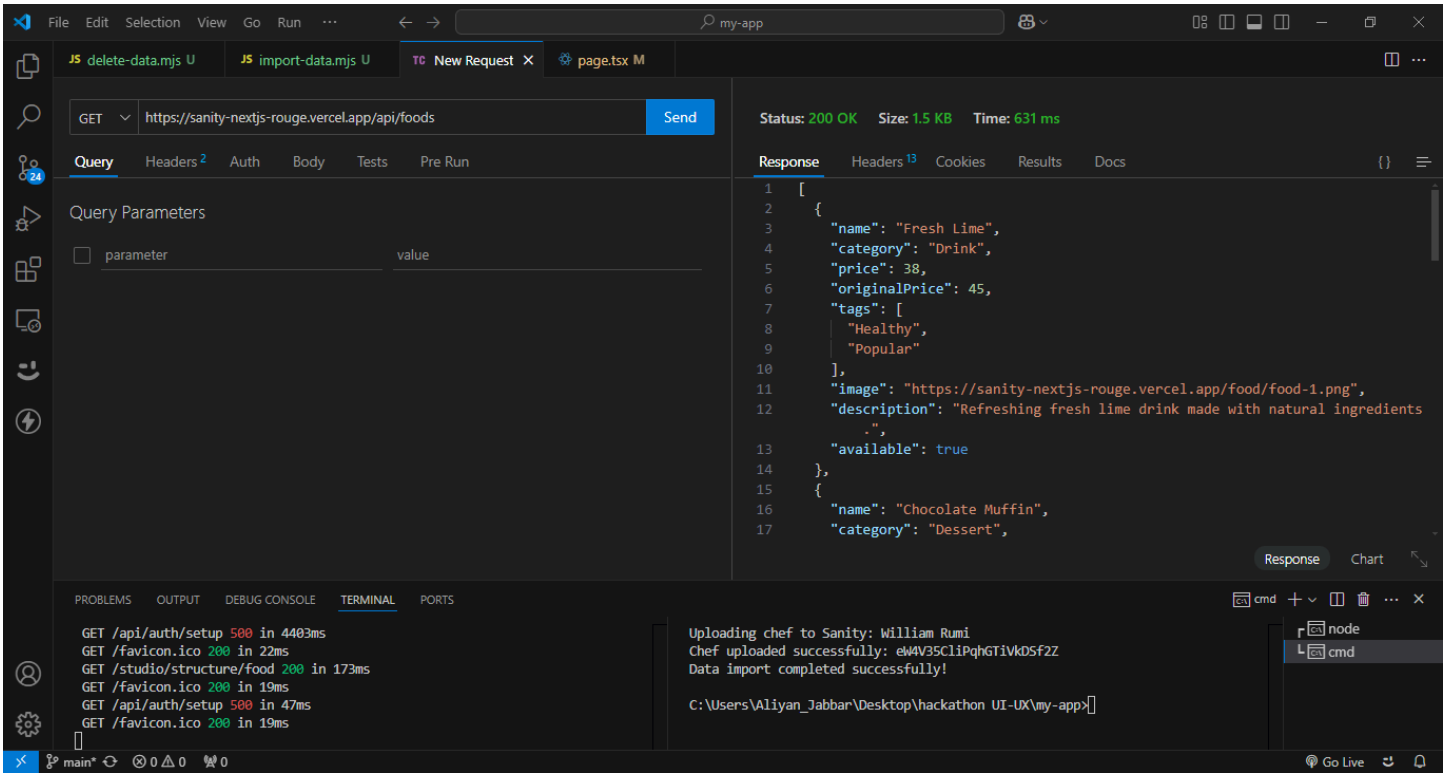
"Day 3 - API Integration Report - Food Chukh"

1. API Integration Process

- **Project Setup**: The project was initiated using create-next-app.
- **Sanity Integration**: Configured a Sanity client using the following setup
- **Environment Variables**:
- Required variables were added for both local development and deployment on Vercel:
  - NEXT_PUBLIC_SANITY_PROJECT_ID
  - NEXT_PUBLIC_SANITY_DATASET
  - SANITY_API_TOKEN
- **Checking API Data on Thunder Client:**



