HACKHATON 3

DAY # 2

Marketplace Technical Foundation [AURA BOX]

Documentation Author: Umar Ali
Slot: Tuesday (2 to 5)
Task given by: Sir Ameen Alam
Class Teacher: Sir Ali Aftab Sheikh & Sir Muhammad Fahad Khan

1. Technical Requirments:

Front-end:

NextJS, HTML & TailwindCSS

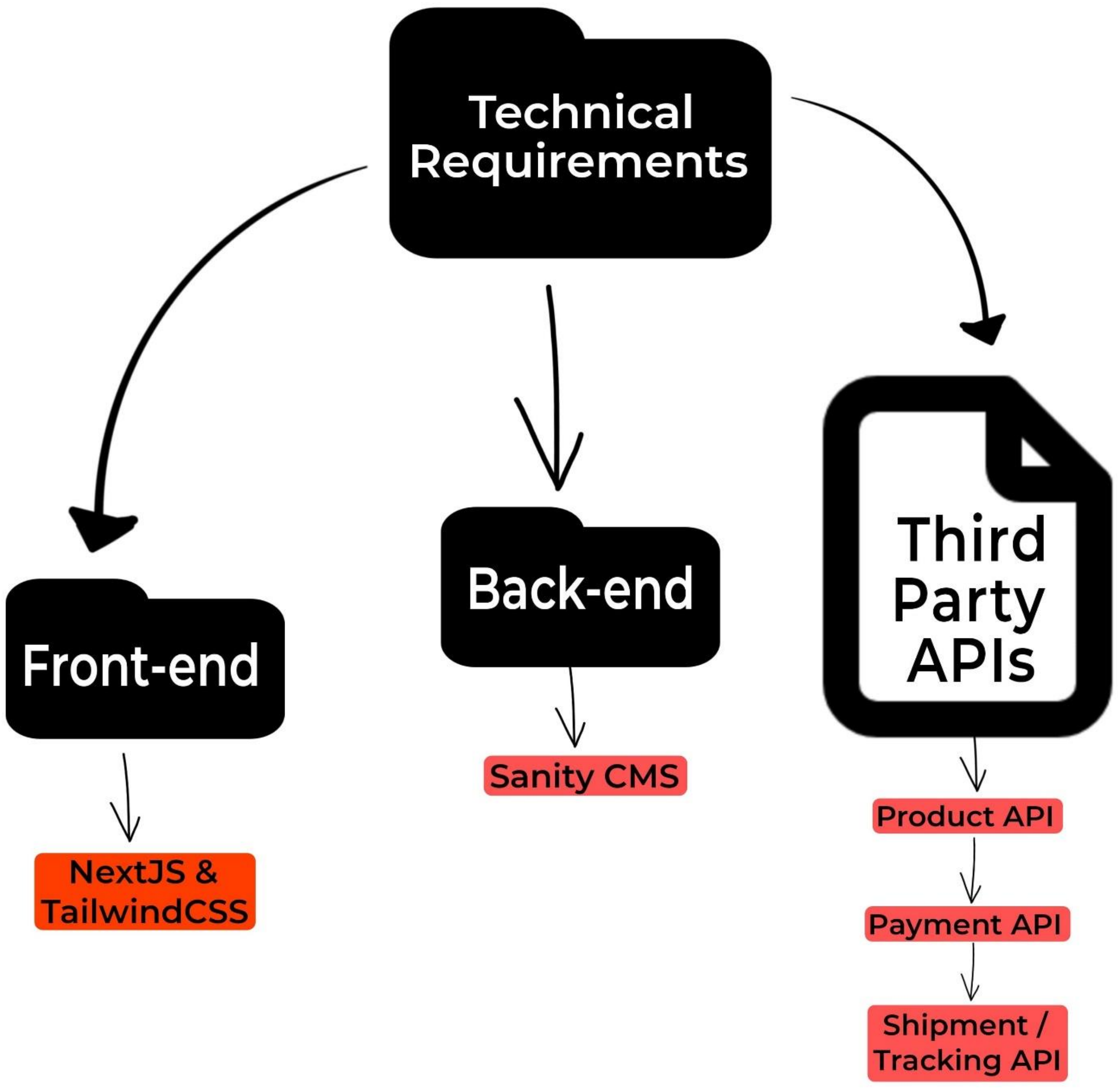
Content Management System:

Sanity

API:

Will use 3rd party APIs for Payment, Shipping, Tracking etc

DIAGRAM



ORDER PROCESSING FLOW

1. Product Selection

- User finds and clicks on a product.
- Front-end sends the product ID to the API.
- API requests product details from the database.
- Database returns product info (images, description, price, etc.).
- API sends details to the front-end for display.

