

The screenshot shows a Visual Studio Code (VS Code) interface with the following details:

- File Explorer:** On the left, it shows the project structure for "SWEET-SHOP". The "tests/integration" folder is expanded, showing "auth.test.ts" and "sweets.test.ts".
- Editor:** The main editor area displays the code for "sweets.test.ts". The code uses Jest and supertest to test endpoints related to the sweets module.
- Terminal:** At the bottom, the terminal window shows the command line output of running Jest tests. It indicates 2 passed test suites and 4 passed tests.
- Bottom Status Bar:** Shows file paths like "main.ts", "sweets.test.ts", and "server.ts", along with status icons for tabs and the current file.

```
backend > tests > integration > TS sweets.test.ts ...
22  describe('Sweets Module', () => {
37    it('should get all sweets', async () => {
42      expect(res.status).toBe(200);
43      expect(Array.isArray(res.body)).toBe(true);
44    });
45    it('should purchase a sweet', async () => {
46      const res = await request(app)
47        .post('/api/sweets/purchase')
48        .set('Authorization', `Bearer ${token}`)
49        .send([
50          {
51            sweetId: 1, // make sure this ID exists
52            quantity: 2,
53          },
54        ]);
55      expect(res.status).toBe(200);
56      expect(res.body.message).toBe('Purchase successful');
57    });
58  });
59
```

```
PROBLEMS 4 OUTPUT DEBUG CONSOLE TERMINAL PORTS SPELL CHECKER 4
PS C:\Users\yash\sweet-shop\backend> pwd
PS C:\Users\yash\sweet-shop\backend> npx jest
>>
  PASS  tests/integration/auth.test.ts
  PASS  tests/integration/sweets.test.ts

Test suites: 2 passed, 2 total
Tests:       4 passed, 4 total
Snapshots:  0 total
Time:        3.536 s
Ran all test suites.

PS C:\Users\yash\sweet-shop\backend>
```

The screenshot shows a Visual Studio Code (VS Code) interface with the following details:

- File Explorer:** On the left, it shows the project structure under "SWEET-SHOP". The "tests/integration" folder is selected, containing "auth.test.ts" and "sweets.test.ts".
- Code Editor:** The main editor area displays "sweets.test.ts". The code is a Jest test for an RBAC Sweets Module. It includes two test cases: one for unauthorized access and one for authorized access.
- Terminal:** At the bottom, the terminal shows the output of running Jest tests. It indicates 2 passed test suites (auth and sweets) and 5 total tests passed.
- Bottom Status Bar:** Shows the current file path as "C:\Users\yashv\sweet-shop\backend", the line number as "Ln 66, Col 1", and the character position as "Spaces: 2, CRLF".

```
backend > tests > integration > sweets.test.ts > ...
29 describe('RBAC Sweets Module', () => {
44   it('USER should NOT add sweet', async () => {
45     const res = await request(app)
        .post('/api/sweets')
        .send({
          name: 'Chocolate',
          price: 10,
          quantity: 20
        });
      expect(res.status).toBe(403);
    });
  });
it('USER should get sweets', async () => {
  const res = await request(app)
    .get('/api/sweets')
    .set('Authorization', `Bearer ${userToken}`);
  expect(res.status).toBe(200);
});
});
```

```
PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL PORTS SPELL CHECKER 2
PS C:\Users\yashv\sweet-shop\backend> npx jest --runInBand
>>
PS C:\Users\yashv\sweet-shop\backend> npx jest --runInBand
>>
  PASS tests/integration/sweets.test.ts
  PASS tests/integration/auth.test.ts

Test Suites: 2 passed, 2 total
Tests:      5 passed, 5 total
Snapshots:  0 total
Time:       3.331 s
Ran all test suites.

PS C:\Users\yashv\sweet-shop\backend>
```

The screenshot shows a code editor interface with the following details:

- File Explorer:** On the left, it displays the project structure under "SWEET-SHOP". The "src" folder contains "backend", "config", "controllers", "models", "routes", and "services". "routes" contains "auth.ts" and "sweets.ts". "controllers" contains "authController.ts" and "sweetsController.ts". "models" contains "database.ts". "services" contains "types".
- Code Editor:** The main area shows a TypeScript file named "sweets.ts". The code defines a router for handling sweet creation and retrieval.

```
backend > src > routes > sweets.ts > ...
  8  router.post("/", authMiddleware, adminMiddleware, async (req, res) => {
  9    if (!name || !price) {
10      return res.status(400).json({ message: "All fields required" });
11    }
12    const sweet = await prisma.sweet.create({
13      data: { name, price },
14    });
15    res.status(201).json(sweet);
16  });
17  // USER gets sweets
18  router.get("/", authMiddleware, async (req, res) => {
19    const sweets = await prisma.sweet.findMany();
20    res.status(200).json(sweets);
21  });
22
23  export default router;
```

- Terminal:** At the bottom, the terminal shows the output of running Jest tests. It cleared the cache and ran two test suites, both of which passed.

```
PS C:\Users\yashv\sweet-shop\backend> npx jest --clearCache
>> npx jest
>>
  Cleared C:\Users\yashv\AppData\Local\Temp\jest
  PASS  tests/integration/auth.test.ts
  PASS  tests/integration/sweets.test.ts

Test Suites: 2 passed, 2 total
Tests:       6 passed, 6 total
Snapshots:  0 total
Time:        5.741 s
Ran all test suites.

PS C:\Users\yashv\sweet-shop\backend>
```

The screenshot shows the Postman application interface. On the left, there's a sidebar with icons for Collections, Environments, History, Flows, and Files (BETA). The main workspace is titled "YashVeer Singh's Workspace". A search bar at the top right says "Search Postman". The top navigation bar includes "Home", "Workspaces", "API Network", "Invite", and "Upgrade".

In the center, a collection named "SweetShop" is selected. A sub-collection named "Get data" is shown. A POST request is being prepared to "http://localhost:5000/api/auth/register". The "Body" tab is selected, showing the following JSON payload:

```
1 {  
2   "name": "Yashveer Singh",  
3   "email": "yashveer.singh3535@gmail.com",  
4   "password": "3535"  
5 }  
6
```

Below the body, the response is displayed as a 201 Created status with 179 ms and 302 B. The response body is also JSON:

```
1 {  
2   "id": 17,  
3   "email": "yashveer.singh3535@gmail.com",  
4   "role": "USER"  
5 }
```

At the bottom, there are links for "Cloud View", "Find and replace", "Console", "Terminal", "Runner", "Start Proxy", "Cookies", "Vault", "Trash", and "Help".

The screenshot shows the Postman application interface. On the left, the sidebar includes 'Collections' (YashVeer Singh's Workspace), 'Environments' (SweetShop selected), 'History', 'Flows', and 'Files (BETA)'. The main area displays a 'POST Get data' collection with a single 'Get data' request. The request details are as follows:

- Method:** POST
- URL:** <http://localhost:5000/api/auth/login>
- Body:**  raw
- JSON Body:**

```
1 {  
2   "name": "Yashveer Singh",  
3   "email": "yashveer.singh3535@gmail.com",  
4   "password": "3535"  
5 }  
6
```

The response section shows a successful 200 OK status with a response time of 126 ms and a response size of 411 B. The response body is displayed as JSON:

```
1 {  
2   "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.  
3       eyJpc2VydWQiOjE3LC3yb2xlijo1VWNFUiIismIhdCI6MTc2NTc0NDkwNCwiZXhwIjoxNzY1NzQ4NTA0fQ.  
4       iLElfjwPURnGg3dvZ0BSY5mEH004rtjhbdkVPI5NIY"  
5 }
```

The screenshot shows the Postman application interface. On the left, there's a sidebar with icons for Collections, Environments, History, Flows, and Files (BETA). The main workspace is titled "YashVeer Singh's Workspace". A search bar at the top right says "Search Postman". The top navigation bar includes "Home", "Workspaces", "API Network", "Invite", "Upgrade", and other standard browser controls.

