DAY 03 – API INTEGRATION REPORT OF FOOD TUCK

Process Of API Integration:

Overview: The AP integration is connects an external API providing foods and chefs data to a Sanity CMS Project.

Environment Setup:

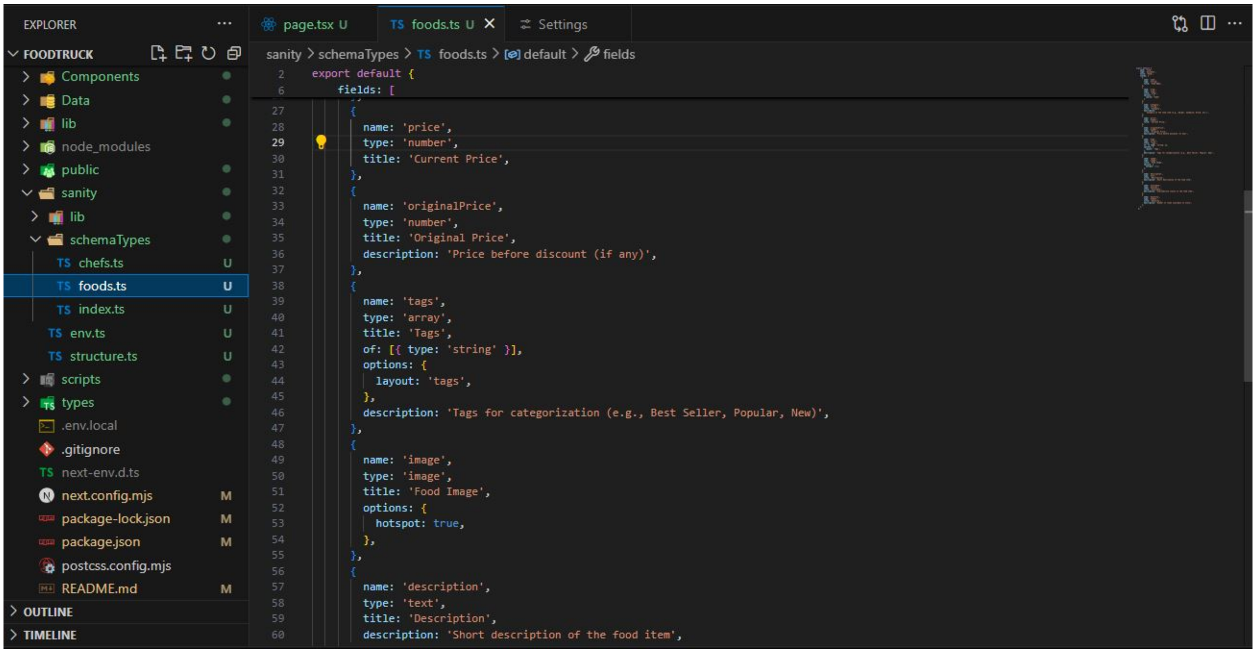
- Install Sanity (CMS) in my project.
- **Set up our environment variables:** By creating .env.local file in our root directory.
- **Include key variables :**
  1. NEXT_PUBLIC_PROJECT_ID=“”
  2. NEXT_PUBLIC_DATASET= “”
  3. SANITY_API_TOKEN= “”

Data Fecting:

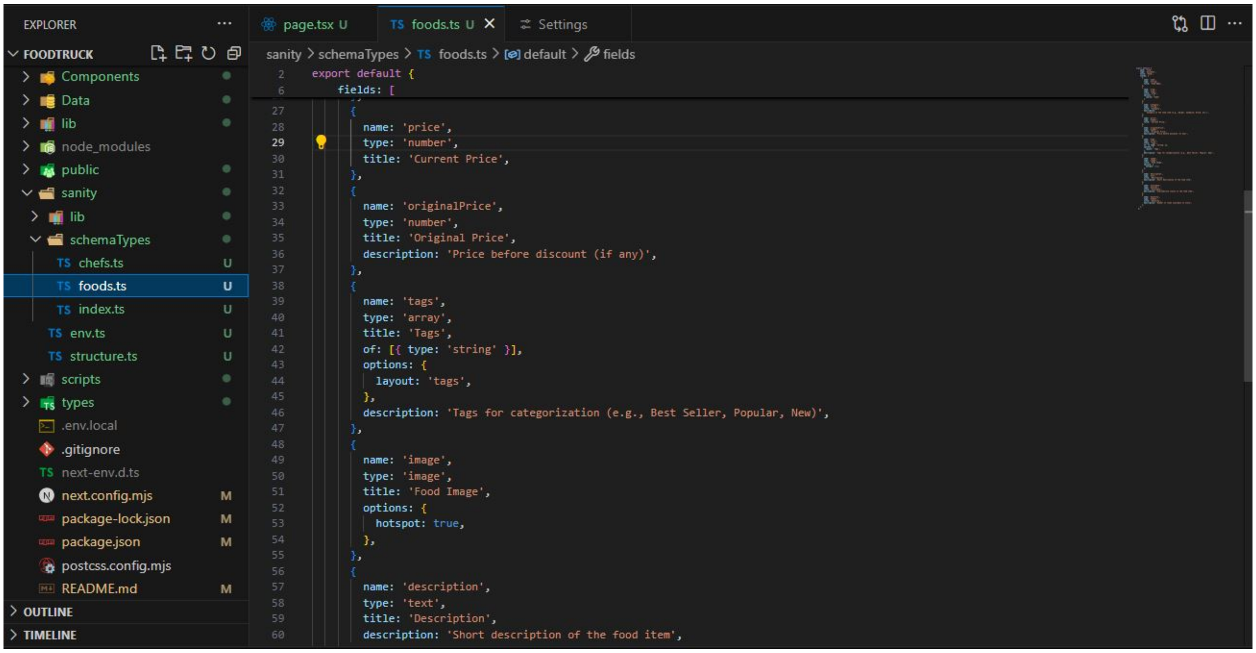
- Download axios and dotenv package and import it in import-data.mjs file to fetch food and chef data from an external API.
- Following are the provided API Endpoints:
  https://sanity-nextjs-rouge.vercel.app/api/foods
  https://sanity-nextjs-rouge.vercel.app/api/chefs

Adjustments made to schemas:

