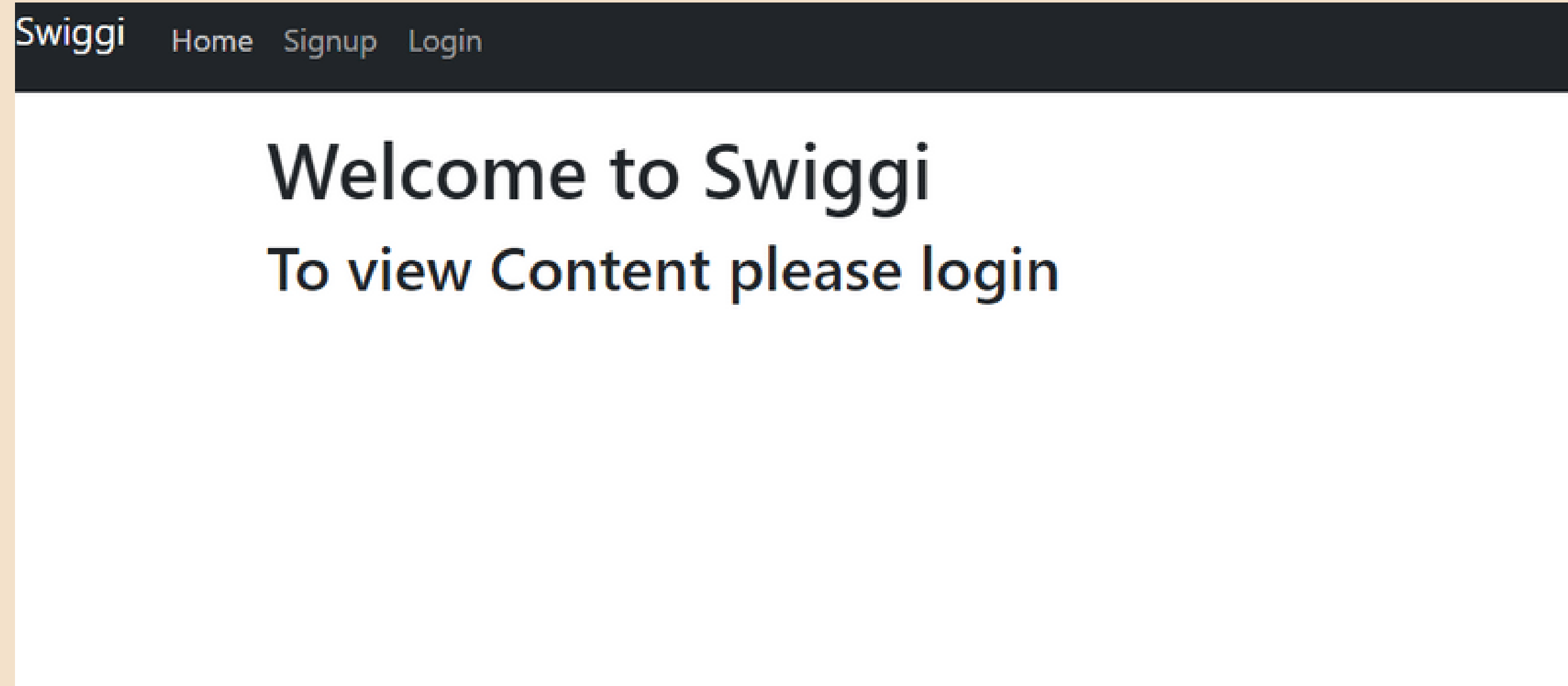
SignIn

API:
HTTP://127.0.0.1:8000/API/AUTH/SIGNIN
FOR THE USERNAME AND PASSWORD IS
REQUIRED. IN THE OUTPUT WE GET ALL THE
USER INFORMATION INCLUDING HIS TOKEN
WHICH IS LATER USED FOR
AUTHENTICATION

API:
HTTP://127.0.0.1:8000/API/AUTH/SIGNIN
FOR THE USERNAME AND PASSWORD IS
REQUIRED. IN THE OUTPUT WE GET ALL THE
USER INFORMATION INCLUDING HIS TOKEN
WHICH IS LATER USED FOR
AUTHENTICATION



Homepage



SignUp Page


Username

Email

Password

Sign Up

Validation on SignUp



Username

The username must be between 3 and 20 characters.


