Prepared By: Mustafa Qazi

  Student ID: 00010241

API INTEGRATION REPORT - urbanstitch

First published: PKT 05:30 PM, January 18, 2025

Document Revision Information

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Amendment** | **Author** |
| 1.0 | 11:30 PM, January 18, 2025 | Initial release | Mustafa Qazi |
|  |  |  |  |

**Table of Contents**

**System Architecture Overview2**

**Detailed API Requirements Document3**

**Sanity Schemas Design 10**

**Screenshots of API Integration and Data Migration Process………………………………………………….13**

### ****1. System Architecture Overview****

#### **Frontend:**

* Framework: **Next.js**
  + Server-Side Rendering (SSR) and Incremental Static Regeneration (ISR) for SEO and performance.
  + Dynamic routing for product categories and detail pages.
* Styling: **Tailwind CSS** for rapid UI development.
* State Management: **Context API** or for cart and user session handling.

#### **Backend:**

* API Routes: Built into Next.js to handle server-side logic.
  + /api/products: Fetch product data from **Sanity CMS.**
  + /api/cart: Manage cart operations (add, remove, update).
  + /api/checkout: Process payments using Stripe / or any other payment gateway
  + /api/shipping: Integrate with ShipEngine for shipping operations.
  + /api/orders: Save and retrieve order details.

**2. Detailed API Requirements Document**

### ****Overview****

* **Platform Name**: UrbanStitch Marketplace API
* **Frontend**: Next.js
* **Backend & Content Management**: Sanity CMS
* **Shipping Integration**: ShipEngine API
* **Payment Integration**: Stripe API
* **Authentication**: JSON Web Tokens (JWT) or API Key (based on user roles)
* **Base URL**:
  + Development & Production: vercel link here ………….
* **Content Type**: JSON (application/json)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Feature** | **EndPoint** | **Method** | **Description** | **Parameters** | **Response Example** |
| Authentication | /auth/login | POST | User Login | email, password, userid, name, role | {  "email": "user@example.com",  "password": "password123" } Response (200):  {  "token": "ey12345exampletoken",  "expires\_in": 3600,  "user": {  "id": "u1",  "name": "Ali Asghar",  "email": "aliasghar@gmail.com",  "role": "customer"  } } Response (401):  {  "error": "Invalid email or password" } |
| Product Management | /products | GET | Get All Products | id, name, desc, price, category, size, colors, image, instock | Response (200):  [  {  "id": "p1",  "name": "Casual T-Shirt",  "description": "100% cotton unisex t-shirt",  "price": 19.99,  "category": "T-Shirts",  "sizes": ["S", "M", "L", "XL"],  "colors": ["White", "Black"],  "image": "https://cdn.example.com/tshirt1.jpg",  "in\_stock": true  },  {  "id": "p2",  "name": "Denim Jacket",  "description": "Premium quality denim jacket",  "price": 59.99,  "category": "Jackets",  "sizes": ["M", "L"],  "colors": ["Blue"],  "image": "https://cdn.example.com/jacket1.jpg",  "in\_stock": false  } ] |
| Product Management | /products/{id} | GET | Get Products by ID | id, name, desc, price, category, size, colors, image, instock | • Response (200):  • { • "id": "p1", • "name": "Casual T-Shirt", • "description": "100% cotton unisex t-shirt", • "price": 19.99, • "category": "T-Shirts", • "sizes": ["S", "M", "L", "XL"], • "colors": ["White", "Black"], • "image": "https://cdn.example.com/tshirt1.jpg", • "in\_stock": true • } |
| Product Management | /products | POST | Create Product (Admin Only) |  | Method: POST Request Body:  {  "name": "Leather Boots",  "description": "Genuine leather boots for men and women",  "price": 129.99,  "category": "Footwear",  "sizes": ["8", "9", "10", "11"],  "colors": ["Brown", "Black"],  "image": "https://cdn.example.com/boots1.jpg",  "in\_stock": true } Response (201):  {  "message": "Product created successfully",  "product": {  "id": "p3",  "name": "Leather Boots",  "price": 129.99  } } |
| Category Management | /categories | GET | Get All Categories |  | Response (200):  [  {  "id": "c1",  "name": "T-Shirts"  },  {  "id": "c2",  "name": "Jackets"  },  {  "id": "c3",  "name": "Footwear"  } ] |
|  | /categories | POST | Create Category (Admin Only) |  | Request Body:  {  "name": "Accessories" } Response (201):  {  "message": "Category created successfully",  "category": {  "id": "c4",  "name": "Accessories"  } } |
| Cart Management | /cart/add | POST | Add to Cart |  | **Request Body**:  {  "userId": "u1",  "productId": "p1",  "quantity": 2  }  **Response (200)**:  {  "message": "Product added to cart successfully",  "cart": {  "userId": "u1",  "items": [  {  "productId": "p1",  "quantity": 2  }  ]  }  } |
| Order Management | /orders | POST | Create Order |  | **Request Body**:  {  "userId": "u1",  "products": [  { "productId": "p1", "quantity": 2 },  { "productId": "p2", "quantity": 1 }  ],  "shippingAddress": {  "line1": "123 Fashion St",  "city": "Los Angeles",  "state": "CA",  "zip": "90001",  "country": "US"  }  }  **Response (201)**:  {  "message": "Order created successfully",  "order": {  "id": "o1",  "totalAmount": 99.97  }  } |
| Payment Processing (Stripe API) | /payments/create-intent | POST | Create Payment Intent |  | **Request Body**:  {  "amount": 99.97,  "currency": "USD"  }  **Response (200)**:  {  "clientSecret": "pi\_123456789\_secret\_abcdefgh"  } |
| Shipping Management (Ship Engine API) | /shipping/rates | POST | Get Shipping Rates |  | **Request Body**:  {  "from": {  "postalCode": "90001",  "countryCode": "US"  },  "to": {  "postalCode": "10001",  "countryCode": "US"  },  "weight": {  "value": 2.5,  "unit": "pounds"  }  }  **Response (200)**:  [  {  "carrier": "UPS",  "service": "Ground",  "rate": 12.99,  "estimatedDeliveryDays": 3  },  {  "carrier": "FedEx",  "service": "Express",  "rate": 24.99,  "estimatedDeliveryDays": 1  }  ] |
|  | /shipping/create-label | POST | Create Shipping Label |  | {  "shipment": {  "fromAddress": {  "line1": "Warehouse 1",  "city": "Los Angeles",  "state": "CA",  "zip": "90001",  "country": "US"  },  "toAddress": {  "line1": "123 Fashion St",  "city": "New York",  "state": "NY",  "zip": "10001",  "country": "US"  },  "weight": {  "value": 2.5,  "unit": "pounds"  },  "carrier": "UPS",  "serviceCode": "ground"  }  }  **Response (200)**:  {  "label": {  "labelUrl": "https://cdn.example.com/shipping\_label.pdf",  "trackingNumber": "1Z12345E1234567890"  }  } |
| Error Handling |  |  | General Error Format |  | **Response (4xx/5xx)**:  {  "error": "Error description",  "details": "Optional details about the error"  } |