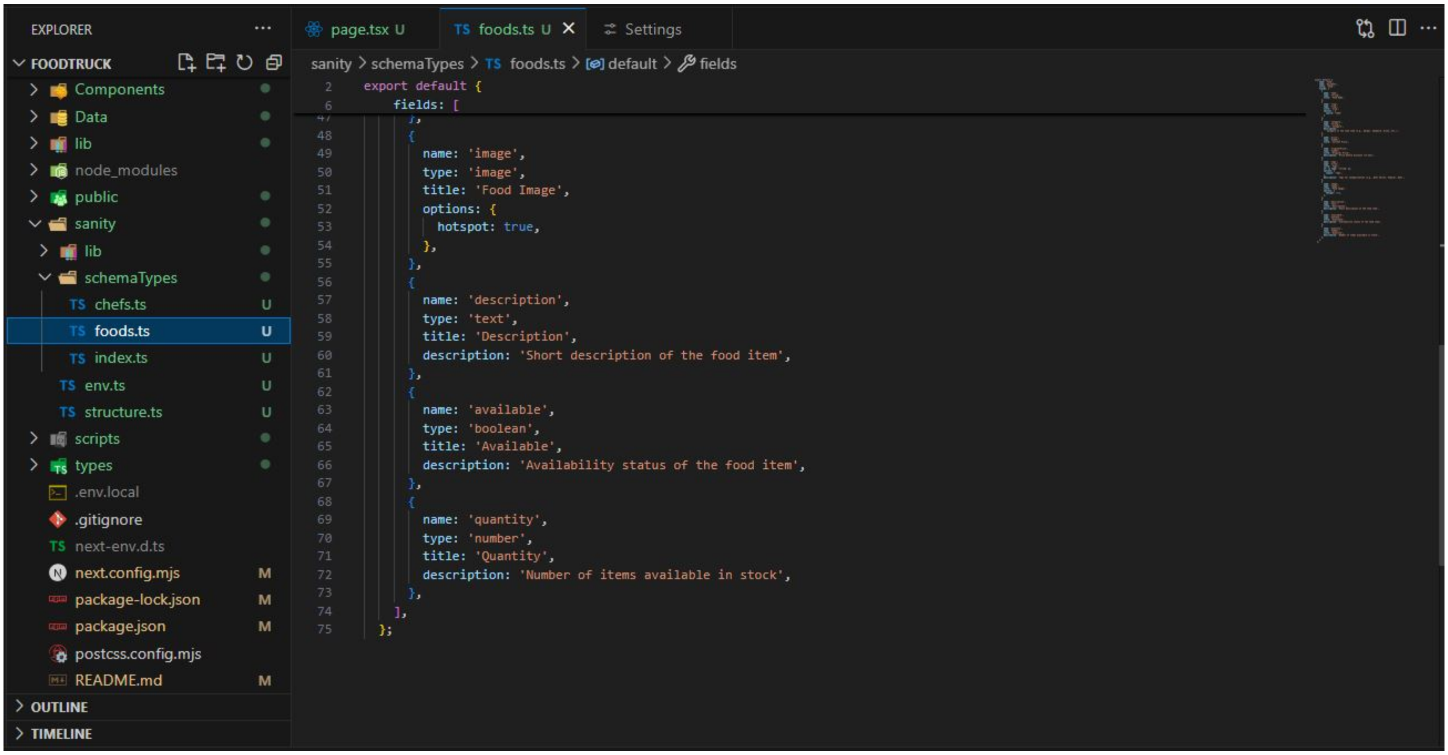
ADJUSTMENTS MADE TO SCHEMA:



```
export default {
    fields: [
      {
        name: 'price',
        type: 'number',
        title: 'Current Price',
      },
      {
        name: 'originalPrice',
        type: 'number',
        title: 'Original Price',
        description: 'Price before discount (if any)',
      },
      {
        name: 'tags',
        type: 'array',
        title: 'Tags',
        of: [{ type: 'string' }],
        options: {
          layout: 'tags',
        },
        description: 'Tags for categorization (e.g., Best Seller, Popular, New)',
      },
      {
        name: 'image',
        type: 'image',
        title: 'Food Image',
        options: {
          hotspot: true,
        },
      },
      {
        name: 'description',
        type: 'text',
        title: 'Description',
        description: 'Short description of the food item',
```

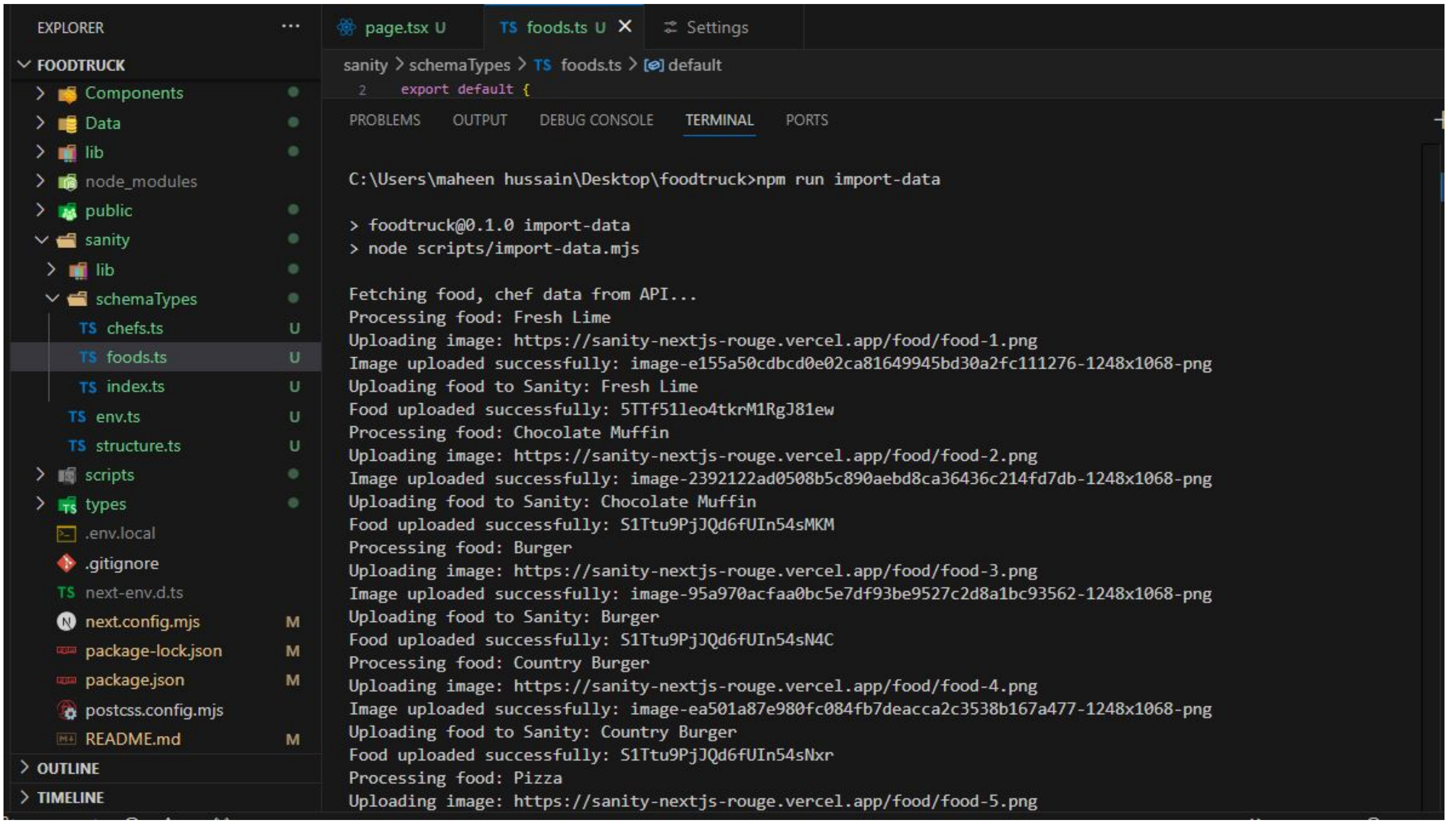


```
export default {
    fields: [
      {
        name: 'price',
        type: 'number',
        title: 'Current Price',
      },
      {
        name: 'originalPrice',
        type: 'number',
        title: 'Original Price',
        description: 'Price before discount (if any)',
      },
      {
        name: 'tags',
        type: 'array',
        title: 'Tags',
        of: [{ type: 'string' }],
        options: {
          layout: 'tags',
        },
        description: 'Tags for categorization (e.g., Best Seller, Popular, New)',
      },
      {
        name: 'image',
        type: 'image',
        title: 'Food Image',
        options: {
          hotspot: true,
        },
      },
      {
        name: 'description',
        type: 'text',
        title: 'Description',
        description: 'Short description of the food item',
```

