



```

|   "description": "string",
|   "price": "number",
|   "category": "string",
|   "imageUrl": "string",
|   "stock": "number",
|   "sizes": ["small", "large"]
| }
| - **Order Schema**:
| {
|   "orderId": "string",
|   "customerId": "string",
|   "customerName": "string",
|   "customerLocation": "string",
|   "items": [
|     {
|       "productId": "string",
|       "quantity": "number",
|       "size": "string"
|     }
|   ],
|   "totalAmount": "number",
|   "paymentStatus": "string",
|   "shippingStatus": "string",
|   "createdAt": "date"
| }

```

↓
v

```

+-----+
|   Build User Authentication   |
| - Implement registration and login using NextAuth.js |
+-----+

```

↓
v

```

+-----+
|   Develop Homepage           |
| - Displays featured food products and categories |
| - Includes search functionality to filter products |
|   based on categories, ingredients, and price |
+-----+

```

↓
v

```

+-----+
|   Create Product Page       |
| - Shows detailed information for each food item |
| - Includes product name, price, ingredients, |
|   dimensions, description, stock status, and |
|   high-quality images |
| - **Example:** |
| - Total Burgers: 5 |
| - After purchase of 1 burger: |
|   - Remaining Burgers: 4 |
|   - Update stock in database |
+-----+

```

|
v

+-----+
| Implement Cart Functionality |
| - Users can add, remove, and modify items |
| - The cart persists user selections until checkout |
| - ****Stock Management:**** |
| - When a burger is purchased, reduce stock by 1 |
| - Notify user of remaining stock |
+-----+

|
v

+-----+
| Develop Checkout Process |
| - Collect user information and payment details |
| - User authentication, shipping address, |
| order confirmation |
+-----+

|
v

+-----+
| Implement Order History |
| - Customers can view their past orders |
| - Includes products purchased, order total, |
| and status |
+-----+

|
v

+-----+	API Endpoints	
- GET /api/products: Retrieve a list of products		
- GET /api/products/[id]: Retrieve product details		
- POST /api/orders: Place an order		
- GET /api/orders/[id]: Retrieve order details		
- GET /api/users/[id]: Retrieve user profile info		
- PATCH /api/products/[id]: Update product stock		
- Decrease stock when a purchase is made		
+-----+		

|
v

+-----+ |

User Roles	
- *Admin:*	
- Manages the product catalog (add, update, delete products)	
- Views customer orders and manages order statuses	
- Manages user accounts (e.g., deactivate or update user details)	
- *Customer:*	
- Browses products, manages the cart, and places orders	
- Manages personal profile, views past orders, and tracks current order status	

+-----+ |

|
v

+-----+ |

Additional Pages	
- About Us: Information about the company	
- Contact Us: Form for customer inquiries	

- FAQ: Frequently asked questions
- Terms and Conditions: Legal information
- Privacy Policy: Data handling practices
- Blog: Articles related to food and recipes
- User Profile: Manage account settings
- Wishlist: Save favorite products
- Order Tracking: Real-time order status updates
- Admin Dashboard: Overview of site metrics
- Product Management: Admin interface for products
- Customer Management: Admin interface for users
- Reports: Sales and order reports for admin
- Promotions: Manage discounts and offers
- Feedback: Collect customer feedback
- Help Center: Support resources for users

|
v

- ### Deployment
- ****Platforms:****
 - Vercel for Next.js deployment
 - MongoDB Atlas for cloud-based database hosting
 - Stripe for secure payments
 - ****Steps:****
 1. Push the code to your GitHub repository
 2. Connect the GitHub repository to Vercel for automated deployment
 3. Set up environment variables on the Vercel dashboard
(e.g., MONGODB_URI, SANITY_PROJECT_ID)
 4. Deploy the app with one click

|
v

- ### Future Improvements
- User Reviews and Ratings: Allow customers to rate and review products
 - Enhanced Search Functionality: Implement full-text search with filtering by price range, ingredients, and customer ratings
 - Mobile App Version: Develop a mobile app with React Native for a better on-the-go shopping experience
 - Multi-language Support: Implement translations to cater to a global audience

|
v

- ### Conclusion
- This project provides an efficient and scalable platform for selling food. |
The use of Next.js, MongoDB, and Sanity CMS ensures fast performance, easy content management, and a rich user experience. |

|
v

- ### End Project