2. Add to Cart

- User selects quantity and clicks "Add to Cart."
- Front-end sends product ID and quantity to the cart API.
- API updates the cart in the database and returns updated cart details.
- Front-end updates the cart and moves the user to checkout.

3. Checkout Process

- User reviews cart, enters shipping details, and selects payment method.
- Front-end sends cart, shipping, and payment info to the third-party API.

- Back-end saves order details and communicates with the payment gateway.

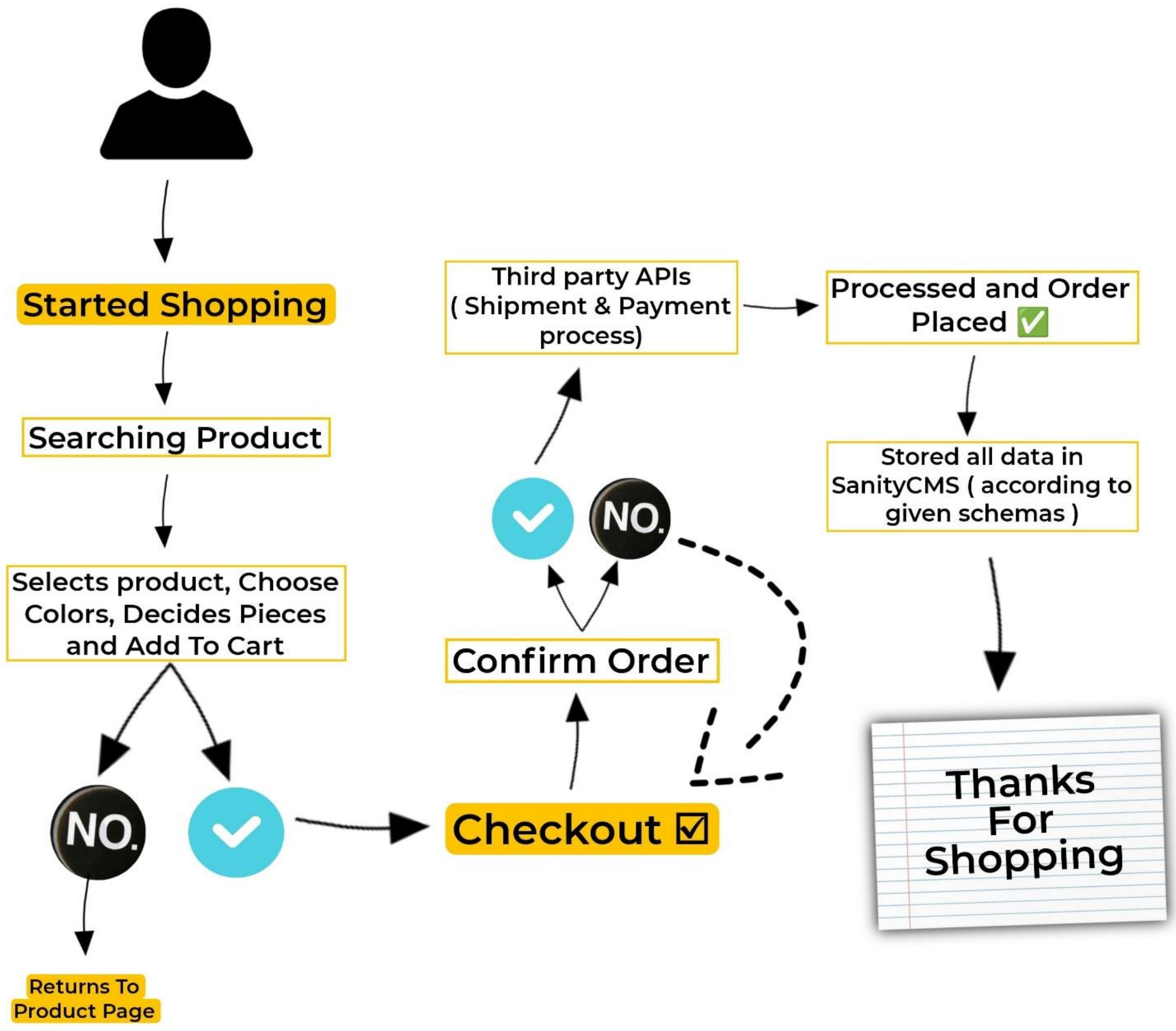
4. Payment & Order Confirmation

- Payment gateway processes the payment and sends success/failure response.
- Back-end updates order status based on payment response.
- API sends confirmation details (order ID, delivery estimate, tracking info) to the front-end.
- Front-end displays confirmation to the user.

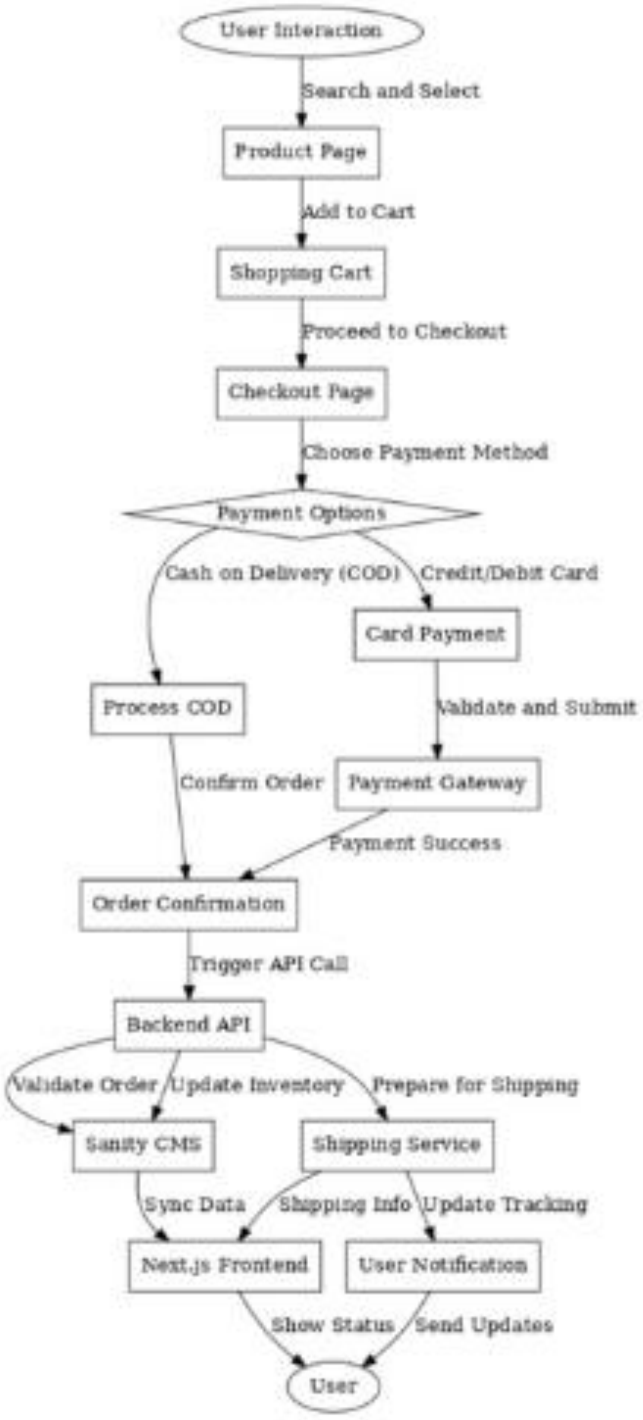
5. Order Tracking & Completion

- API helps track the order and updates the user on the status.
- Once completed, order data is saved in the database.

