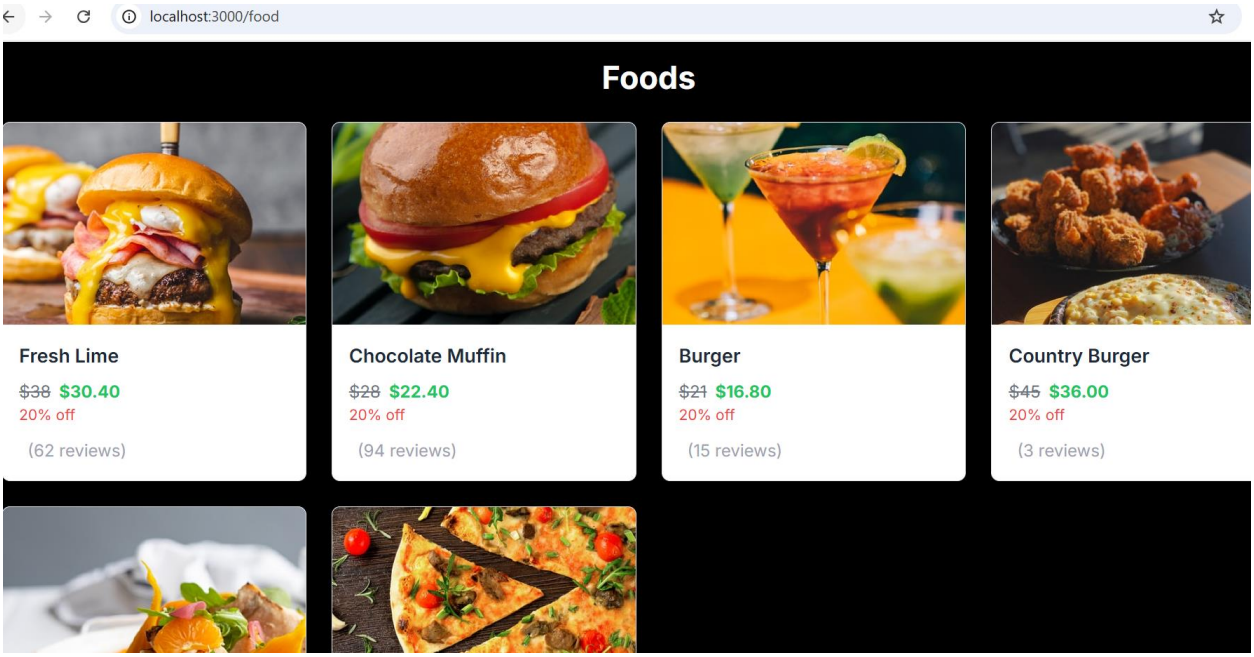
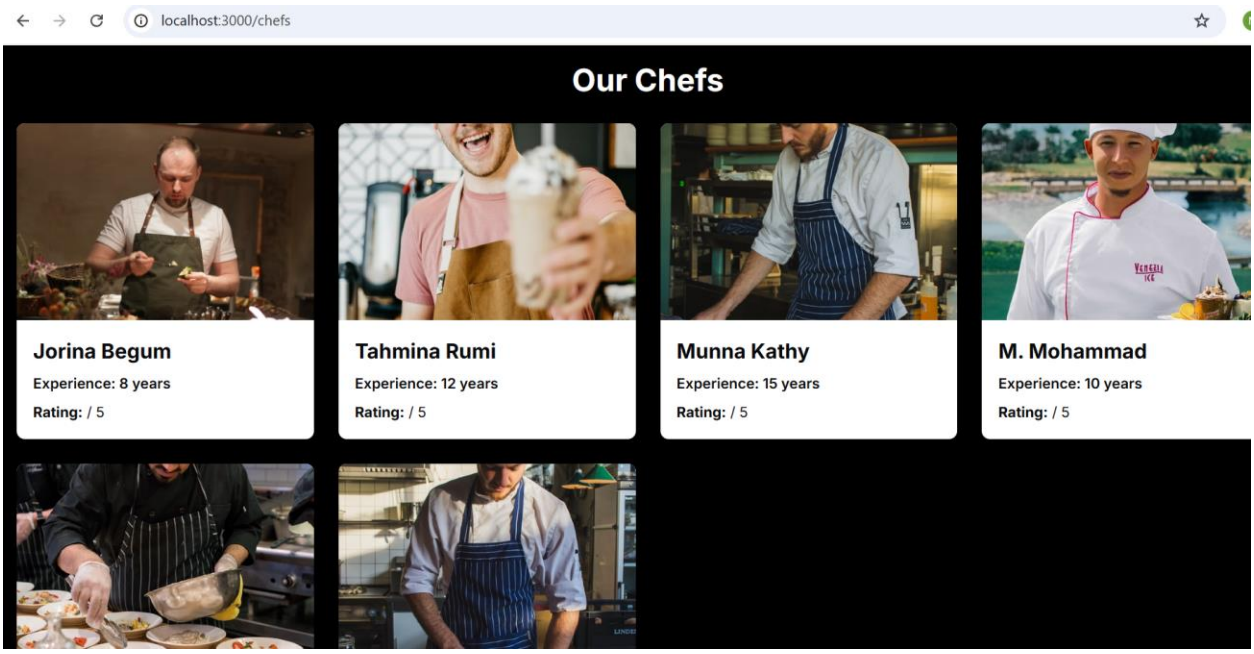
Day 3 - API Integration Report - Q-Commerce Food Tuck

