

Day 3 - API Integration Report – FOOD TUCK

API INTEGRATION PROCESS

1. Environment Setup:

- Install Sanity: `npm create sanity@latest -- --template clean --create-project "learning-sanity-project" --dataset production`

- Setting Up Environment Variables

Set up our environment variables. Create a `.env.local` file in your project root. Add the following variables:

`NEXT_PUBLIC_SANITY_PROJECT_ID=your_project_id`

`NEXT_PUBLIC_SANITY_DATASET=production`

`SANITY_API_TOKEN=your_sanity_token`

- Data Fetching:

Used Axios to retrieve product data from an external API (foods and chefs).

Accessed the following Endpoints:

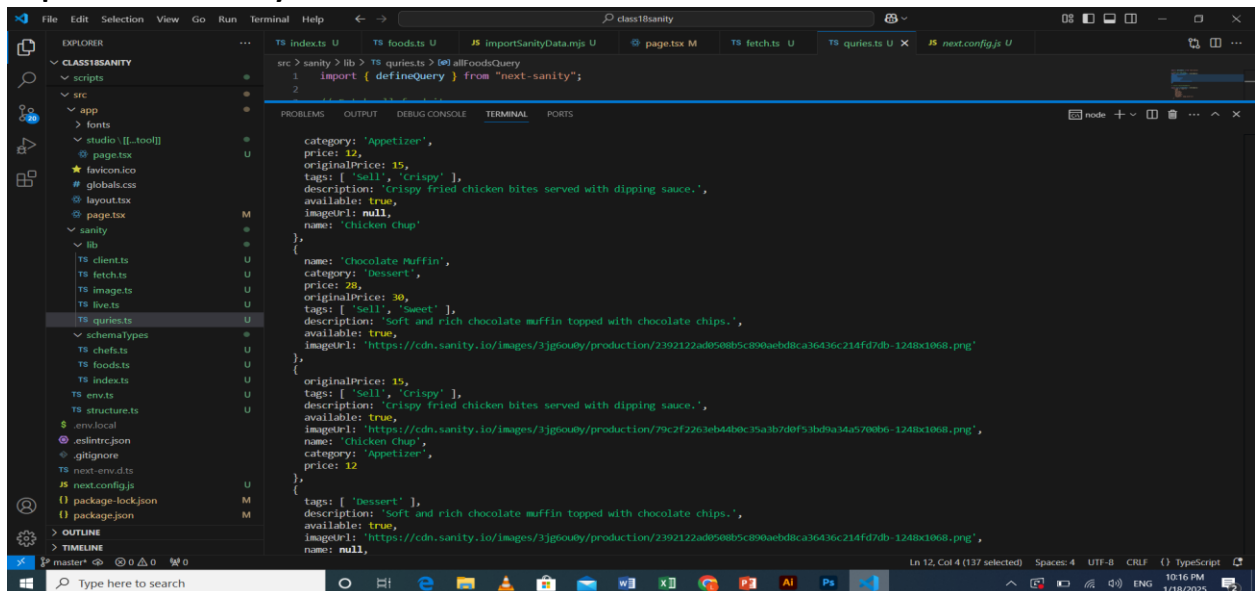
<https://sanity-nextjs-rouge.vercel.app/api/foods>