In the center, a collection named "SweetShop" is selected. A specific request named "Get data" is shown under the "My Collection / Get data" section. The request method is "GET" and the URL is "http://localhost:5000/api/sweets".

The "Headers" tab is active, showing one header entry:

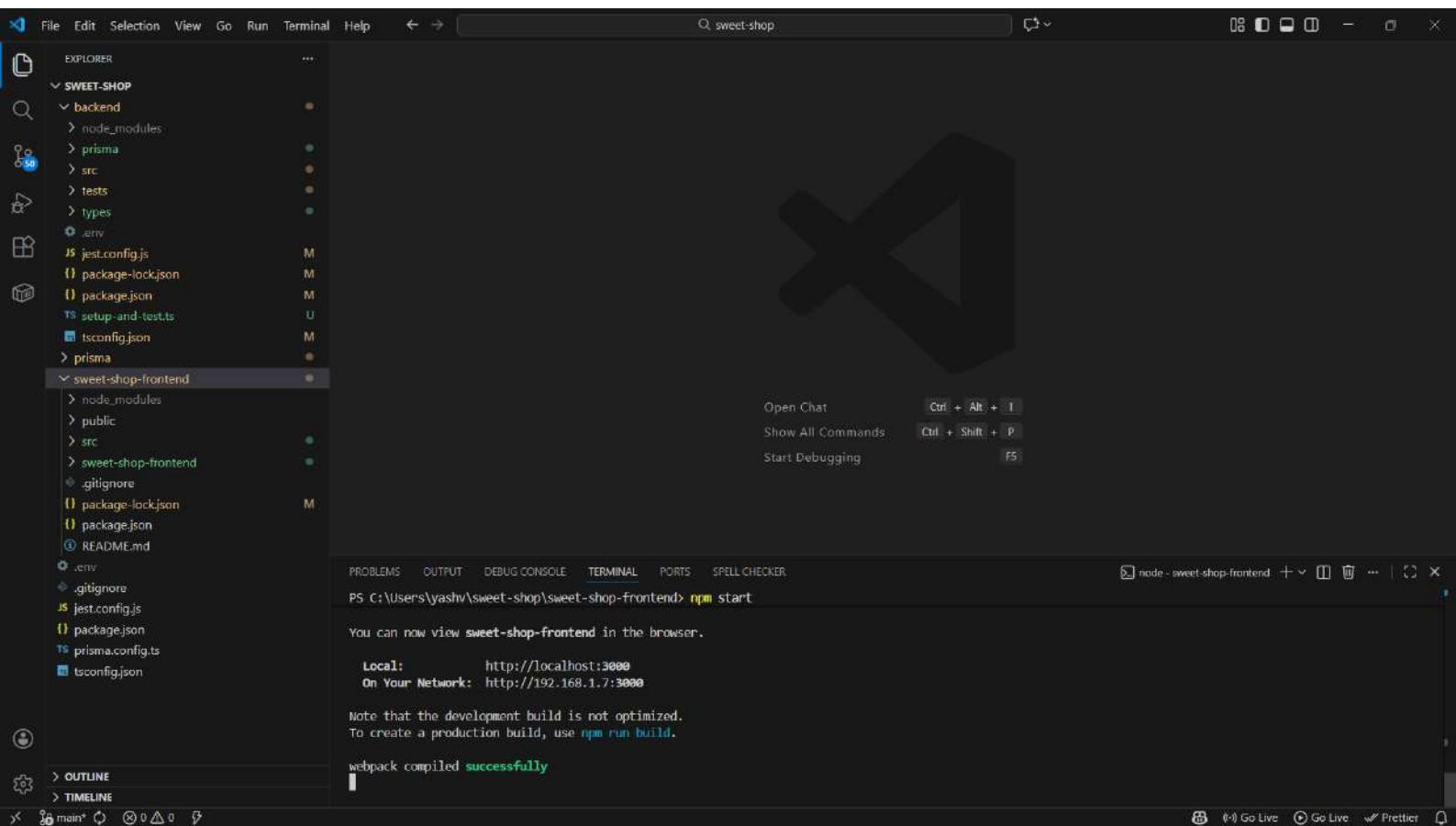
Key	Value	Description
Authorization	Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6Ikp...	
Key	Value	Description