2. Schema Adjustments

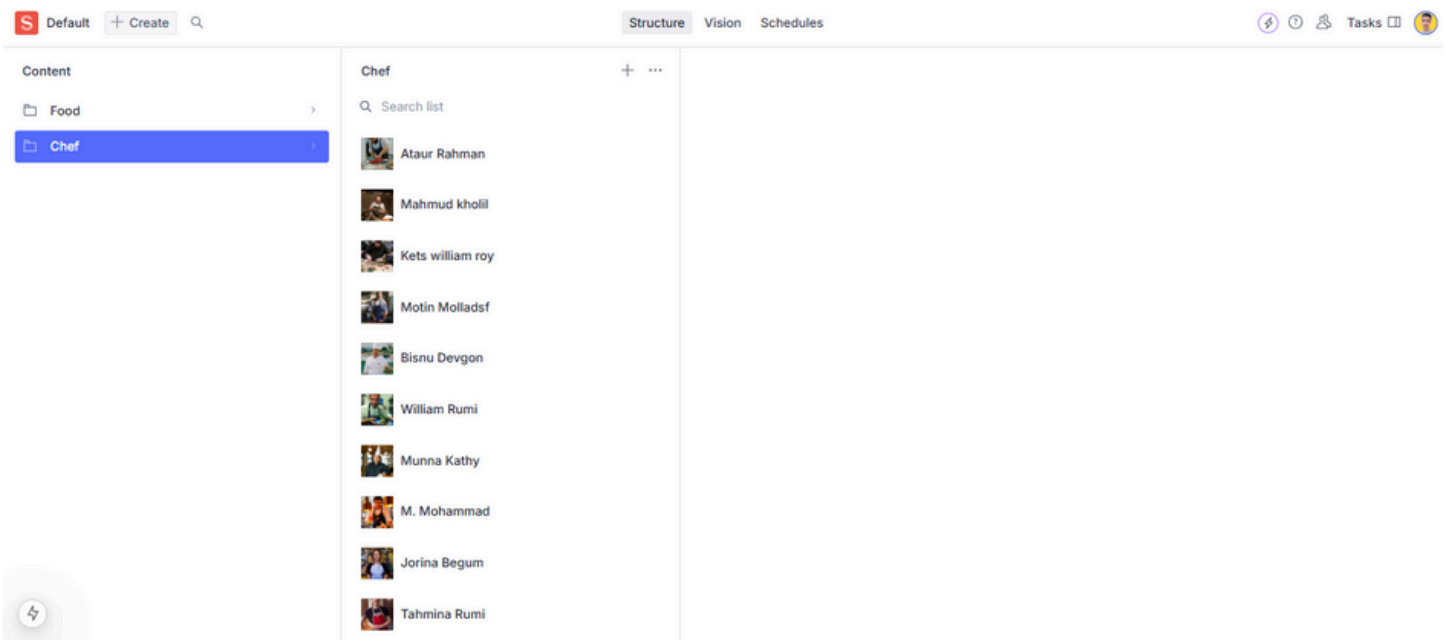
- **Schema Creation**:
  - Created custom schema types within Sanity Studio to define content structures.
- **TypeScript Integration**:
  - Added TypeScript interfaces for type safety when working with Sanity data.
- **Validation Rules**:
  - Defined validation rules for Sanity content fields to ensure data integrity.

- **Guidance From Given Schema:**
  - Notice some main fields which were missing in my schema like availiability & description.
- **Making New Schema:**
  - Made a new schema by combining my schema and the schema provided.