**3. Sanity Schemas Design**

### ****1. Product Schema****

Defines the structure for products in the marketplace, including key attributes like name, price, category, sizes, and colors.

**1. Product Schema**

| **Field Name** | **Type** | **Title** | **Validation** | **Additional Options** |
| --- | --- | --- | --- | --- |
| name | string | Product Name | Required, max 100 characters | N/A |
| slug | slug | Slug | Required, max length 96 | Source: name |
| description | text | Description | Required, max 500 characters | N/A |
| price | number | Price | Required, minimum 0 | N/A |
| category | reference | Category | Required | References category type |
| sizes | array | Available Sizes | N/A | List: XS, S, M, L, XL, XXL |
| colors | array | Available Colors | N/A | Of type string |
| inStock | boolean | In Stock | N/A | Default value: true |
| images | array | Product Images | N/A | Of type image, hotspot enabled |

### ****2. Category Schema****

Organizes products into categories, such as "T-Shirts," "Jackets," or "Footwear."

| **Field Name** | **Type** | **Title** | **Validation** | **Additional Options** |
| --- | --- | --- | --- | --- |
| name | string | Category Name | Required, max 50 characters | N/A |
| slug | slug | Slug | Required, max length 96 | Source: name |
| description | text | Description | Max 200 characters | N/A |

### ****3. Customer Schema****

Manages customer profiles, including contact information and addresses.