Below the headers, the "Body" tab is active, displaying a JSON response:

```
1 [  
2   {  
3     "id": 3,  
4     "name": "Chocolate",  
5     "price": 50  
6   }  
7 ]
```

The status bar at the bottom shows "200 OK" with a timestamp of "15 ms" and a size of "275 B". Other options like "Save Response" and "Visualize" are also present.

At the very bottom, there are links for "Cloud View", "Find and replace", "Console", "Terminal", "Runner", "Start Proxy", "Cookies", "Vault", "Trash", and "Help".



Frontend design suggestions Sweet Shop localhost:3000

Sweet Shop Home Login Register

## Our Delicious Sweets 🎂

Search sweets...

Sweet Type	Name	Price	
Chocolate Cake	Gulab Jamun	Laddu	Cupcake
₹ 250	₹ 50	₹ 30	₹ 100
Add to Cart	Add to Cart	Add to Cart	Add to Cart



## Register

Name

Email

Password

**Register**

Frontend design suggestions × Sweet Shop × +

localhost:3000/register

Gmail Maps YouTube News Translate All Bookmarks

Sweet Shop Home Login Register

## Register

yashveer

yashveer.singh3535@gmail.com

....

Register

Frontend design suggestions Sweet Shop localhost:3000

Home Login Register

## Our Delicious Sweets 🎂

Search sweets...

 Chocolate Cake ₹ 250 <a href="#">Add to Cart</a>	 Gulab Jamun ₹ 50 <a href="#">Add to Cart</a>	 Laddu ₹ 30 <a href="#">Add to Cart</a>	 Cupcake ₹ 100 <a href="#">Add to Cart</a>
--	--	--	---

**Cart 🛒**

Laddu - ₹30  
Cupcake - ₹100  
Gulab Jamun - ₹50  
Chocolate Cake - ₹250  
Total: ₹430

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** On the left, it shows the project structure for "SWEET-SHOP". The "components" folder contains "Navbar.jsx", "ProtectedRoute.jsx", and "SweetCard.jsx".
- Editor:** The main editor area displays the content of "Navbar.jsx".

```
// Navbar.jsx
// ...
import React from "react"; // 7.9k (gzipped: 3.1k)
import { Link } from "react-router-dom"; // 197.3k (gzipped: 62.3k)

function Navbar() {
  return (
    <nav className="bg-pink-400 text-white p-4 flex justify-between items-center shadow-lg sticky top-0 z-50">
      <Link to="/" className="text-2xl font-bold hover:text-yellow-200 transition">Sweet Shop </Link>
      <div className="space-x-4">
        <Link to="/" className="hover:text-yellow-200 transition">Home</Link>
        <Link to="/login" className="hover:text-yellow-200 transition">Login</Link>
        <Link to="/register" className="hover:text-yellow-200 transition">Register</Link>
      </div>
    </nav>
  );
}

export default Navbar; // Make sure it's default export
```
- Terminal:** At the bottom, the terminal shows the command "npm start" and the local host URL "http://localhost:3000". It also includes a note about the development build being not optimized and instructions to use "npm run build".
- Status Bar:** The status bar at the bottom right shows "Ln 19, Col 1", "Spaces: 2", "UTF-B", "CRLF", "Babel JavaScript", and icons for Go Live and Prettier.