<https://sanity-nextjs-rouge.vercel.app/api/chefs>

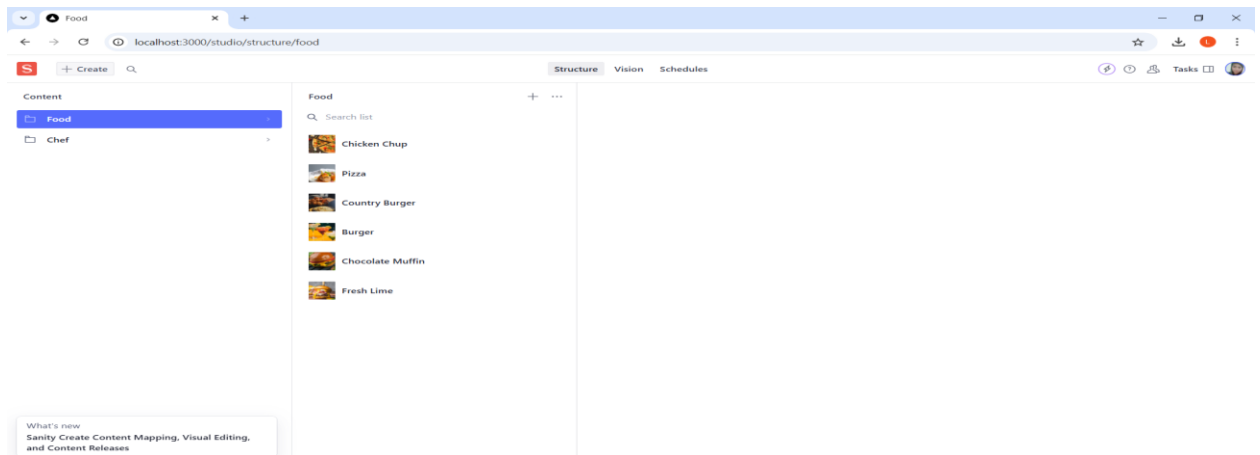
2. Adjustments made to schemas



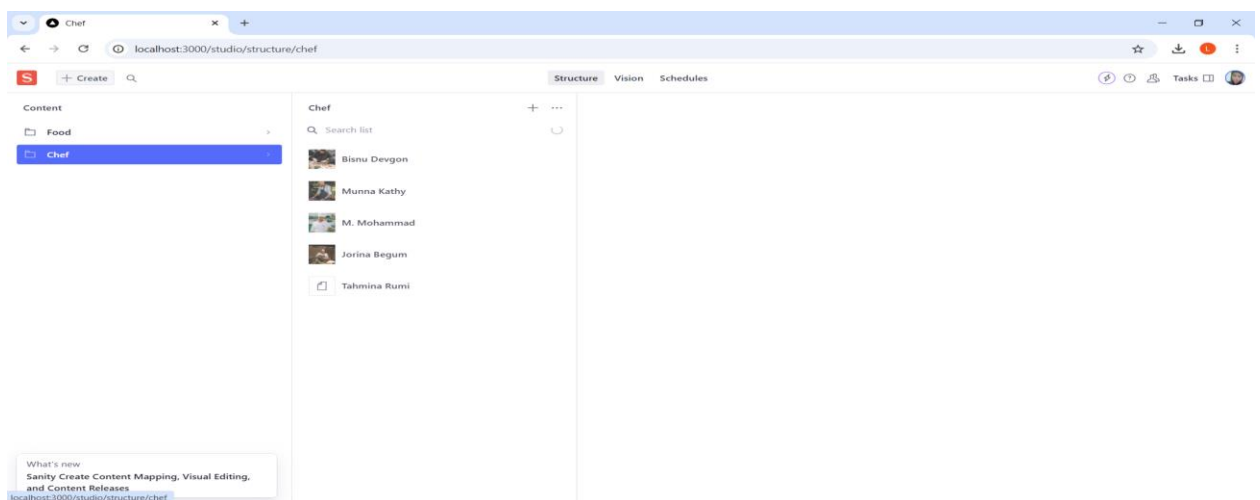
3. Import data to sanity:



Output: Finally show data. (Food)