API Integration Process

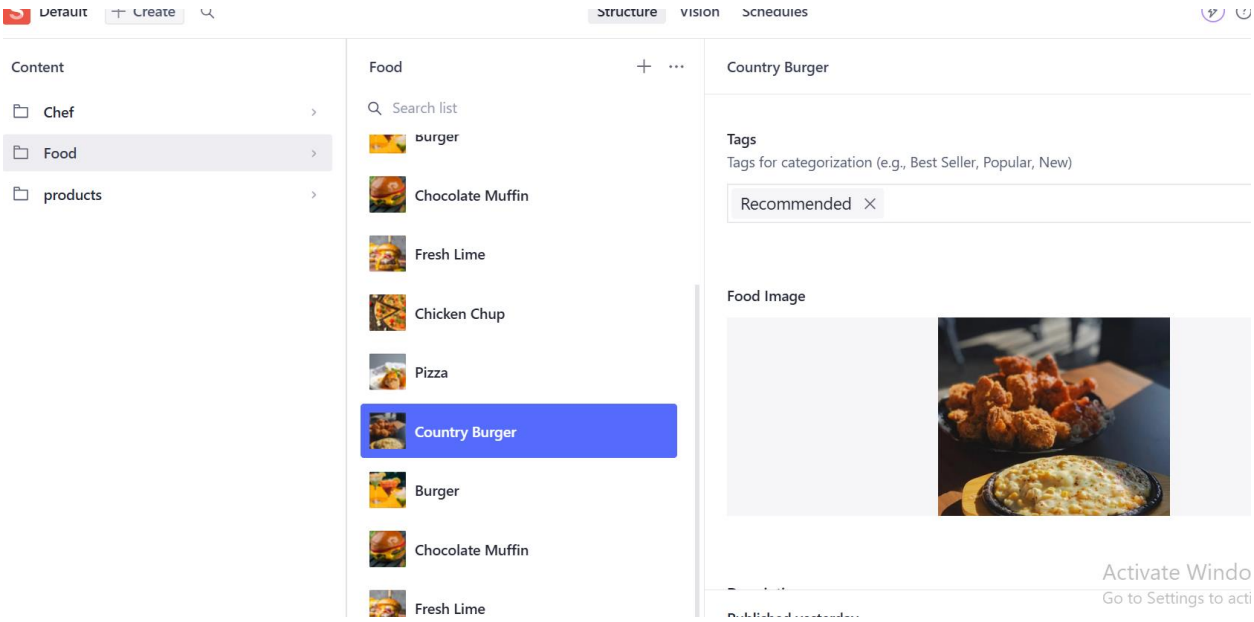
1. **Understanding API Requirements**
    a. Identified required endpoints for fetching and updating data.
    b. Reviewed API documentation for authentication and rate limits.
2. **Setting Up API Calls in Next.js**
    a. Utilized `fetch` and `axios` for making API requests.
    b. Implemented API handling in `api/` folder for modularity.
3. **Authentication & Security**
    a. Stored API keys securely in `.env.local`.
    b. Used token-based authentication where required.
4. **Error Handling & Response Management**
    a. Implemented error handling for timeout and invalid responses.
    b. Displayed appropriate messages for different HTTP status codes.





Adjustments Made to Schemas

1. **Sanity CMS Adjustments**
    a. Added missing fields for `product` and `order` schemas.
    b. Modified field types to align with API response data.
2. **Database Updates**
    a. Migrated existing schema to include `inventory` and `order_status` fields.
    b. Ensured backward compatibility for previous data.