Email

This is not a valid email.

Password

The password must be between 6 and 40 characters.

Sign Up




Username

Email

Password

Sign Up

Failed! Username is already in use!



Username

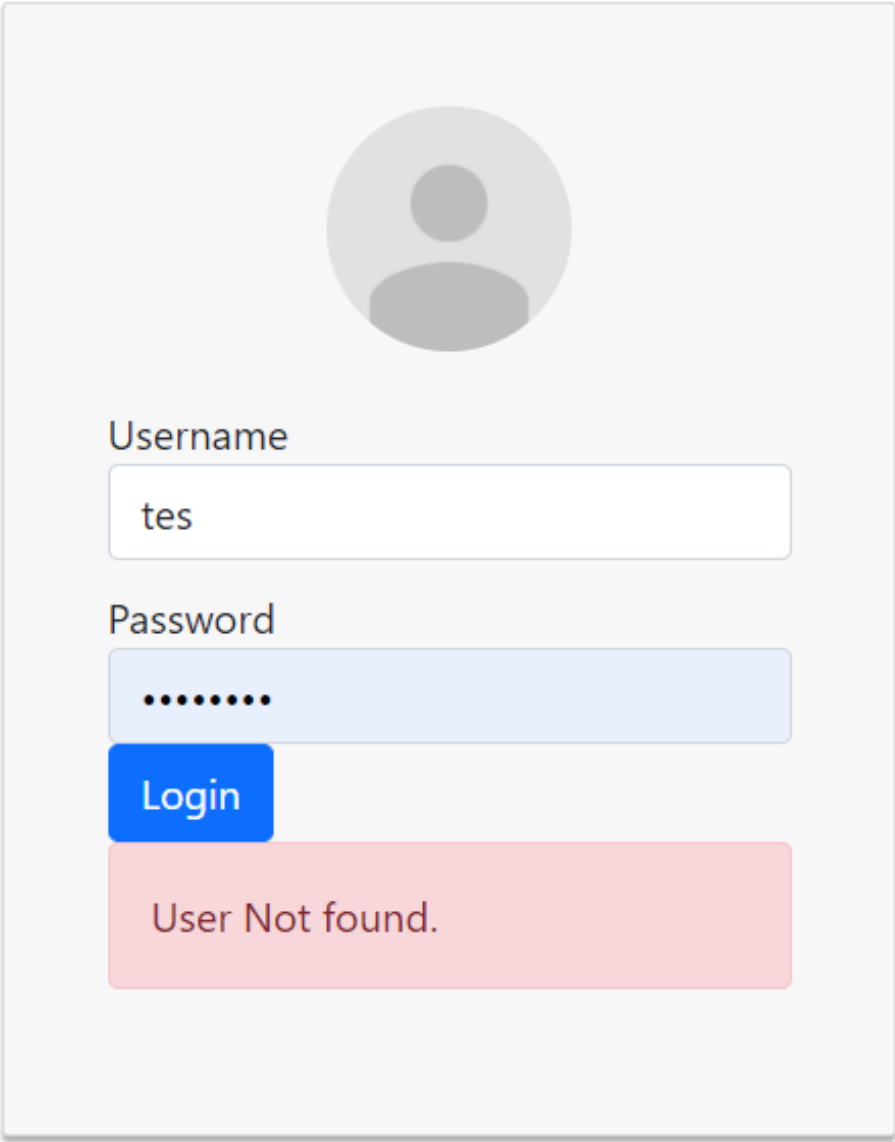
Email

Password

Sign Up

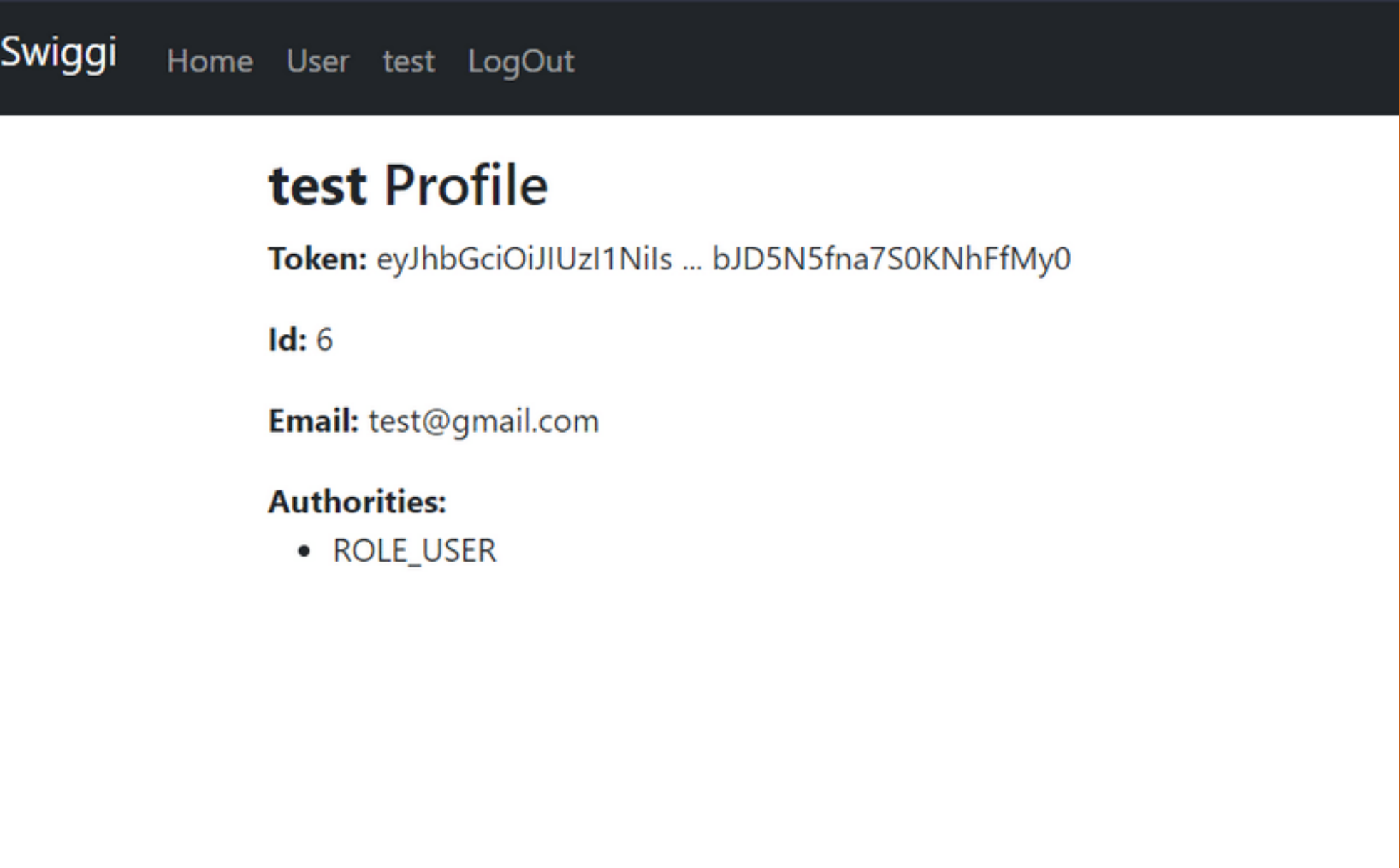
Failed! Email is already in use!

Login



A login form with a light gray background. At the top is a circular placeholder for a profile picture. Below it are two input fields: 'Username' containing 'tes' and 'Password' with masked dots. A blue 'Login' button is positioned below the password field. A pink error message box at the bottom states 'User Not found.'