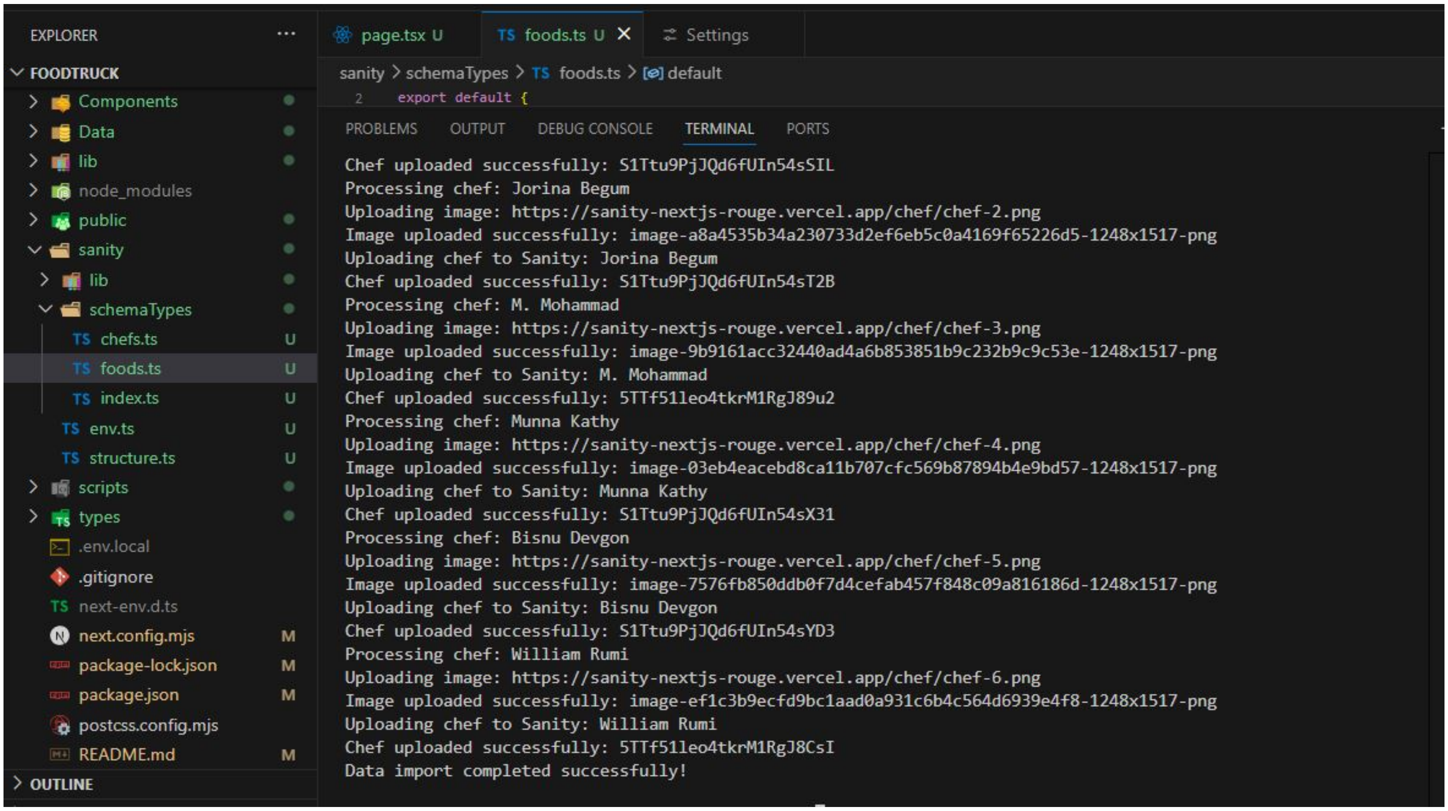


```
export default {
  fields: [
    },
    {
      name: 'image',
      type: 'image',
      title: 'Food Image',
      options: {
        hotspot: true,
      },
    },
    {
      name: 'description',
      type: 'text',
      title: 'Description',
      description: 'Short description of the food item',
    },
    {
      name: 'available',
      type: 'boolean',
      title: 'Available',
      description: 'Availability status of the food item',
    },
    {
      name: 'quantity',
      type: 'number',
      title: 'Quantity',
      description: 'Number of items available in stock',
    },
  ],
};
```

Import data to sanity:



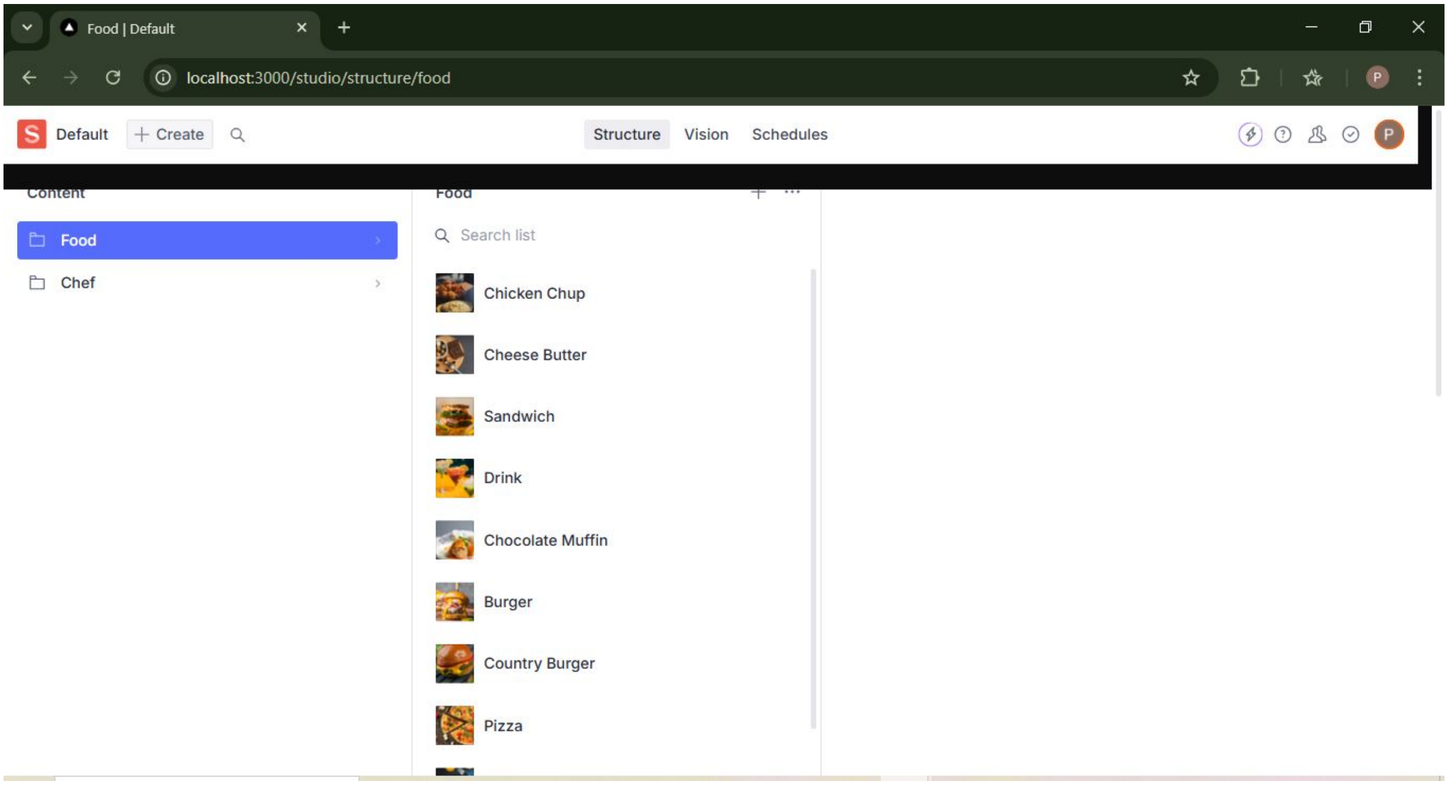
```
C:\Users\maheen hussain\Desktop\foodtruck>npm run import-data

> foodtruck@0.1.0 import-data
> node scripts/import-data.mjs

Fetching food, chef data from API...
Processing food: Fresh Lime
Uploading image: https://sanity-nextjs-rouge.vercel.app/food/food-1.png
Image uploaded successfully: image-e155a50cdbcd0e02ca81649945bd30a2fc111276-1248x1068-png
Uploading food to Sanity: Fresh Lime
Food uploaded successfully: 5TTf51leo4tkrM1RgJ81ew
Processing food: Chocolate Muffin
Uploading image: https://sanity-nextjs-rouge.vercel.app/food/food-2.png
Image uploaded successfully: image-2392122ad0508b5c890aebd8ca36436c214fd7db-1248x1068-png
Uploading food to Sanity: Chocolate Muffin
Food uploaded successfully: S1Ttu9PjJQd6fUIn54sMKM
Processing food: Burger
Uploading image: https://sanity-nextjs-rouge.vercel.app/food/food-3.png
Image uploaded successfully: image-95a970acfaa0bc5e7df93be9527c2d8a1bc93562-1248x1068-png
Uploading food to Sanity: Burger
Food uploaded successfully: S1Ttu9PjJQd6fUIn54sN4C
Processing food: Country Burger
Uploading image: https://sanity-nextjs-rouge.vercel.app/food/food-4.png
Image uploaded successfully: image-ea501a87e980fc084fb7deacca2c3538b167a477-1248x1068-png
Uploading food to Sanity: Country Burger
Food uploaded successfully: S1Ttu9PjJQd6fUIn54sNxr
Processing food: Pizza
Uploading image: https://sanity-nextjs-rouge.vercel.app/food/food-5.png
```



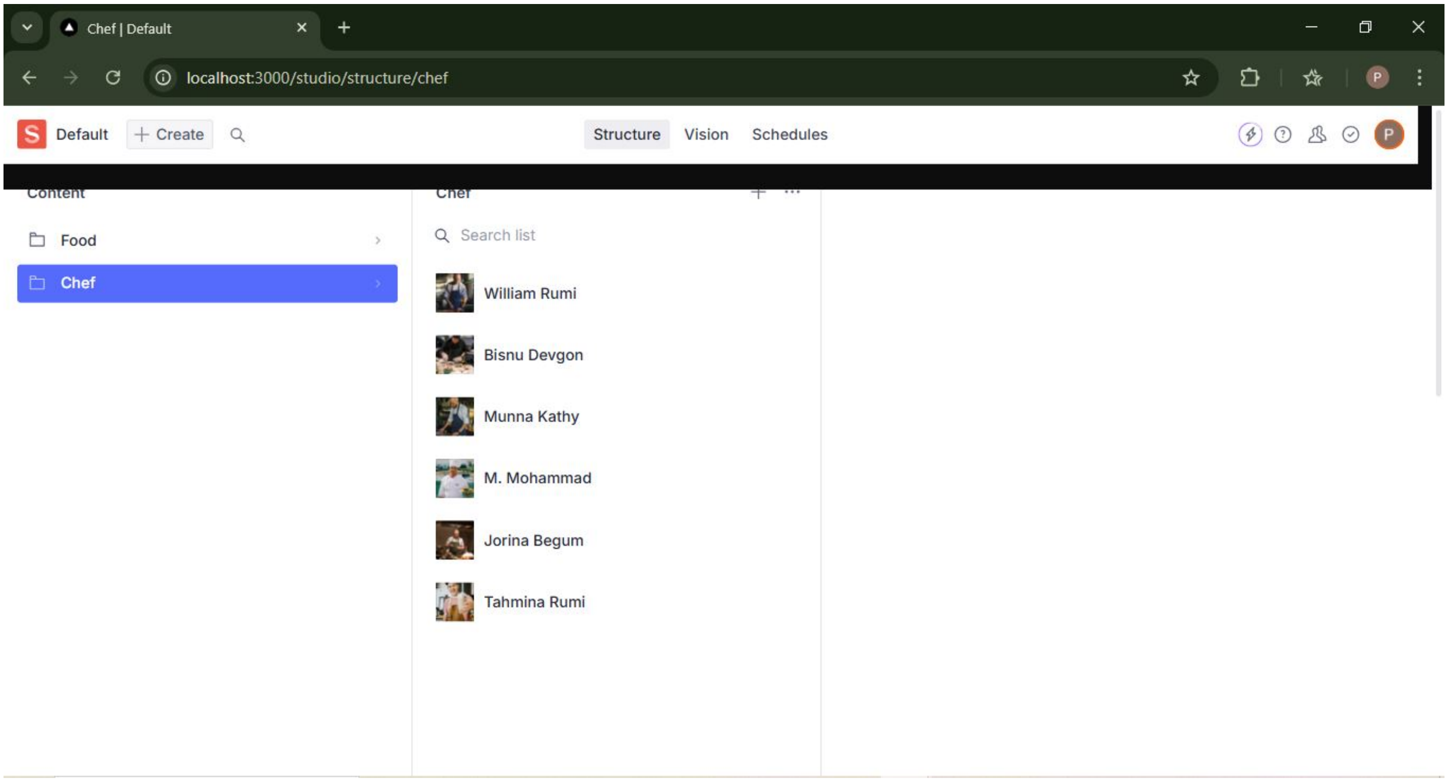
```
export default {
```

```
Chef uploaded successfully: S1Ttu9PjJQd6fUIn54sSIL
Processing chef: Jorina Begum
Uploading image: https://sanity-nextjs-rouge.vercel.app/chef/chef-2.png
Image uploaded successfully: image-a8a4535b34a230733d2ef6eb5c0a4169f65226d5-1248x1517-png
Uploading chef to Sanity: Jorina Begum
Chef uploaded successfully: S1Ttu9PjJQd6fUIn54sT2B
Processing chef: M. Mohammad
Uploading image: https://sanity-nextjs-rouge.vercel.app/chef/chef-3.png
Image uploaded successfully: image-9b9161acc32440ad4a6b853851b9c232b9c9c53e-1248x1517-png
Uploading chef to Sanity: M. Mohammad
Chef uploaded successfully: 5TTf51leo4tkrM1RgJ89u2
Processing chef: Munna Kathy
Uploading image: https://sanity-nextjs-rouge.vercel.app/chef/chef-4.png
Image uploaded successfully: image-03eb4eacebd8ca11b707cfc569b87894b4e9bd57-1248x1517-png
Uploading chef to Sanity: Munna Kathy
Chef uploaded successfully: S1Ttu9PjJQd6fUIn54sX31
Processing chef: Bisnu Devgon
Uploading image: https://sanity-nextjs-rouge.vercel.app/chef/chef-5.png
Image uploaded successfully: image-7576fb850ddb0f7d4cefab457f848c09a816186d-1248x1517-png
Uploading chef to Sanity: Bisnu Devgon
Chef uploaded successfully: S1Ttu9PjJQd6fUIn54sYD3
Processing chef: William Rumi
Uploading image: https://sanity-nextjs-rouge.vercel.app/chef/chef-6.png
Image uploaded successfully: image-ef1c3b9ecfd9bc1aad0a931c6b4c564d6939e4f8-1248x1517-png
Uploading chef to Sanity: William Rumi
Chef uploaded successfully: 5TTf51leo4tkrM1RgJ8CsI
Data import completed successfully!
```