| **Field Name** | **Type** | **Title** | **Validation** | **Additional Options** |
| --- | --- | --- | --- | --- |
| name | string | Full Name | Required | N/A |
| email | string | Email | Required, must be a valid email | N/A |
| phone | string | Phone Number | Required, regex validation for phone numbers | Format: ^\+?\d{10,15}$ |
| addresses | array | Addresses | N/A | Array of objects with fields: line1, city, state, zip, country |

### ****4. Order Schema****

Tracks customer orders, including product details, shipping, and payment status.

| **Field Name** | **Type** | **Title** | **Validation** | **Additional Options** |
| --- | --- | --- | --- | --- |
| orderId | string | Order ID | Required | N/A |
| customer | reference | Customer | Required | References customer type |
| products | array | Products | N/A | Array of objects with fields: product, quantity, price |
| totalAmount | number | Total Amount | Required, minimum 0 | N/A |
| shippingAddress | object | Shipping Address | N/A | Fields: line1, city, state, zip, country |
| status | string | Order Status | N/A | Options: Pending, Processing, Shipped, Delivered, Canceled |
| paymentStatus | string | Payment Status | N/A | Options: Pending, Paid, Failed, Refunded |

### ****5. Review Schema****

Stores customer reviews for products.

| **Field Name** | **Type** | **Title** | **Validation** | **Additional Options** |
| --- | --- | --- | --- | --- |
| product | reference | Product | Required | References product type |
| customer | reference | Customer | Required | References customer type |
| rating | number | Rating | Required, minimum 1, maximum 5 | N/A |
| reviewText | text | Review Text | Max 300 characters | N/A |

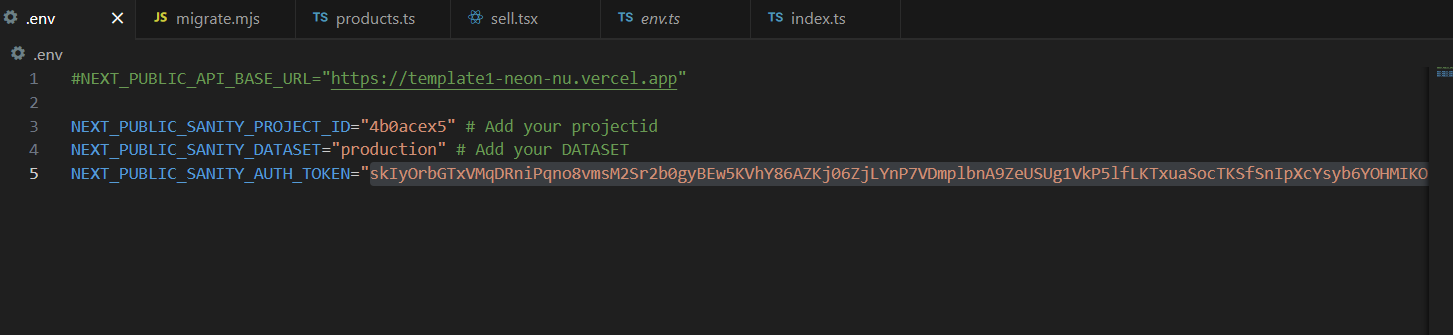
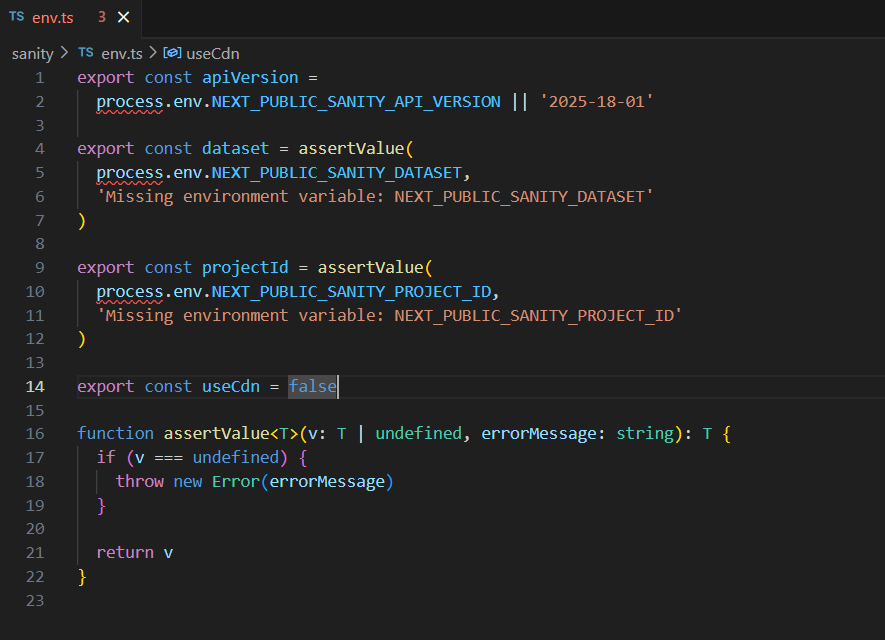
**6. Banner Schema**