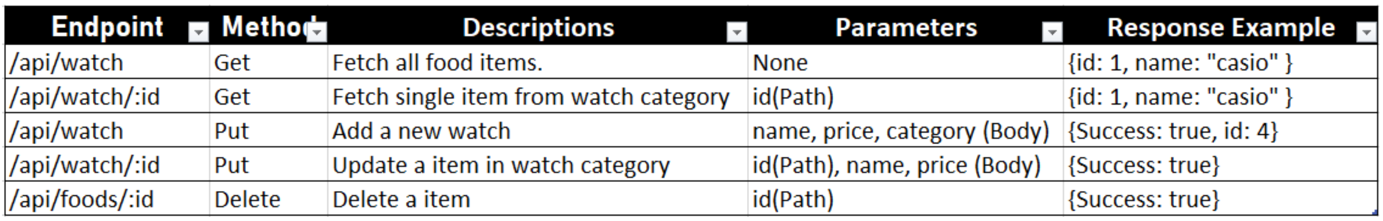
WORK FLOW DIAGRAM


Started Shopping
Searching Product
Selects product, Choose Colors, Decides Pieces and Add To Cart
NO.
Returns To Product Page
Checkout ☑
Confirm Order
NO.
Third party APIs (Shipment & Payment process)
Processed and Order Placed ✅
Stored all data in SanityCMS (according to given schemas)
Thanks For Shopping

