Hackathon – 3 (Day 2)
Marketplace Technical Foundation

System Overview

SpeedyEats is my Q-Commerce food marketplace designed to provide ultra-fast food delivery services. The system architecture is built to support rapid order processing, real-time tracking, and efficient delivery logistics. It combines a Next.js frontend, Sanity CMS backend, and various third-party APIs to create a seamless, scalable, and responsive platform.

Technical Documentation

Frontend (Next.js)

The frontend is built using Next.js, a React-based framework that provides server-side rendering, static site generation, and efficient routing.

Key Features:

- Server-Side Rendering (SSR) for improved SEO and initial load times
- Dynamic routing for seamless navigation between pages
- API routes for handling server-side logic
- Image optimization using the Next.js Image component
- Responsive design using Tailwind CSS

Components:

1. Home Page: Displays featured restaurants, popular dishes, and personalized recommendations

Hackathon – 3 (Day 2)
Marketplace Technical Foundation

2. Restaurant Listing: Shows a grid or list of restaurants with filtering and sorting options

3. Menu Page: Presents restaurant menus with item details and customization options

4. Cart & Checkout: Manages the order process, including item selection and payment

5. Order Tracking: Provides real-time updates on order status and delivery progress

6. User Profile: Allows users to manage their information, addresses, and order history

Backend (Sanity CMS)

Sanity CMS serves as the backend, providing a flexible and real-time content management system.

Key Features:

- Custom schemas for restaurants, menu items, orders, and user profiles
- Real-time content updates
- API for data retrieval and manipulation
- Content versioning and rollback capabilities

Data Models:

Hackathon – 3 (Day 2)
Marketplace Technical Foundation

1. Restaurant: Stores information about partner restaurants, including name, cuisine type, location, and operating hours

2. Menu Items: Contains details about dishes, including name, description, price, and customization options

3. Orders: Manages order information, status, and related customer details

4. User Profiles: Stores user account information, preferences, and order history

5. Delivery Zones: Defines geographical areas for delivery services and associated rules

WebSocket Server

Facilitates real-time communication between the frontend and backend for live updates.

Key Features:

- Instant order status updates

- Real-time delivery tracking

- Live inventory updates for menu items

Third-Party API Integrations

1. Auth0:

1. Handles user authentication and authorization

Hackathon – 3 (Day 2)
Marketplace Technical Foundation

2. Supports social media logins and secure token-based authentication

2. Stripe API:

1. Processes secure payments

2. Manages refunds and disputes

3. Supports multiple payment methods

3. Google Maps API:

1. Provides restaurant location services

2. Enables real-time delivery tracking

3. Assists in route optimization for delivery partners

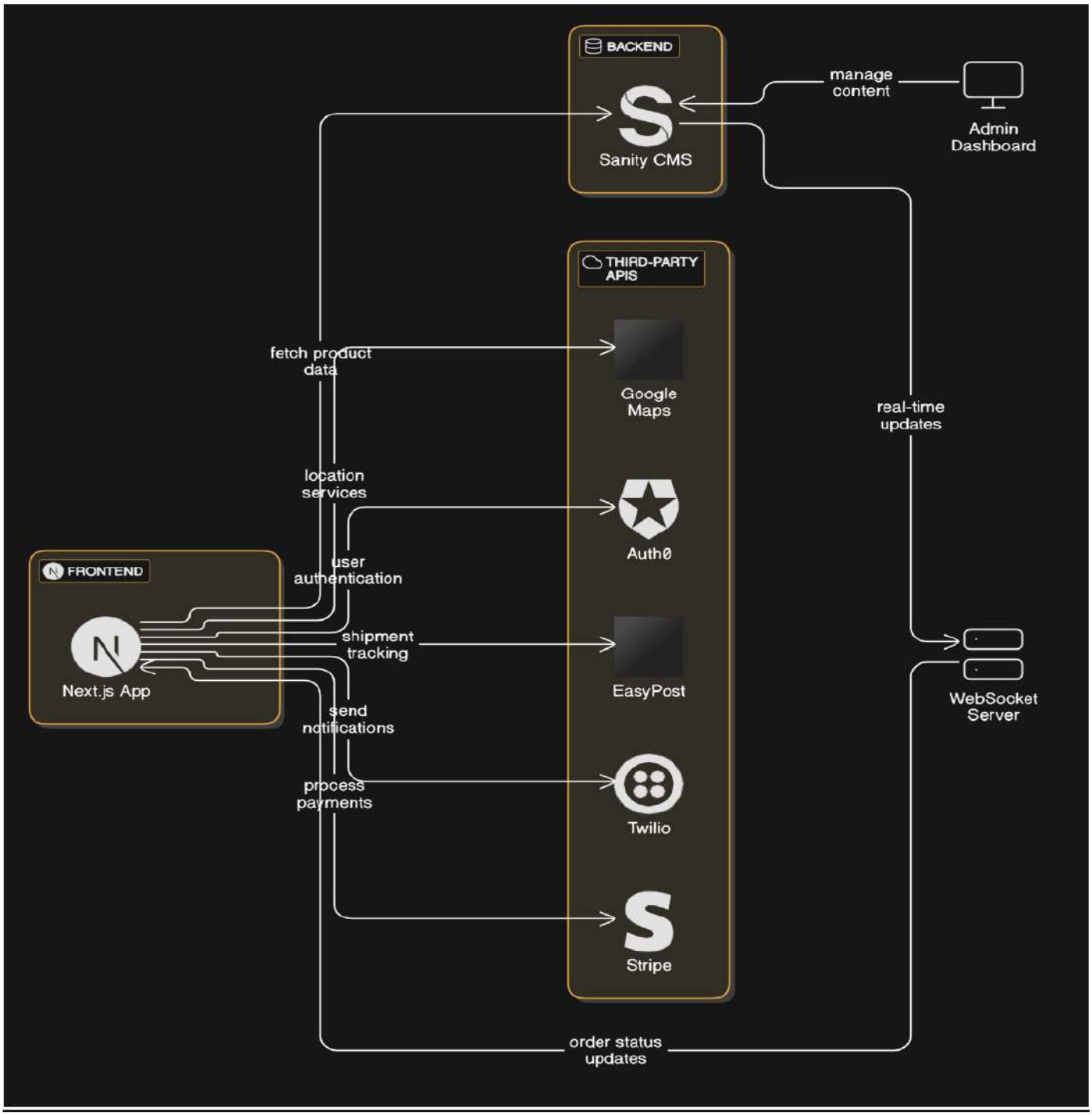
4. Twilio API:

1. Sends SMS notifications for order updates

2. Delivers email notifications for order confirmations and promotions

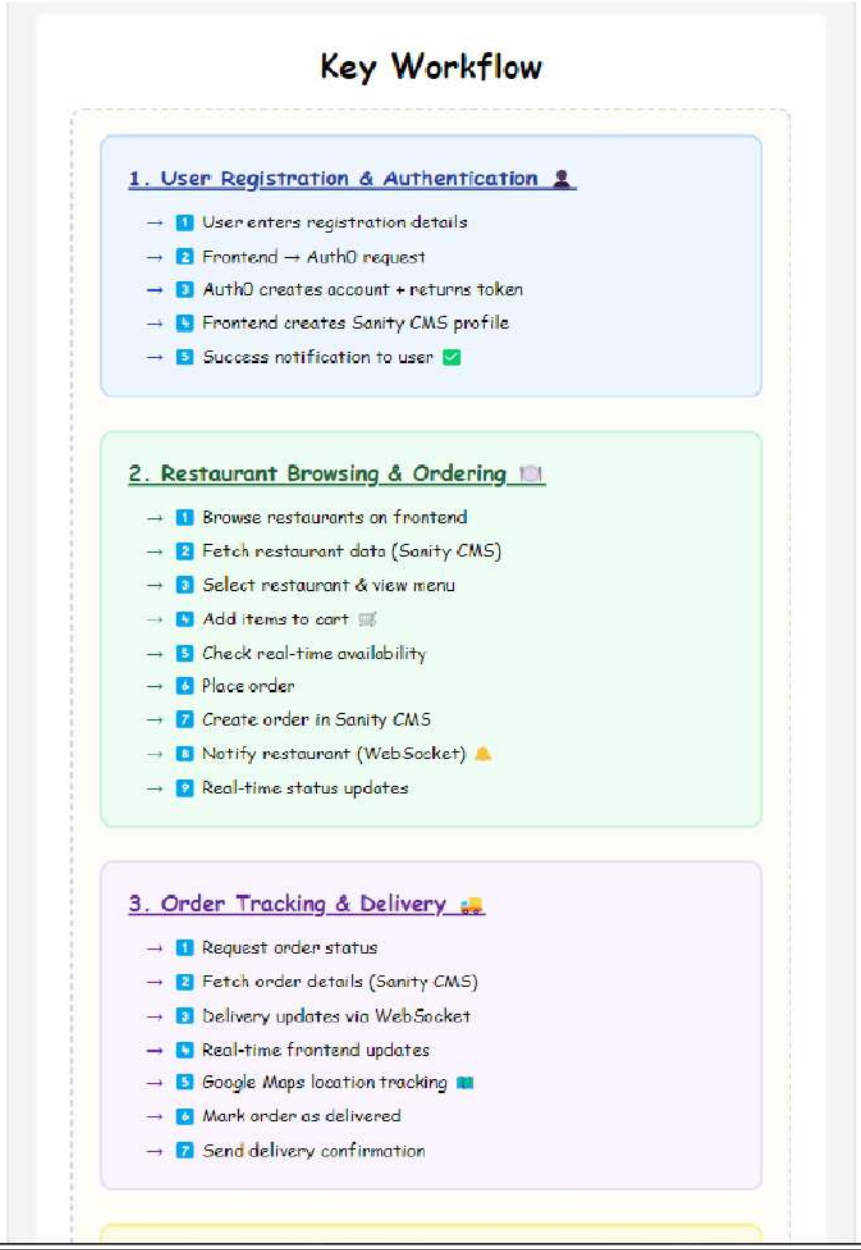
Hackathon – 3 (Day 2)
Marketplace Technical Foundation