For Homepage Promotions

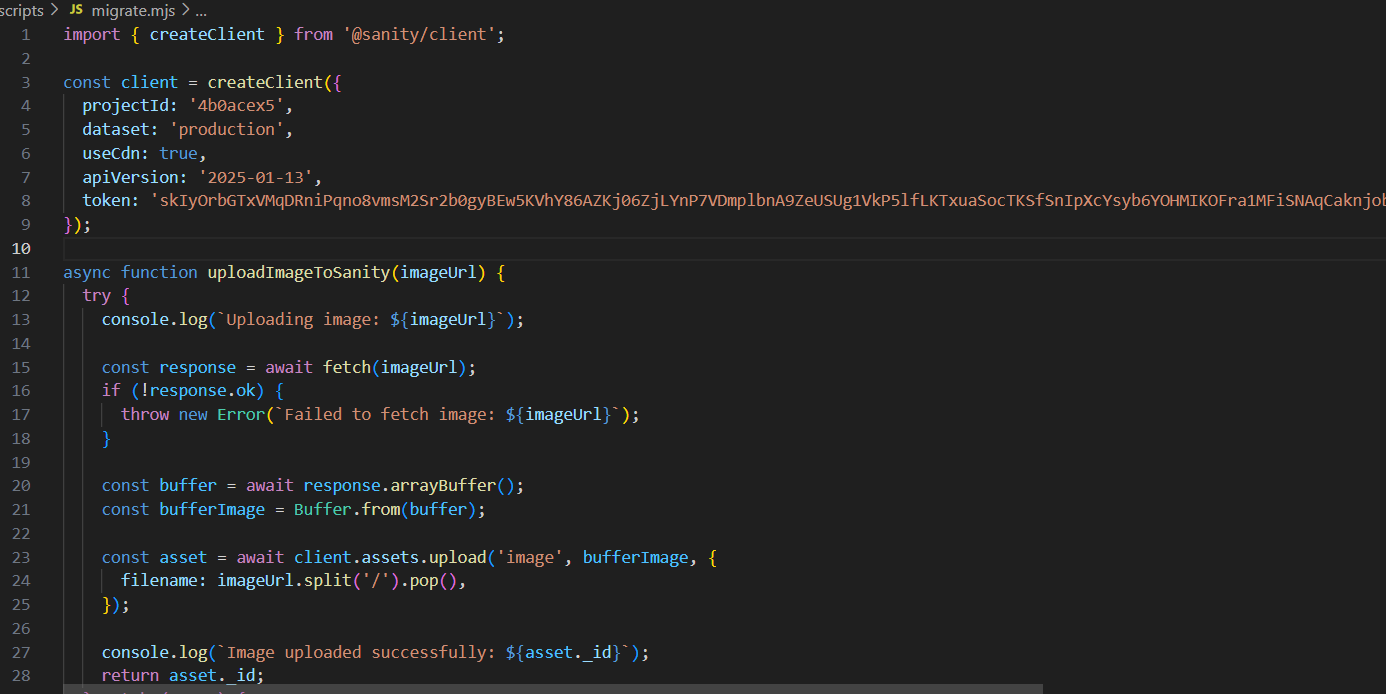
| **Field Name** | **Type** | **Title** | **Validation** | **Additional Options** |
| --- | --- | --- | --- | --- |
| title | string | Title | Required | N/A |
| image | image | Banner Image | N/A | Hotspot enabled |
| link | url | Redirect Link | N/A | N/A |

**Screenshots of API Integration and Data Migration Process**

Use .env file



Migration code for sanity



./ products fetch from sanity