2. **System Architecture**

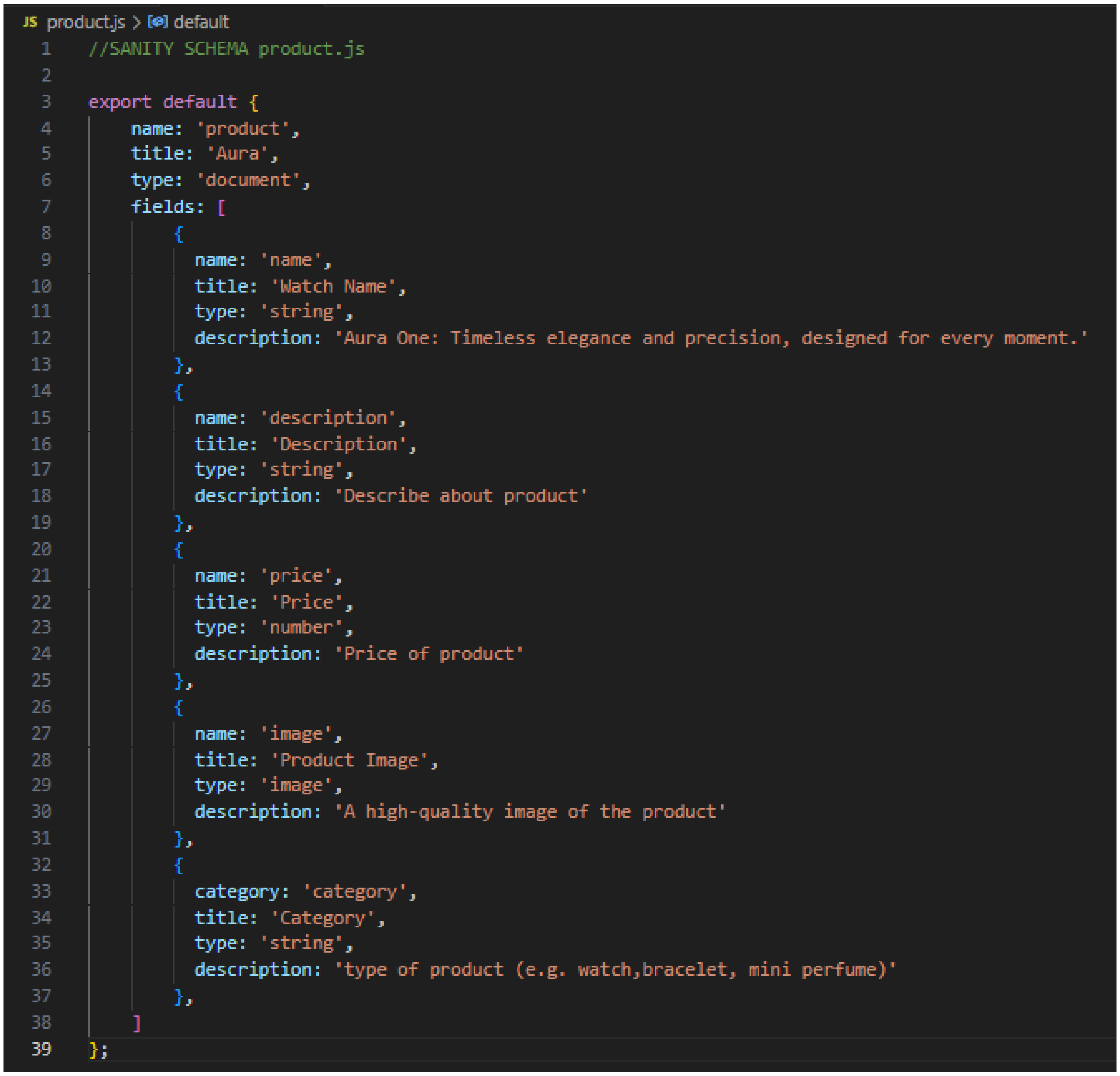


3. **API Endpoints:**



Endpoint	Method	Descriptions	Parameters	Response Example
/api/watch	Get	Fetch all food items.	None	{id: 1, name: "casio" }
/api/watch/:id	Get	Fetch single item from watch category	id(Path)	{id: 1, name: "casio" }