Technical Requirements Flow Chart



Hackathon – 3 (Day 2)
Marketplace Technical Foundation

Key Workflow



Hackathon – 3 (Day 2)
Marketplace Technical Foundation

Key Workflows Overview

User Registration and Authentication:

1. User enters registration details on frontend
2. Frontend sends registration request to Auth0
3. Auth0 creates user account and returns token
4. Frontend creates user profile in Sanity CMS
5. User notified of successful registration

b) Restaurant Browsing and Ordering:

1. User browses restaurants on frontend
2. Frontend fetches restaurant data from Sanity CMS
3. User selects restaurant and views menu
4. User adds items to cart
5. Frontend checks item availability in real-time
6. User places order
7. Order created in Sanity CMS
8. Restaurant notified of new order via WebSocket
9. Order status updated in real-time on frontend

c) Order Tracking and Delivery:

1. User requests order status on frontend
2. Frontend fetches order details from Sanity CMS
3. Delivery partner updates status via WebSocket
4. Frontend receives real-time updates
5. Google Maps API provides real-time location tracking
6. Upon delivery, order status updated to "Delivered"
7. User receives delivery confirmation

Hackathon – 3 (Day 2)
Marketplace Technical Foundation

API ENDPOINTS

API Endpoints Details (Table Format)		
API Endpoint	**Method**	**Response**
Restaurants		
/api/restaurants	GET	List all restaurants
/api/restaurants/{id}	GET	Details of a specific restaurant
/api/restaurants/nearby?lat={latitude}&lon={longitude}	GET	Nearby restaurants
Menu Items		
/api/restaurants/{id}/menu	GET	Menu for a specific restaurant
/api/menu-items/{id}	GET	Details of a specific menu item
Orders		
/api/orders	POST	Create a new order
/api/orders/{id}	GET	Details of a specific order
/api/orders/{id}	PATCH	Update order status
/api/orders/user/{userId}	GET	Order history for a user
Users		
/api/users	POST	Register a new user
/api/users/{id}	GET	Get user profile
/api/users/{id}	PATCH	Update user profile
Delivery		
/api/delivery/{orderId}/track	GET	Real-time delivery status
/api/delivery/{orderId}/location	PATCH	Update delivery location
Reviews		
/api/reviews	POST	Create a new review
/api/restaurants/{id}/reviews	GET	Reviews for a specific restaurant

```
export default {
  name: 'restaurant',
  title: 'Restaurant',
  type: 'document',
  fields: [
    {
      name: 'name',
      title: 'Name',
      type: 'string',
      validation: Rule => Rule.required()
    },
    {
      name: 'slug',
      title: 'Slug',
      type: 'slug',
      options: { source: 'name', maxLength: 96 }
    },
    {
      name: 'image',
      title: 'Image',
      type: 'image',
      options: { hotspot: true }
    },
    {
      name: 'cuisineType',
      title: 'Cuisine Type',
      type: 'array',
      of: [{ type: 'string' }]
    },
    {
      name: 'address',
      title: 'Address',
      type: 'object',
      fields: [
        { name: 'street', type: 'string' },
        { name: 'city', type: 'string' },
        { name: 'postcode', type: 'string' },
        { name: 'location', type: 'geopoint' }
      ]
    },
    {
      name: 'rating',
      title: 'Rating',
      type: 'number',
      validation: Rule => Rule.min(0).max(5)
    },
    {
      name: 'averageDeliveryTime',
      title: 'Average Delivery Time (minutes)',
      type: 'number'
    },
    {
      name: 'isOpen',
      title: 'Is Open',
      type: 'boolean'
    }
  ]
}