Wrong Username

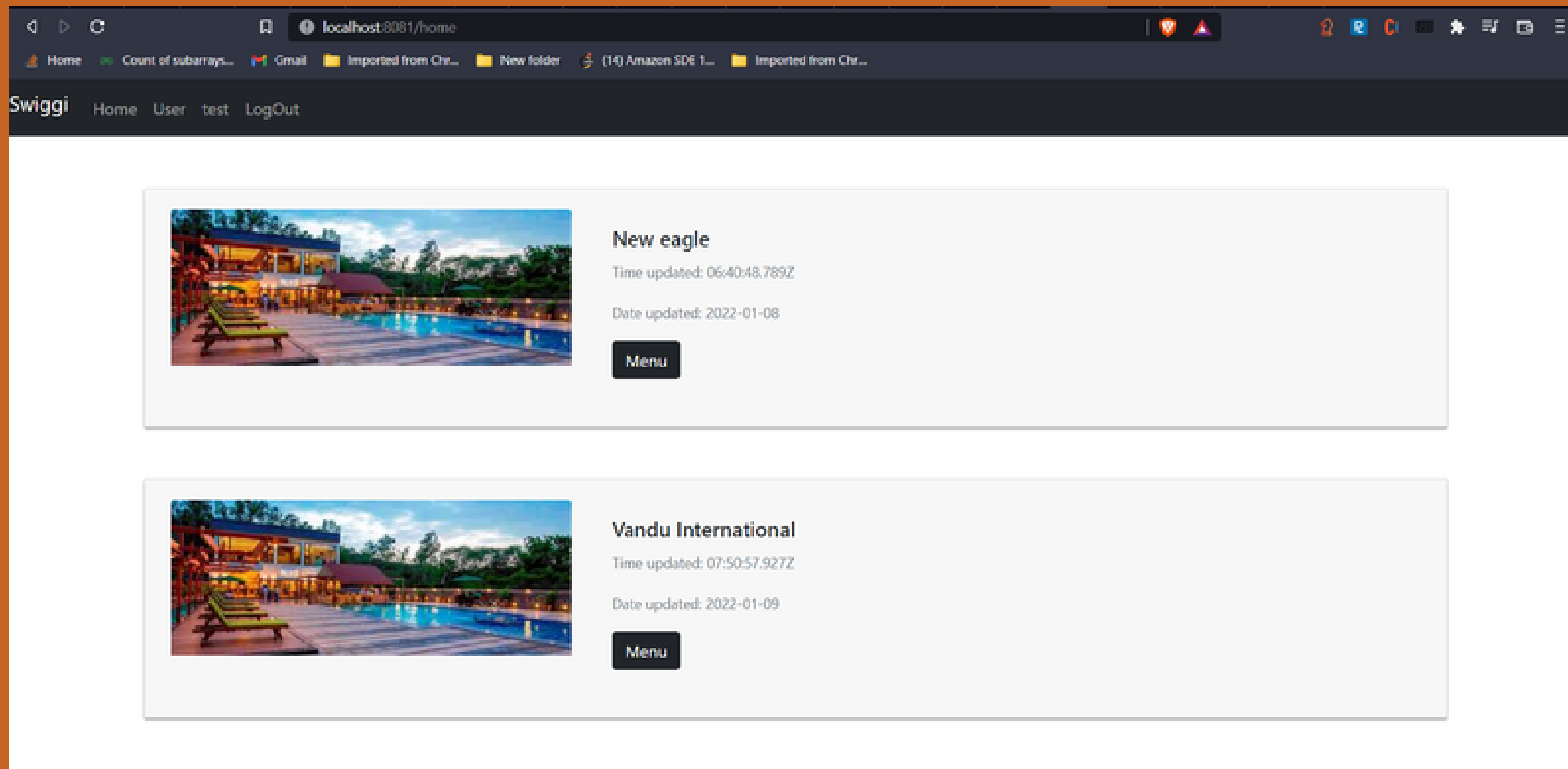


A user profile page with a dark header bar containing the text 'Swiggi Home User test LogOut'. The main content area displays the 'test Profile' with the following details: 'Token: eyJhbGciOiJIUzI1NiIs ... bJD5N5fna7S0KNhFfMy0', 'Id: 6', 'Email: test@gmail.com', and 'Authorities: ROLE_USER'.