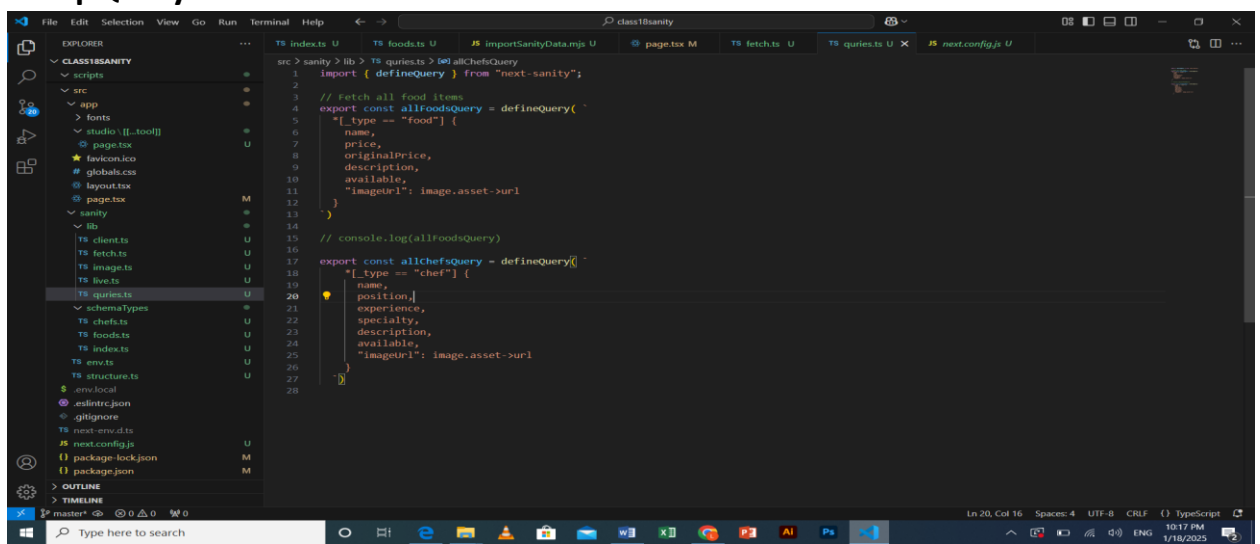