Output: Finally Data Showed in Sanity. (Food Data)



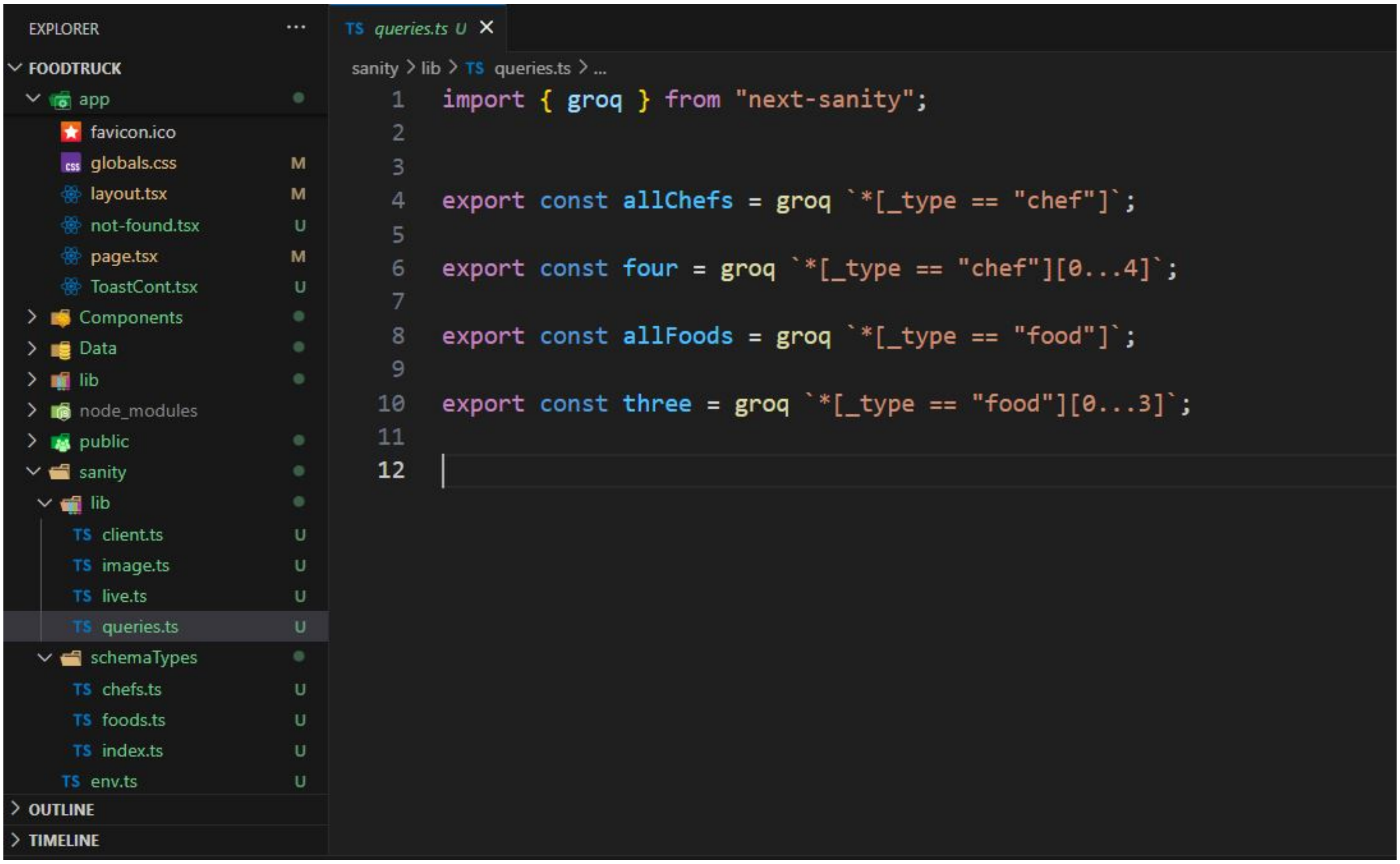
CHEFS DATA:



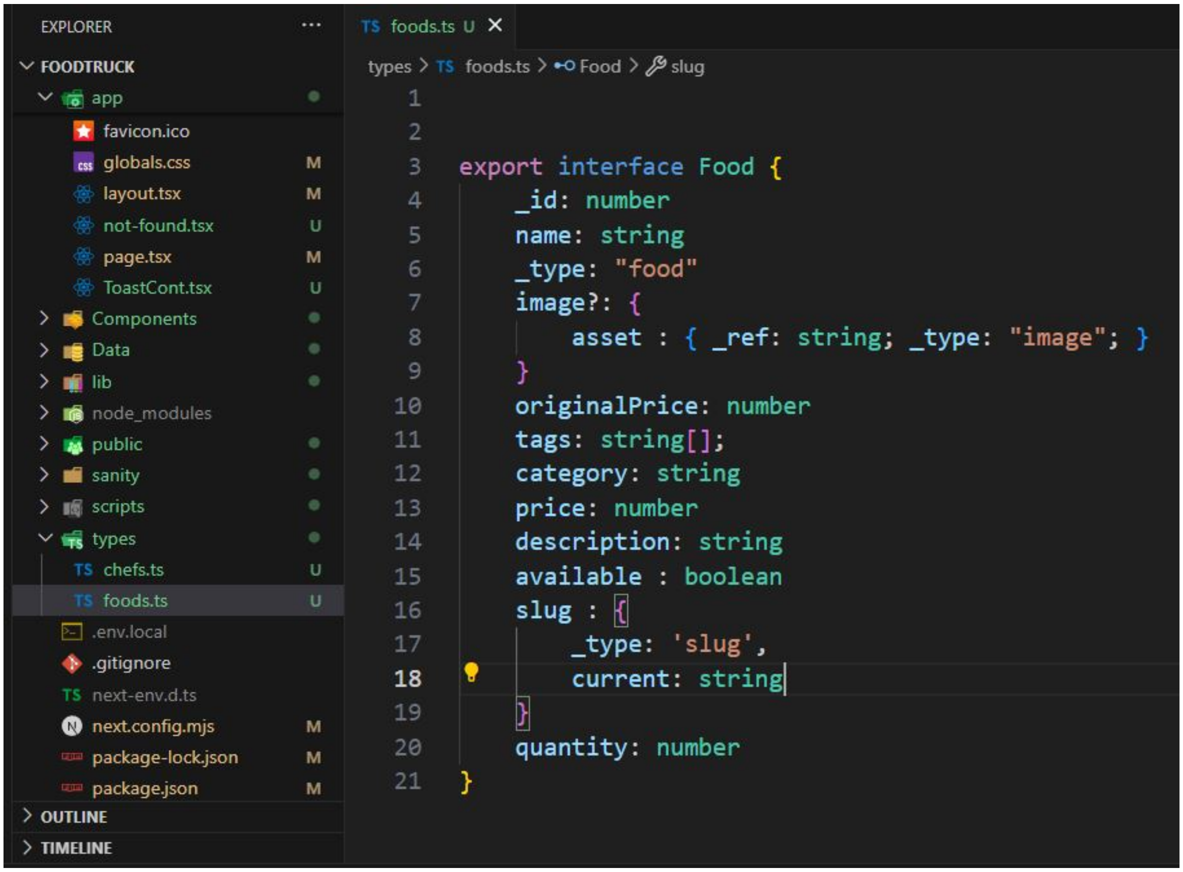
Chef

- William Rumi
- Bisnu Devgon
- Munna Kathy
- M. Mohammad
- Jorina Begum
- Tahmina Rumi

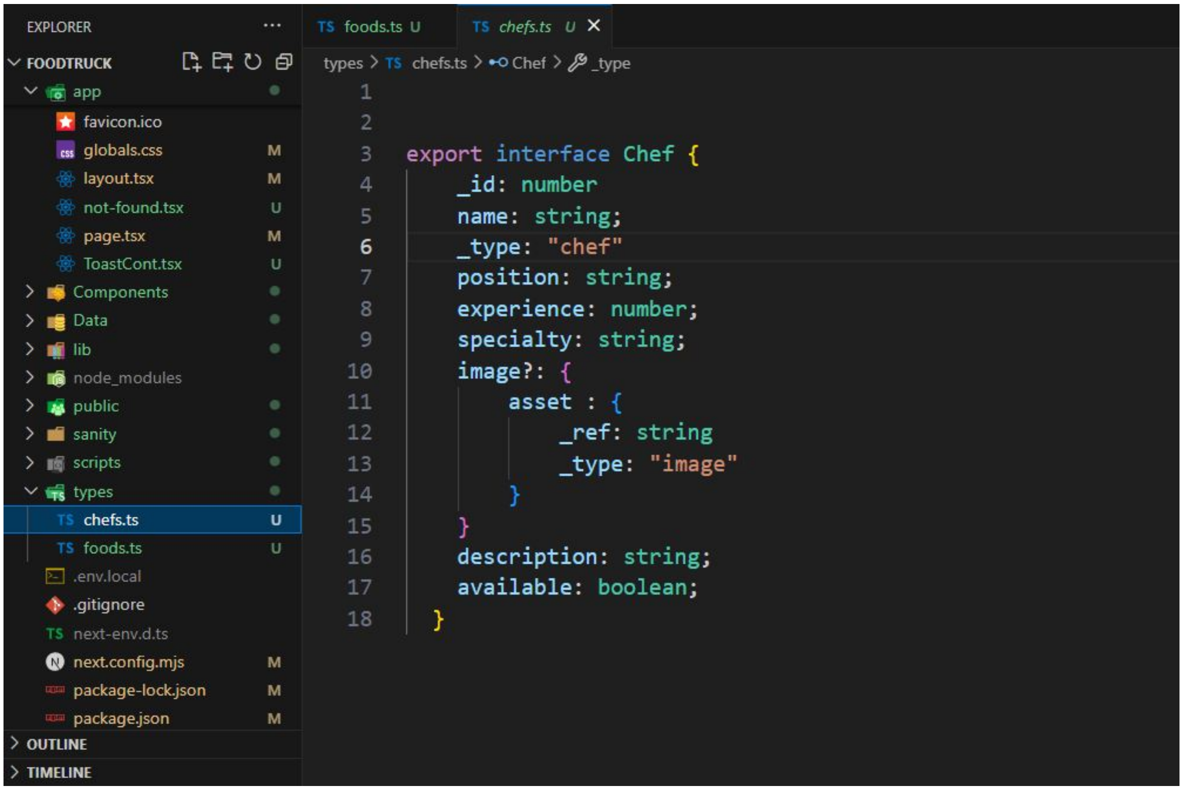
GROQ QUERY:



```
import { groq } from "next-sanity";


export const allChefs = groq `*[_type == "chef"]`;

export const four = groq `*[_type == "chef"][0...4]`;

export const allFoods = groq `*[_type == "food"]`;

export const three = groq `*[_type == "food"][0...3]`;