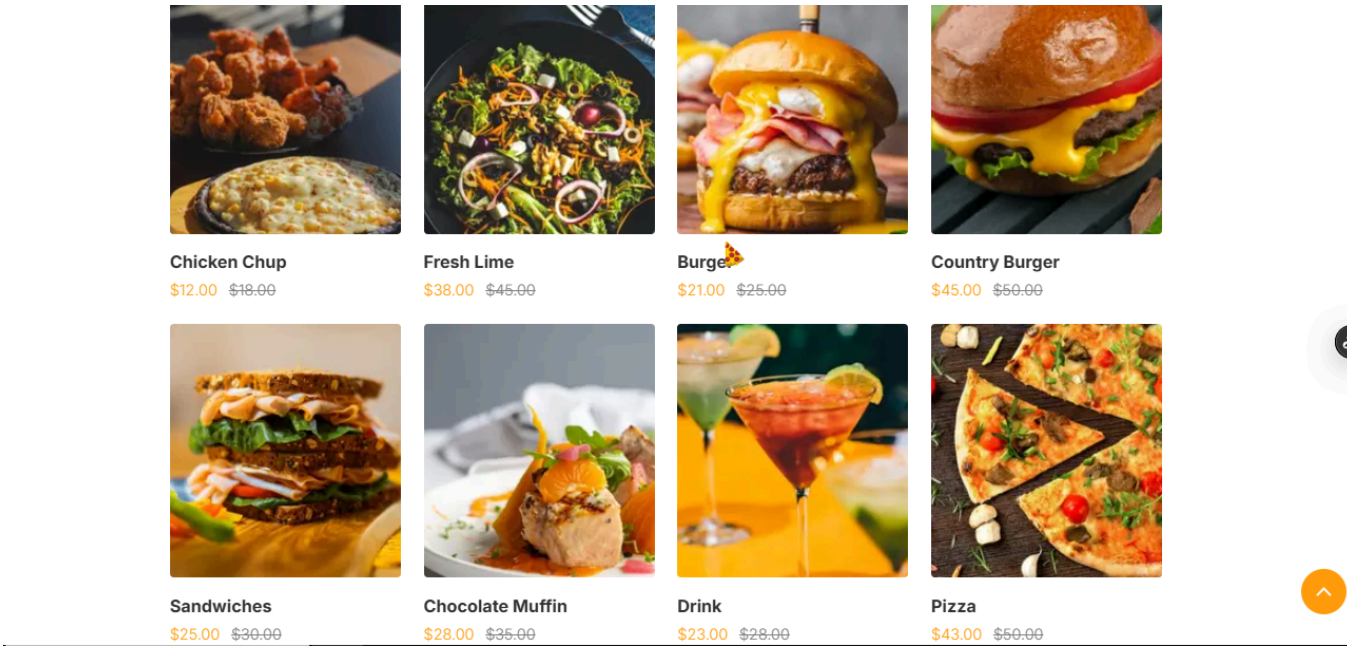
3. Migration and Optimization

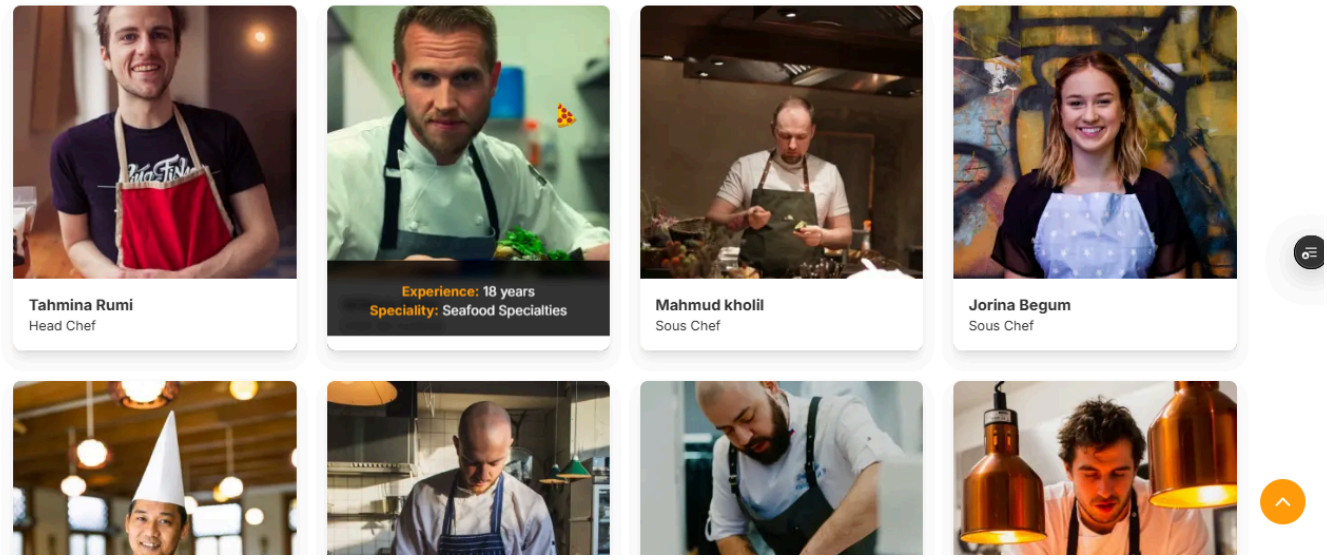
- **Next.js Version**: Used Next.js 15 latest version to enhance capabilities.
- **Font Optimization**: Implemented next/font to streamline font loading and improve performance.
- **Deployment**: Leveraged Vercel as the deployment platform for seamless CI/CD integration and hosting.
- **Pushing Documents to sanity:**

pushed the documents to sanity through scripts availiable on my github repo in root/src/scripts



- **Fetching from Sanity**:After Pushing all documents from api to sanity, the next task is to display documents on frontend, following are some sanity documents fetched and displayed on frontend:





Tools Used

- **Next.js**: Frontend framework for building the application.
- **Sanity CMS**: Content management system for managing and delivering data.
- **TypeScript**: Ensured type safety and reduced runtime errors.
- **ThunderClient**: For Testing API within VSCode through thunder client's extension
- **Vercel Platform**: Hosted the application with automated builds and deployments.