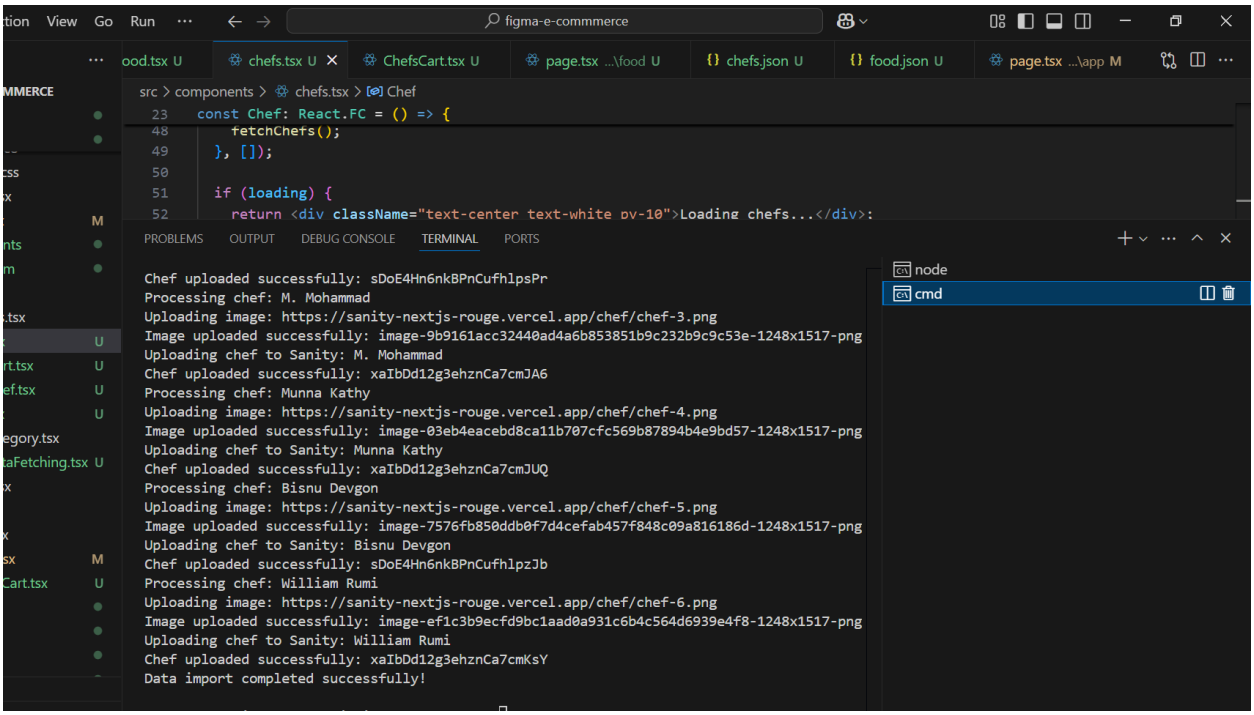


Migration Steps and Tools Used

1. **Database Backup**
    a. Created a backup of existing data before migration.
2. **Data Transformation**
    a. Used scripts to convert old data formats to new structures.
    b. Validated data types and constraints.
3. **Migration Execution**
    a. Used `Sanity CLI` for importing adjusted schema.
    b. Verified data consistency post-migration.

Screenshots

- API calls tested in Postman.
- Successfully displayed data on the frontend.
- Populated fields in Sanity CMS.



Best Practices Followed

1. **Use .env for sensitive data storage.**
2. **Clean coding practices:**
    a. Descriptive variable names.
    b. Modularized functions.
    c. Added comments for clarity.
3. **Validated data during migration:**
    a. Ensured field type consistency.
    b. Logged errors for further debugging.
4. **Detailed Documentation:**
    a. Included screenshots, scripts, and testing notes.
    b. Maintained a changelog for schema updates.
5. **Version Control:**
    a. Frequent commits with meaningful messages.
    b. Tagged major milestones.
6. **Comprehensive Testing:**
    a. Tested API endpoints with Postman.
    b. Handled edge cases for missing or invalid data.
7. **Peer Review:**
    a. Shared code for feedback.

b. Incorporated improvements.

Self-Validation Checklist

Task	Status
API Understanding	✓
Schema Validation	✓
Data Migration	✓
API Integration in Next.js	✓
Submission Preparation	✓

- API Integration and Data Migration for the **Q-Commerce Food Tuck** project. 🚀

BY: UROOJ MEMON