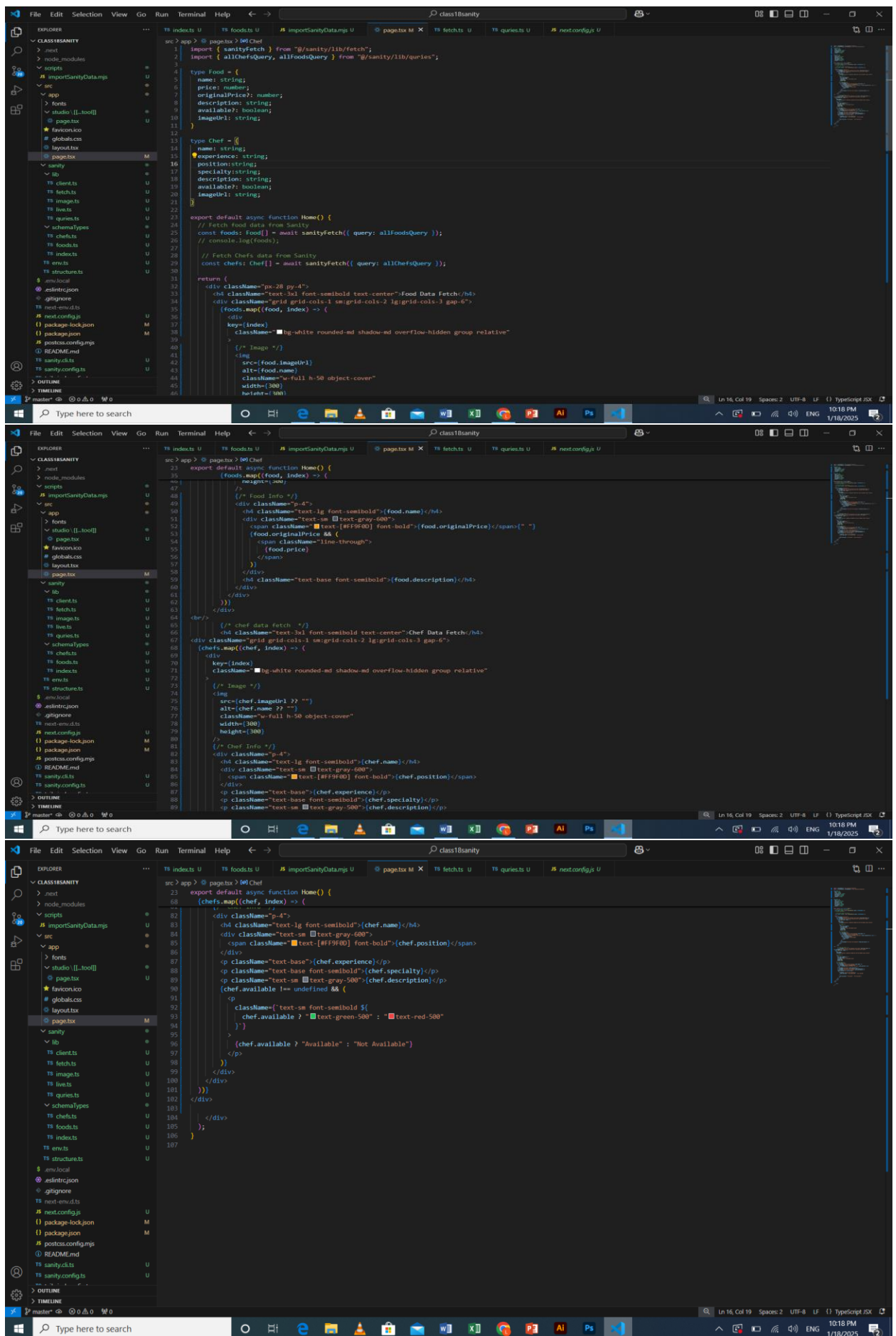
Chefs data:



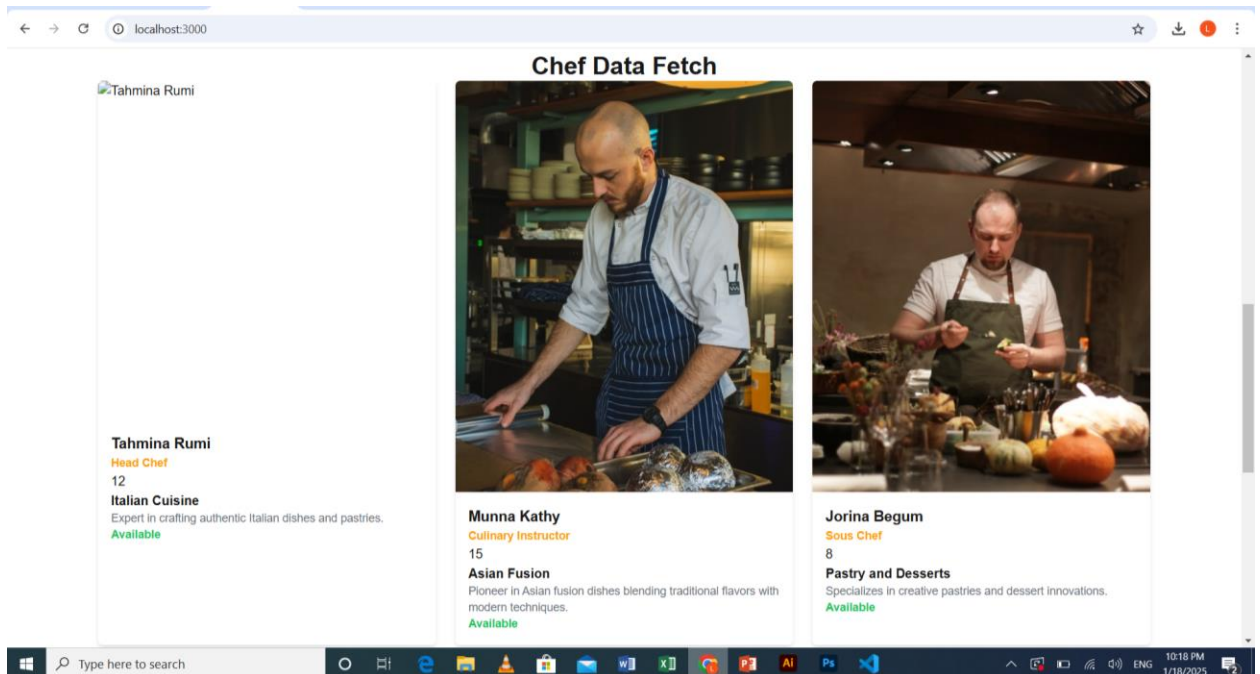
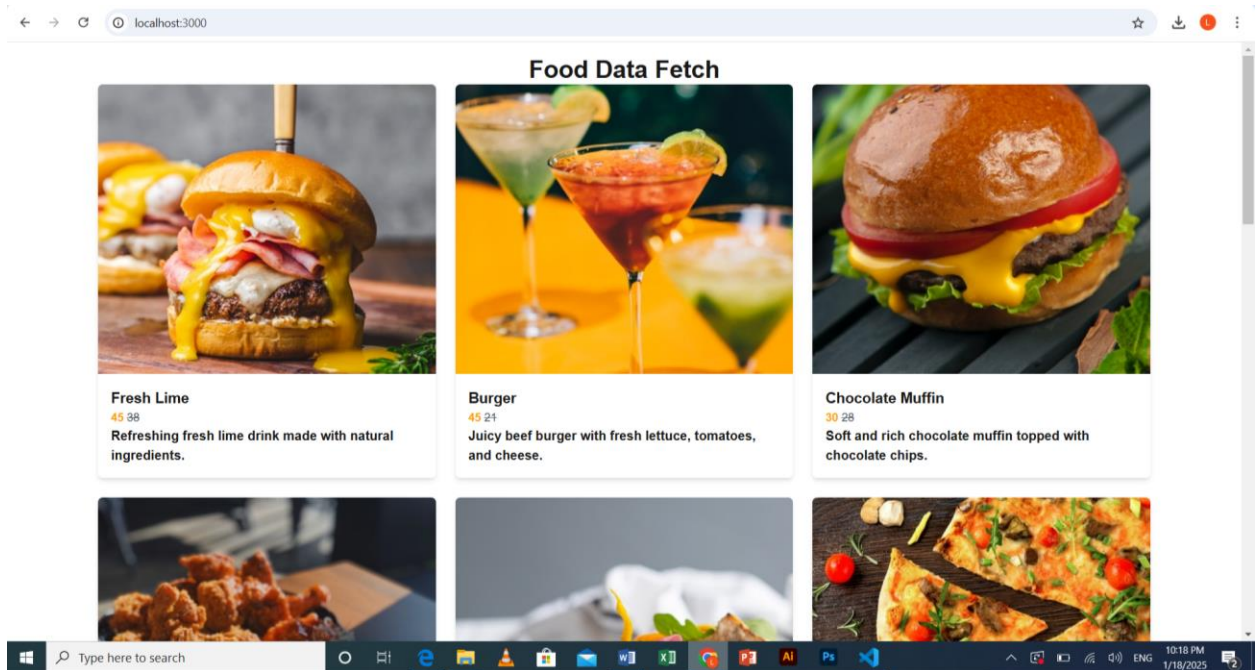
Data successfully displayed in the frontend:

Groq Query:





Made By: Mahnoor (00070019) Sunday 9 to 12



Goal of this task:

- Set up a connection with Sanity CMS to manage product data easily and effectively.
- Use GROQ queries to fetch organized data from Sanity dynamically.
- Modify the Sanity schema to match the changing needs of the API.
- Move product data from an external API into Sanity CMS, including uploading images.
- Test and confirm the integration to make sure data flows smoothly between the backend and frontend.