```

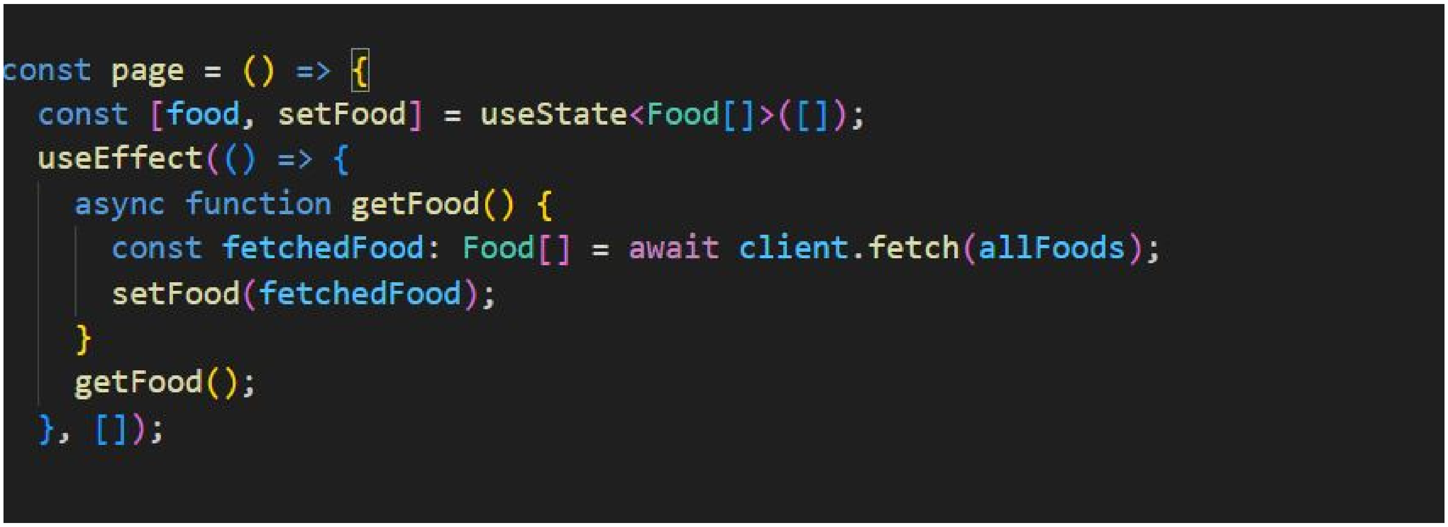


```ts


export interface Food {
    _id: number
    name: string
    _type: "food"
    image?: {
        asset : { _ref: string; _type: "image"; }
    }
    originalPrice: number
    tags: string[];
    category: string
    price: number
    description: string
    available : boolean
    slug : {
        _type: 'slug',
        current: string
    }
    quantity: number
}
```

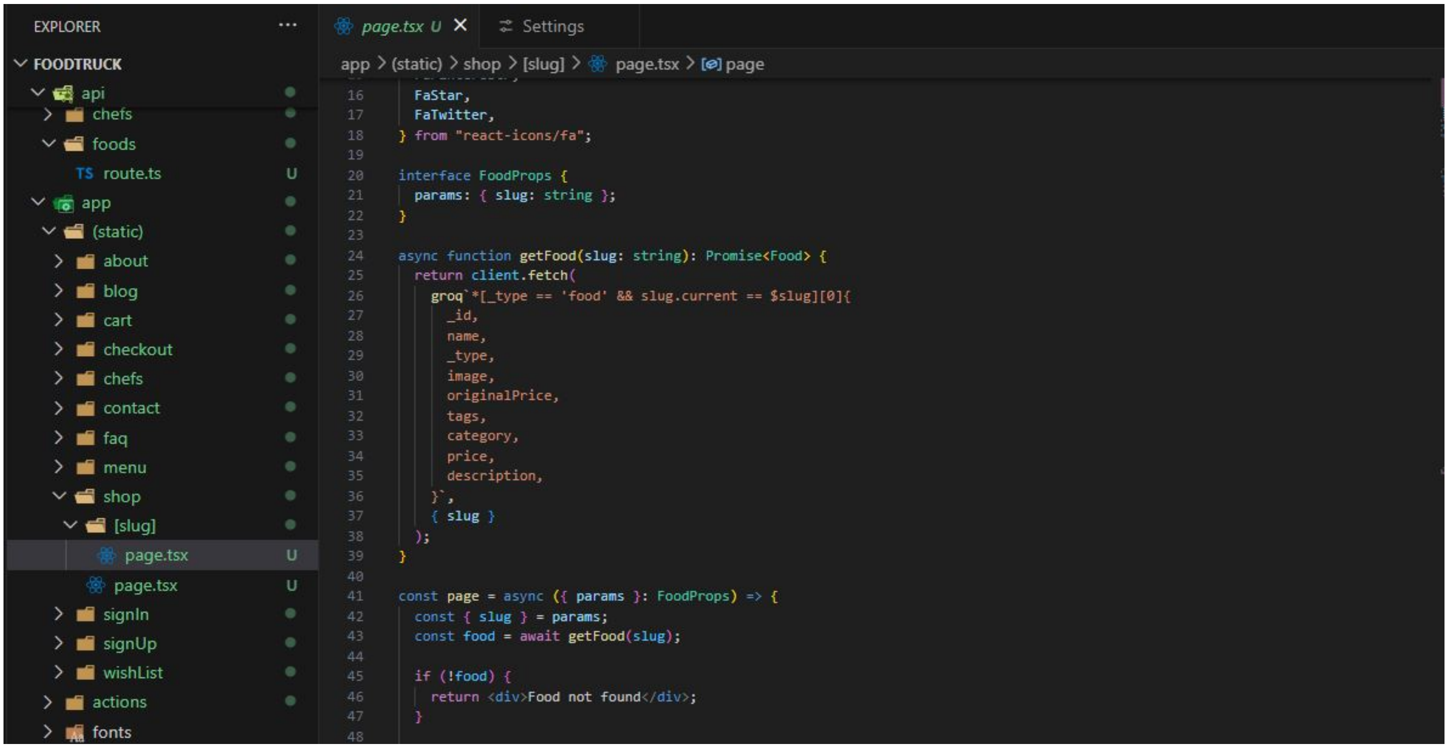


```ts


export interface Chef {
    _id: number
    name: string;
    _type: "chef"
    position: string;
    experience: number;
    specialty: string;
    image?: {
        asset : {
            _ref: string
            _type: "image"
        }
    }
    description: string;
    available: boolean;
 }
```

Displaying Data on the frontend:

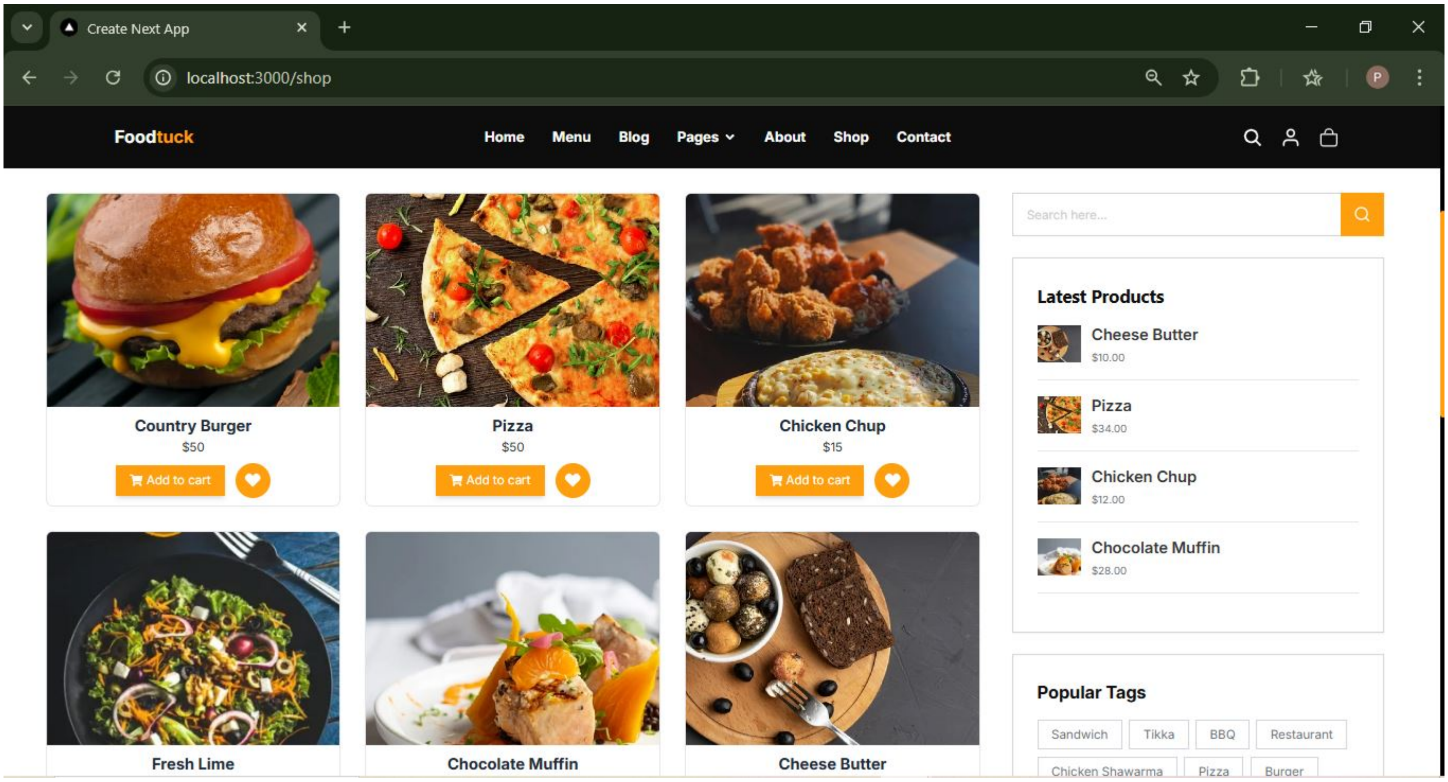


```
const page = () => {
  const [food, setFood] = useState<Food[]>([]);
  useEffect(() => {
    async function getFood() {
      const fetchedFood: Food[] = await client.fetch(allFoods);
      setFood(fetchedFood);
    }
    getFood();
  }, []);
```

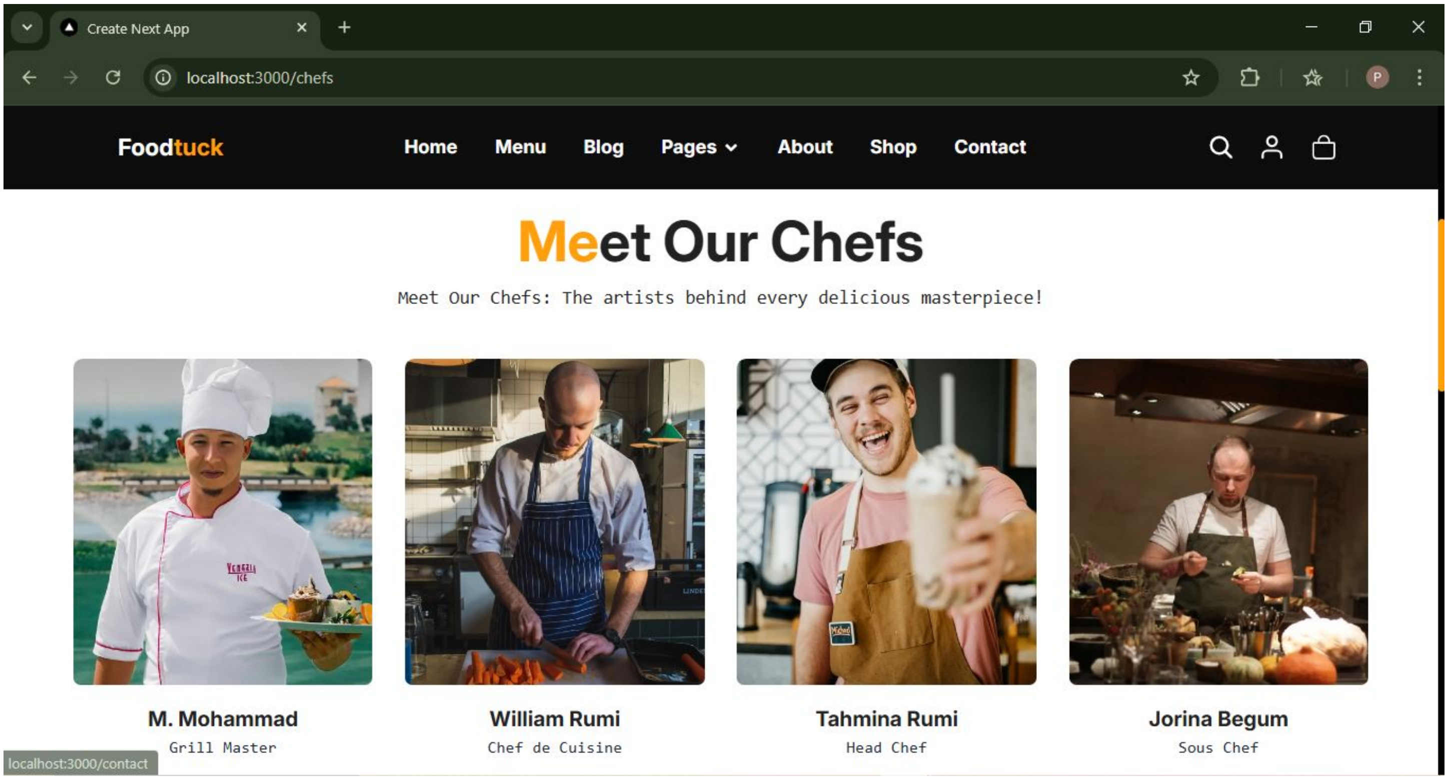


```
  FaStar,
  FaTwitter,
} from "react-icons/fa";

interface FoodProps {
  params: { slug: string };
}

async function getFood(slug: string): Promise<Food> {
  return client.fetch(
    groq`*[_type == 'food' && slug.current == $slug][0]{
      _id,
      name,
      _type,
      image,
      originalPrice,
      tags,
      category,
      price,
      description,
    }`,
    { slug }
  );
}

const page = async ({ params }: FoodProps) => {
  const { slug } = params;
  const food = await getFood(slug);

  if (!food) {
    return <div>Food not found</div>;
  }

```

Finally: Displayed Food Data on frontend



Displayed Chef Data on Frontend:



Meet Our Chefs

Meet Our Chefs: The artists behind every delicious masterpiece!