/api/watch	Put	Add a new watch	name, price, category (Body)	{Success: true, id: 4}
/api/watch/:id	Put	Update a item in watch category	id(Path), name, price (Body)	{Success: true}
/api/foods/:id	Delete	Delete a item	id(Path)	{Success: true}

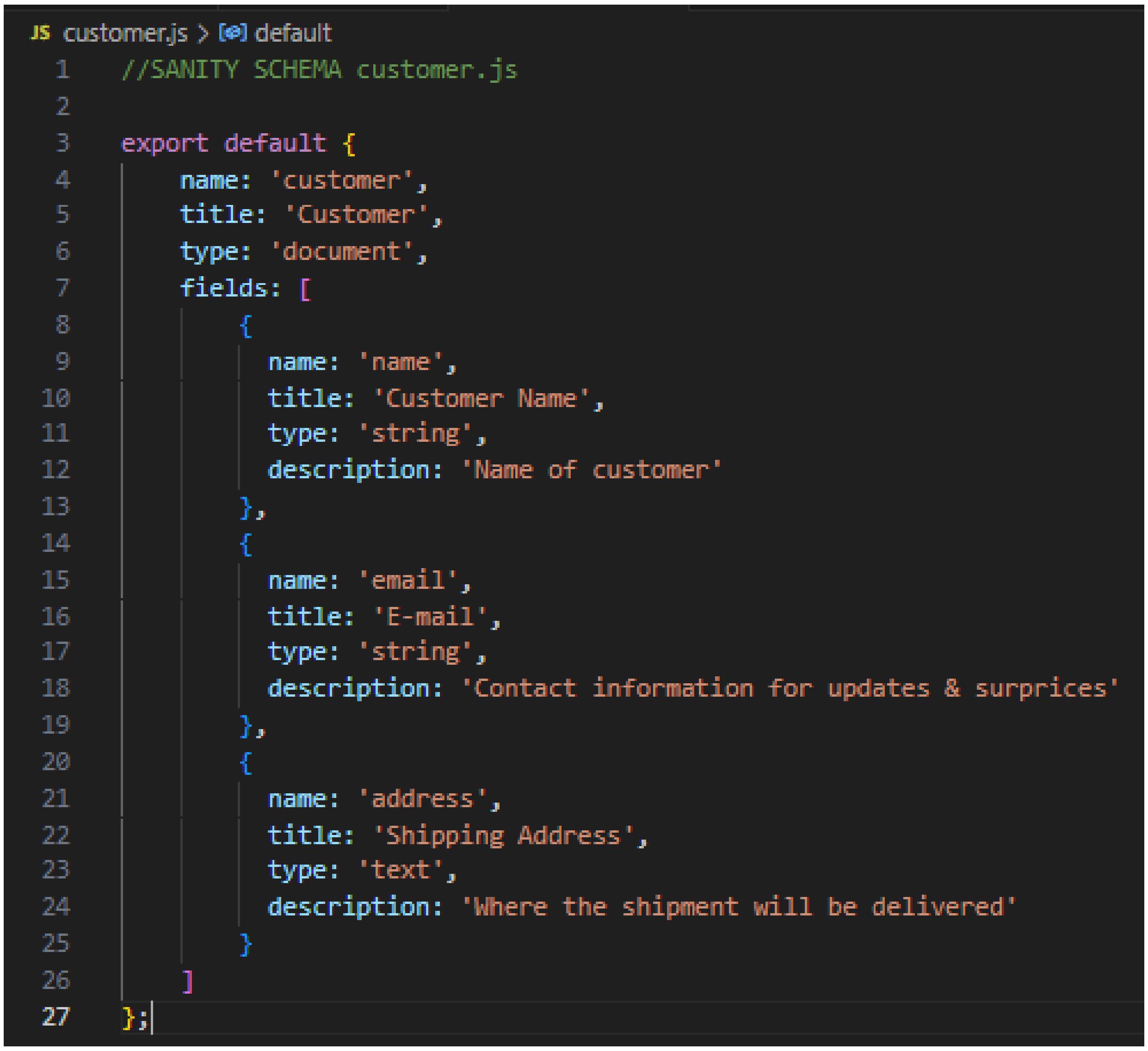
4. Schema:



product.js > default

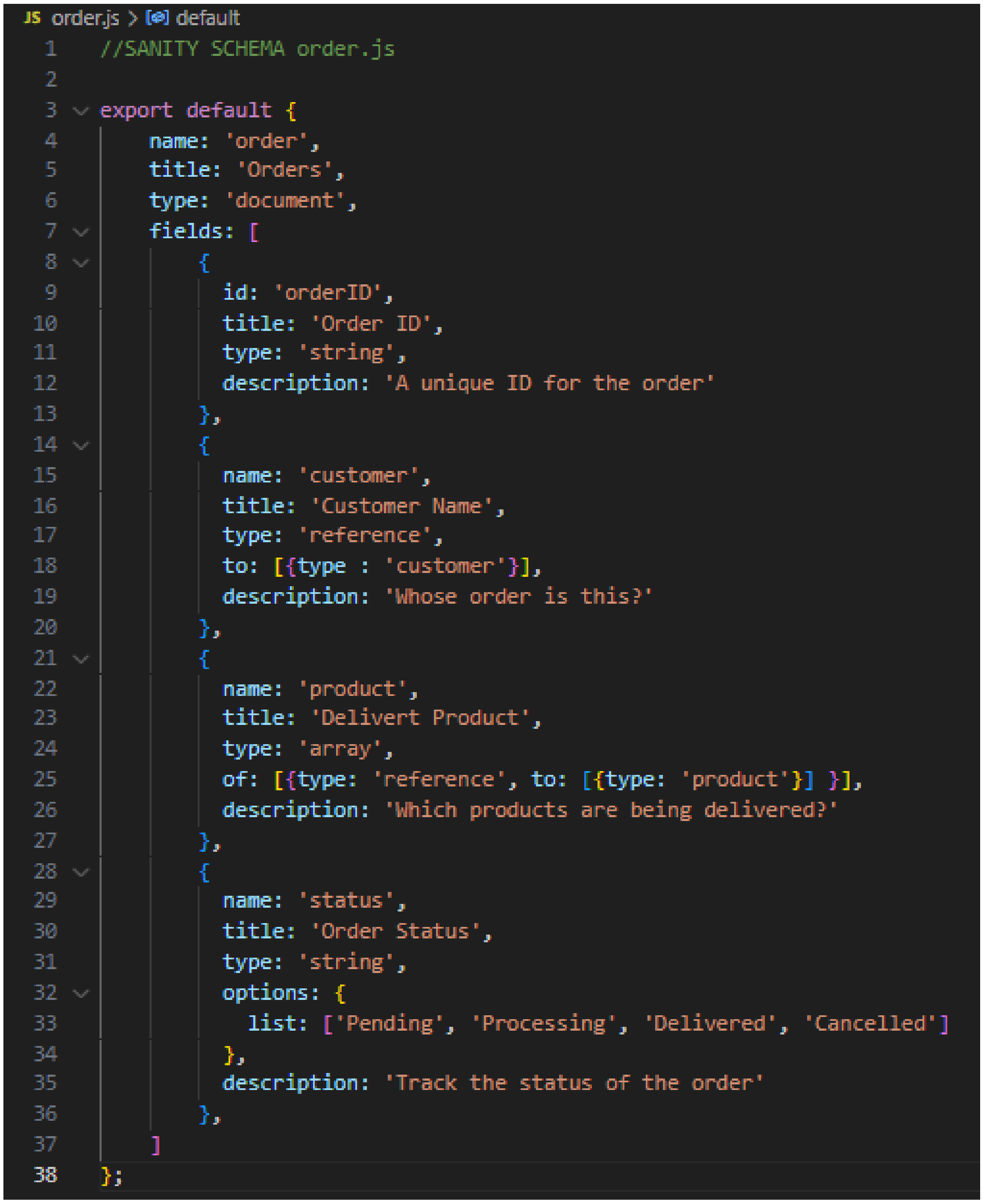
```js
//SANITY SCHEMA product.js

export default {
    name: 'product',
    title: 'Aura',
    type: 'document',
    fields: [
        {
          name: 'name',
          title: 'Watch Name',
          type: 'string',
          description: 'Aura One: Timeless elegance and precision, designed for every moment.'
        },
        {
          name: 'description',
          title: 'Description',
          type: 'string',
          description: 'Describe about product'
        },
        {
          name: 'price',
          title: 'Price',
          type: 'number',
          description: 'Price of product'
        },
        {
          name: 'image',
          title: 'Product Image',
          type: 'image',
          description: 'A high-quality image of the product'
        },
        {
          category: 'category',
          title: 'Category',
          type: 'string',
          description: 'type of product (e.g. watch,bracelet, mini perfume)'
        },
    ]
};
```