```

```
export default {
  name: 'menuItem',
  title: 'Menu Item',
  type: 'document',
  fields: [
    {
      name: 'name',
      title: 'Name',
      type: 'string',
      validation: Rule => Rule.required()
    },
    {
      name: 'description',
      title: 'Description',
      type: 'text'
    },
    {
      name: 'image',
      title: 'Image',
      type: 'image',
      options: { hotspot: true }
    },
    {
      name: 'price',
      title: 'Price',
      type: 'number',
      validation: Rule => Rule.required().positive()
    },
    {
      name: 'category',
      title: 'Category',
      type: 'string',
      options: {
        list: [
          { title: 'Appetizer', value: 'appetizer' },
          { title: 'Main Course', value: 'main' },
          { title: 'Dessert', value: 'dessert' },
          { title: 'Beverage', value: 'beverage' }
        ]
      }
    },
    {
      name: 'restaurant',
      title: 'Restaurant',
      type: 'reference',
      to: [{ type: 'restaurant' }]
    },
    {
      name: 'isAvailable',
      title: 'Is Available',
      type: 'boolean',
      initialValue: true
    },
    {
      name: 'preparationTime',
      title: 'Preparation Time (minutes)',
      type: 'number'
    }
  ]
}
```

```
export default {
  name: 'order',
  title: 'Order',
  type: 'document',
  fields: [
    {
      name: 'orderNumber',
      title: 'Order Number',
      type: 'string',
      readOnly: true
    },
    {
      name: 'user',
      title: 'User',
      type: 'reference',
      to: [{ type: 'user' }]
    },
    {
      name: 'restaurant',
      title: 'Restaurant',
      type: 'reference',
      to: [{ type: 'restaurant' }]
    },
    {
      name: 'items',
      title: 'Order Items',
      type: 'array',
      of: [
        {
          type: 'object',
          fields: [
            { name: 'menuItem', type: 'reference', to: [{ type: 'menuItem' }] },
            { name: 'quantity', type: 'number' },
            { name: 'specialInstructions', type: 'text' }
          ]
        }
      ]
    },
    {
      name: 'totalAmount',
      title: 'Total Amount',
      type: 'number'
    },
    {
      name: 'status',
      title: 'Order Status',
      type: 'string',
      options: {
        list: [
          { title: 'Pending', value: 'pending' },
          { title: 'Preparing', value: 'preparing' },
          { title: 'Out for Delivery', value: 'out_for_delivery' },
          { title: 'Delivered', value: 'delivered' },
          { title: 'Cancelled', value: 'cancelled' }
        ]
      }
    },
    {
      name: 'deliveryAddress',
      title: 'Delivery Address',
      type: 'object',
      fields: [
        { name: 'street', type: 'string' },
        { name: 'city', type: 'string' },
        { name: 'postcode', type: 'string' },
        { name: 'location', type: 'geopoint' }
      ]
    },
    {
      name: 'createdAt',
      title: 'Created At',
      type: 'datetime',
      readOnly: true
    },
    {
      name: 'estimatedDeliveryTime',
      title: 'Estimated Delivery Time',
      type: 'datetime'
    }
  ]
}
```

```
export default {
  name: 'user',
  title: 'User',
  type: 'document',
  fields: [
    {
      name: 'name',
      title: 'Name',
      type: 'string',
      validation: Rule => Rule.required()
    },
    {
      name: 'email',
      title: 'Email',
      type: 'string',
      validation: Rule => Rule.required().email()
    },
    {
      name: 'phoneNumber',
      title: 'Phone Number',
      type: 'string'
    },
    {
      name: 'addresses',
      title: 'Addresses',
      type: 'array',
      of: [
        {
          type: 'object',
          fields: [
            { name: 'street', type: 'string' },
            { name: 'city', type: 'string' },
            { name: 'postcode', type: 'string' },
            { name: 'location', type: 'geopoint' },
            { name: 'isDefault', type: 'boolean' }
          ]
        }
      ]
    },
    {
      name: 'createdAt',
      title: 'Created At',
      type: 'datetime',
      readOnly: true
    }
  ]
}


```

```
export default {
  name: 'review',
  title: 'Review',
  type: 'document',
  fields: [
    {
      name: 'user',
      title: 'User',
      type: 'reference',
      to: [{ type: 'user' }]
    },
    {
      name: 'restaurant',
      title: 'Restaurant',
      type: 'reference',
      to: [{ type: 'restaurant' }]
    },
    {
      name: 'rating',
      title: 'Rating',
      type: 'number',
      validation: Rule => Rule.required().min(1).max(5)
    },
    {
      name: 'comment',
      title: 'Comment',
      type: 'text'
    },
    {
      name: 'createdAt',
      title: 'Created At',
      type: 'datetime',
      readOnly: true
    }
  ]
}


```