After login

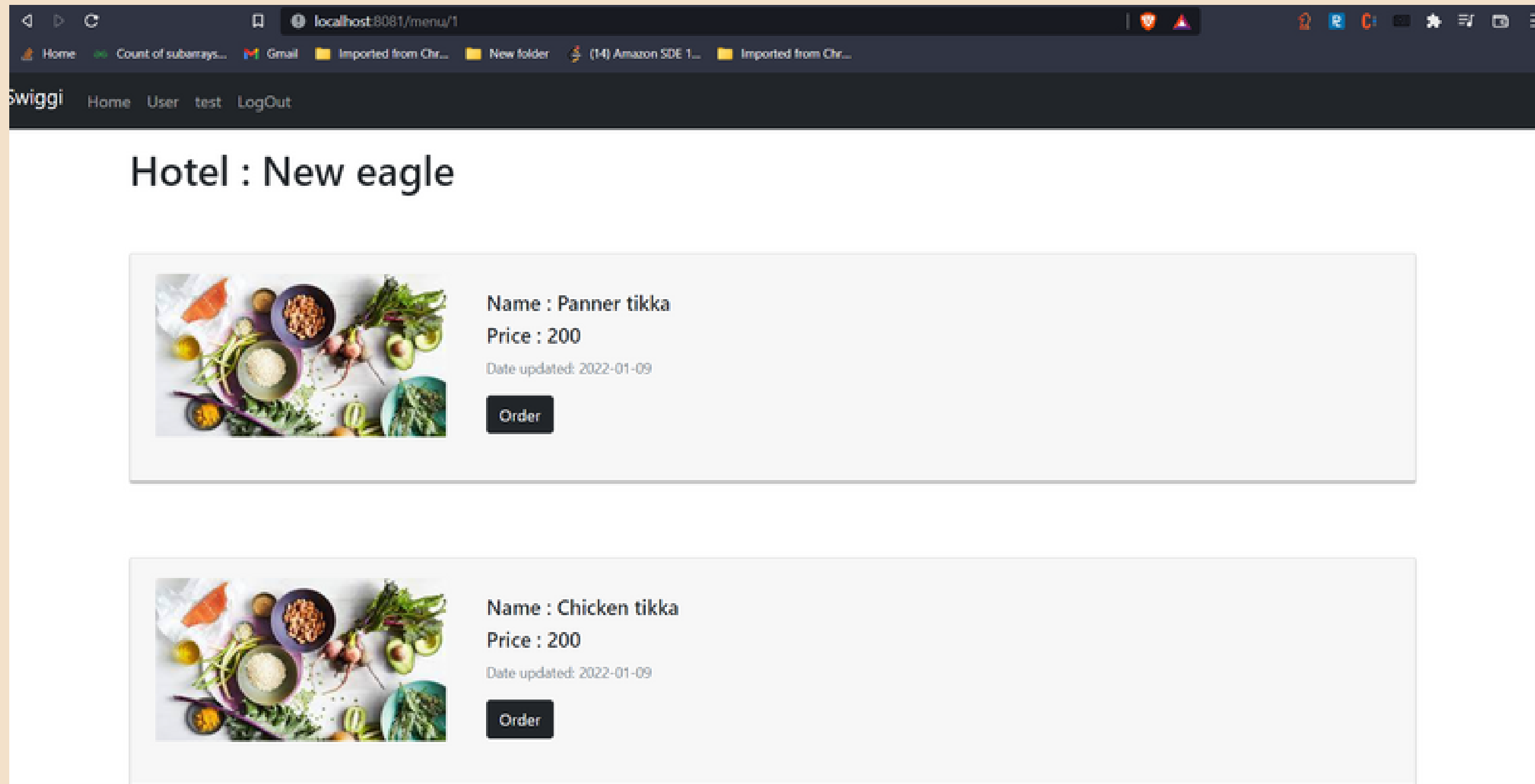
Homepage after login

It shows all the restaurant from the database



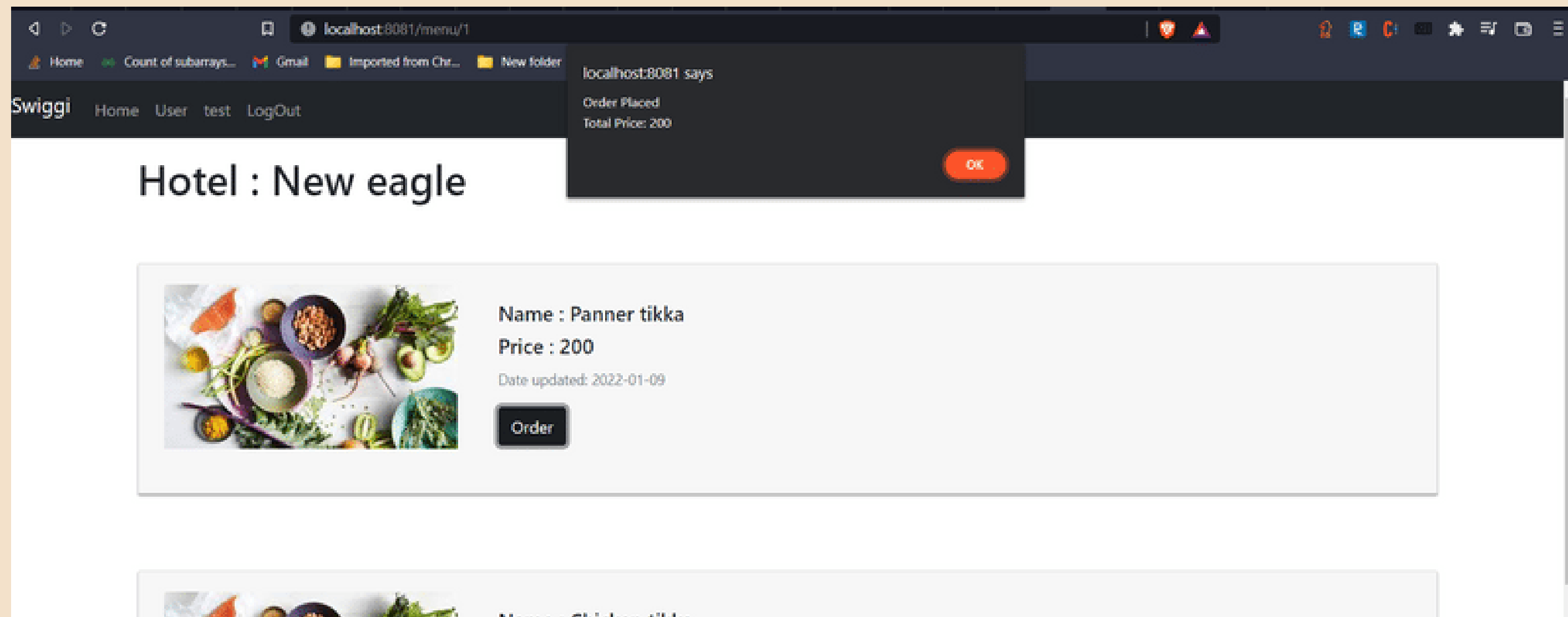
Menu

After clicking the menu option on the restaurant of your choice it will show the corresponding menu of the restaurant



Order Placed

After clicking order, total amount of the order will be shown in the alert box



Future Plans

1. Add checkout cart
2. Option for users to become moderators and add restaurant and menu
3. Store the order information in the database
4. Dashboard for the restaurant owner
5. User reviews and ratings for the food
6. Add security around the APIs
7. Add User profile option



Thank you!

I AM OPEN FOR QUESTIONS