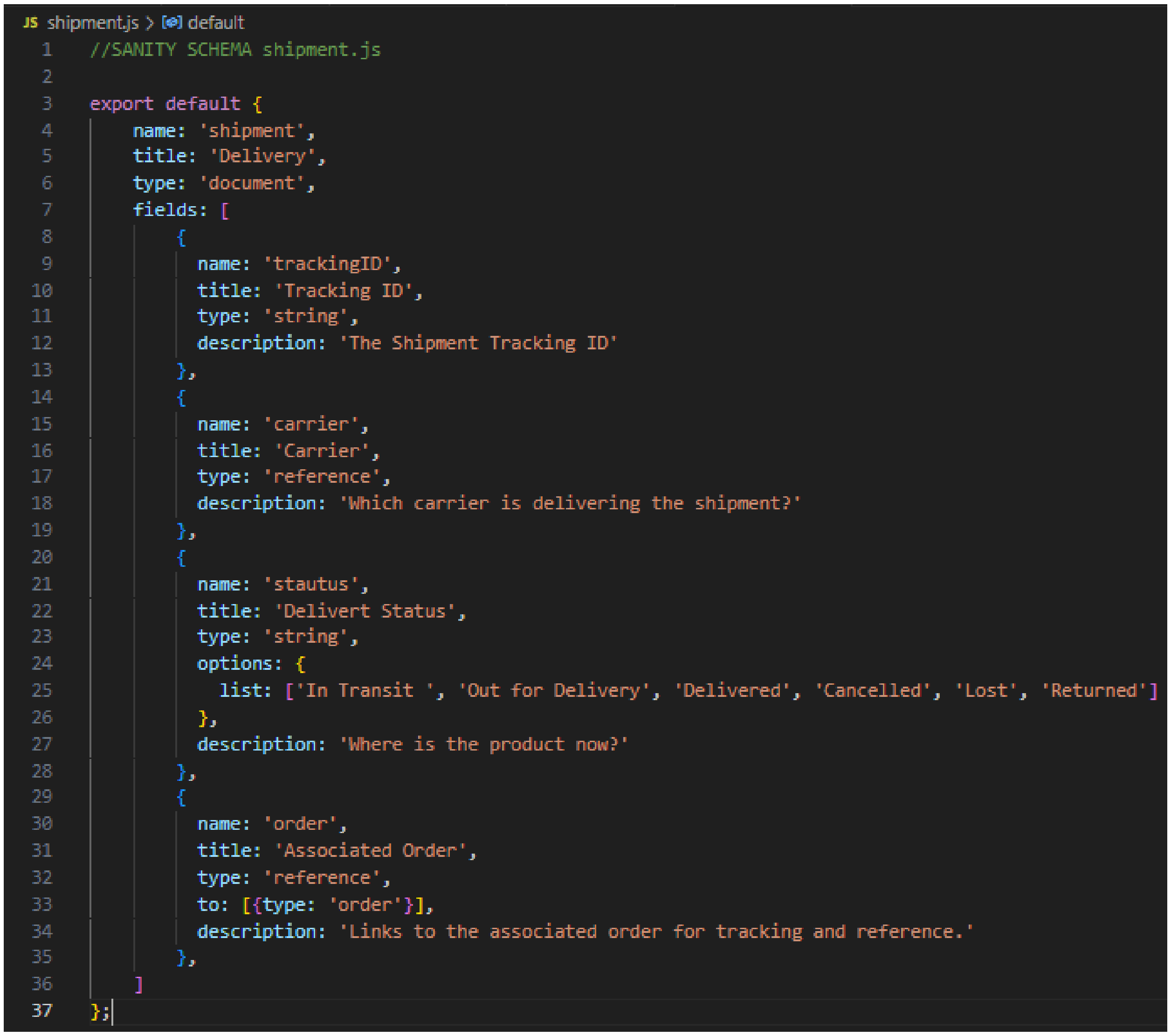
```
//SANITY SCHEMA customer.js

export default {
    name: 'customer',
    title: 'Customer',
    type: 'document',
    fields: [
        {
          name: 'name',
          title: 'Customer Name',
          type: 'string',
          description: 'Name of customer'
        },
        {
          name: 'email',
          title: 'E-mail',
          type: 'string',
          description: 'Contact information for updates & surprices'
        },
        {
          name: 'address',
          title: 'Shipping Address',
          type: 'text',
          description: 'Where the shipment will be delivered'
        }
    ]
};
```



```
//SANITY SCHEMA order.js

export default {
    name: 'order',
    title: 'Orders',
    type: 'document',
    fields: [
        {
          id: 'orderID',
          title: 'Order ID',
          type: 'string',
          description: 'A unique ID for the order'
        },
        {
          name: 'customer',
          title: 'Customer Name',
          type: 'reference',
          to: [{type : 'customer'}],
          description: 'Whose order is this?'
        },
        {
          name: 'product',
          title: 'Delivert Product',
          type: 'array',
          of: [{type: 'reference', to: [{type: 'product'}] }],
          description: 'Which products are being delivered?'
        },
        {
          name: 'status',
          title: 'Order Status',
          type: 'string',
          options: {
            list: ['Pending', 'Processing', 'Delivered', 'Cancelled']
          },
          description: 'Track the status of the order'
        },
    ]
};
```



```
//SANITY SCHEMA shipment.js

export default {
    name: 'shipment',
    title: 'Delivery',
    type: 'document',
    fields: [
        {
          name: 'trackingID',
          title: 'Tracking ID',
          type: 'string',
          description: 'The Shipment Tracking ID'
        },
        {
          name: 'carrier',
          title: 'Carrier',
          type: 'reference',
          description: 'Which carrier is delivering the shipment?'
        },
        {
          name: 'stautus',
          title: 'Delivert Status',
          type: 'string',
          options: {
            list: ['In Transit ', 'Out for Delivery', 'Delivered', 'Cancelled', 'Lost', 'Returned']
          },
          description: 'Where is the product now?'
        },
        {
          name: 'order',
          title: 'Associated Order',
          type: 'reference',
          to: [{type: 'order'}],
          description: 'Links to the associated order for tracking and reference.'
        },
    ]
};
```