Hackathon 3

Day 2

GENERAL E COMMERCE

NAME: Umm E Aeiman

ROLL NO: 00265615

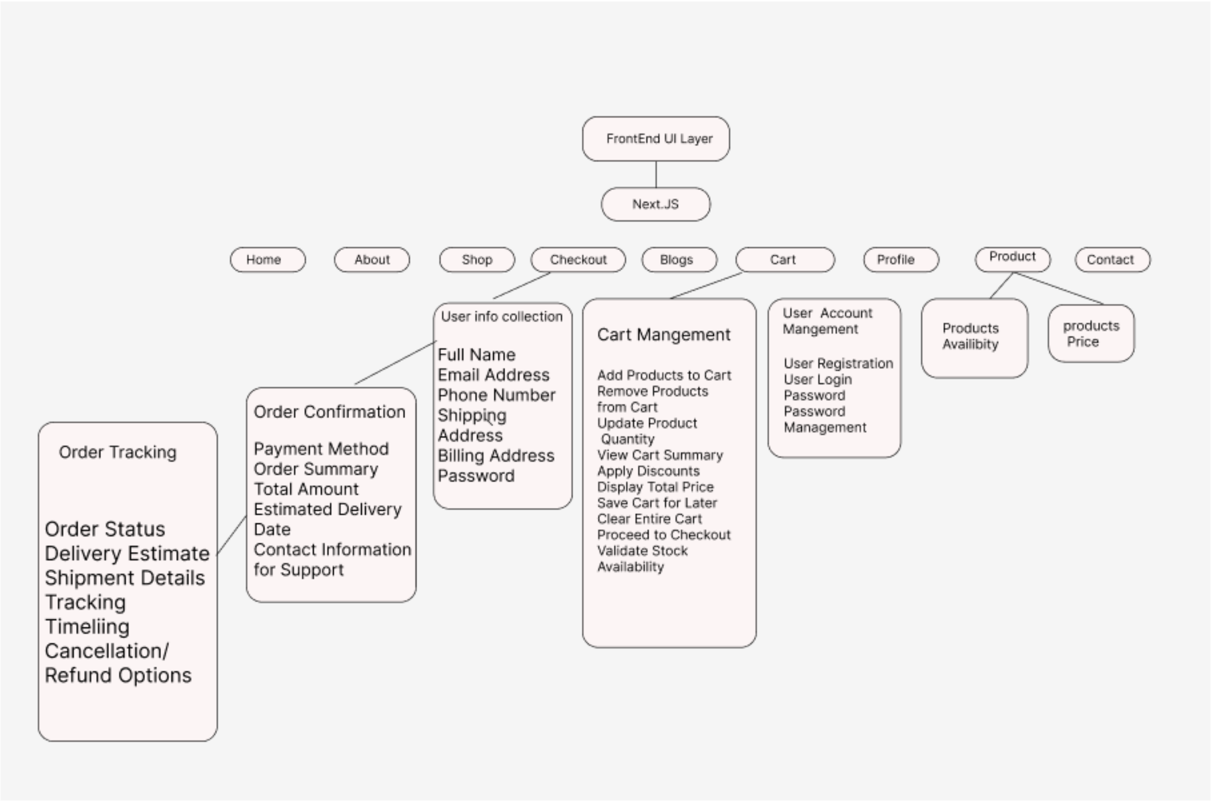
TIMMING: Sat (9.00 am to 12.00pm)

Day_2_MarketPlace_Technical_Foundation

General E Commerce: Niche (Clothing)

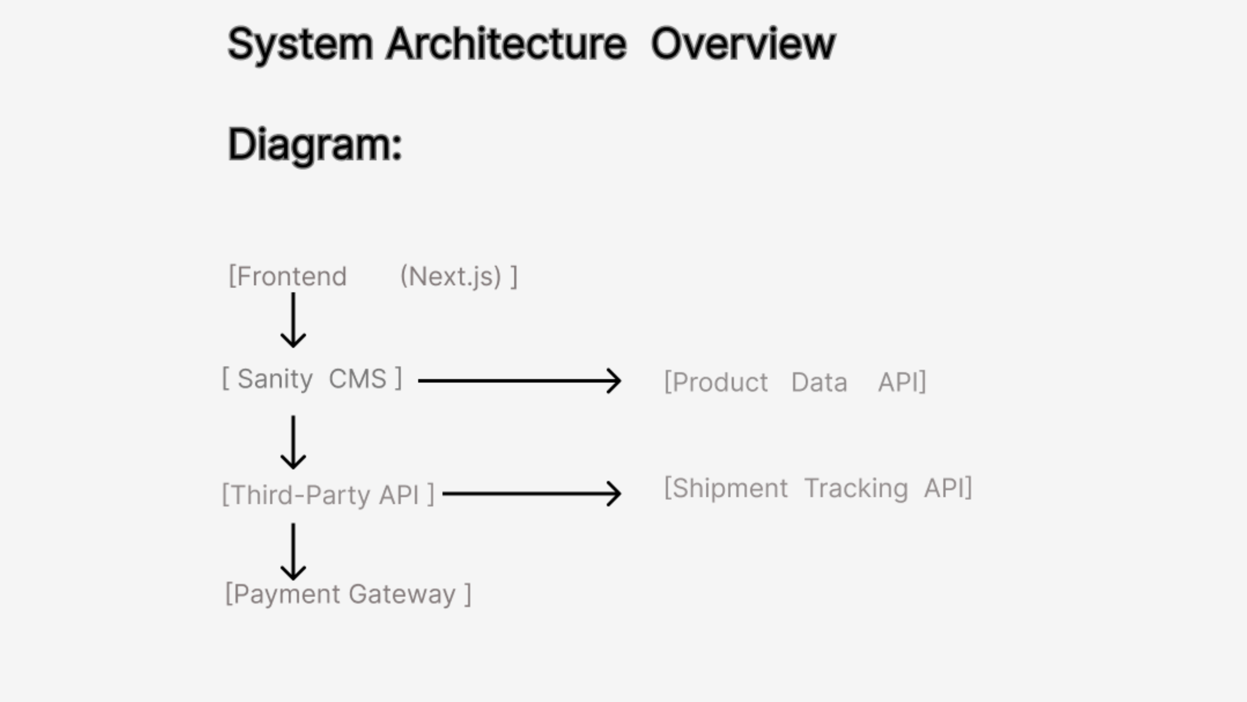
Template :05

MARKETPLACE TECHNICAL FOUNDATION_[Steemaur]



Frontend Requirments:

Design System Architecture:



System Architecture Overview

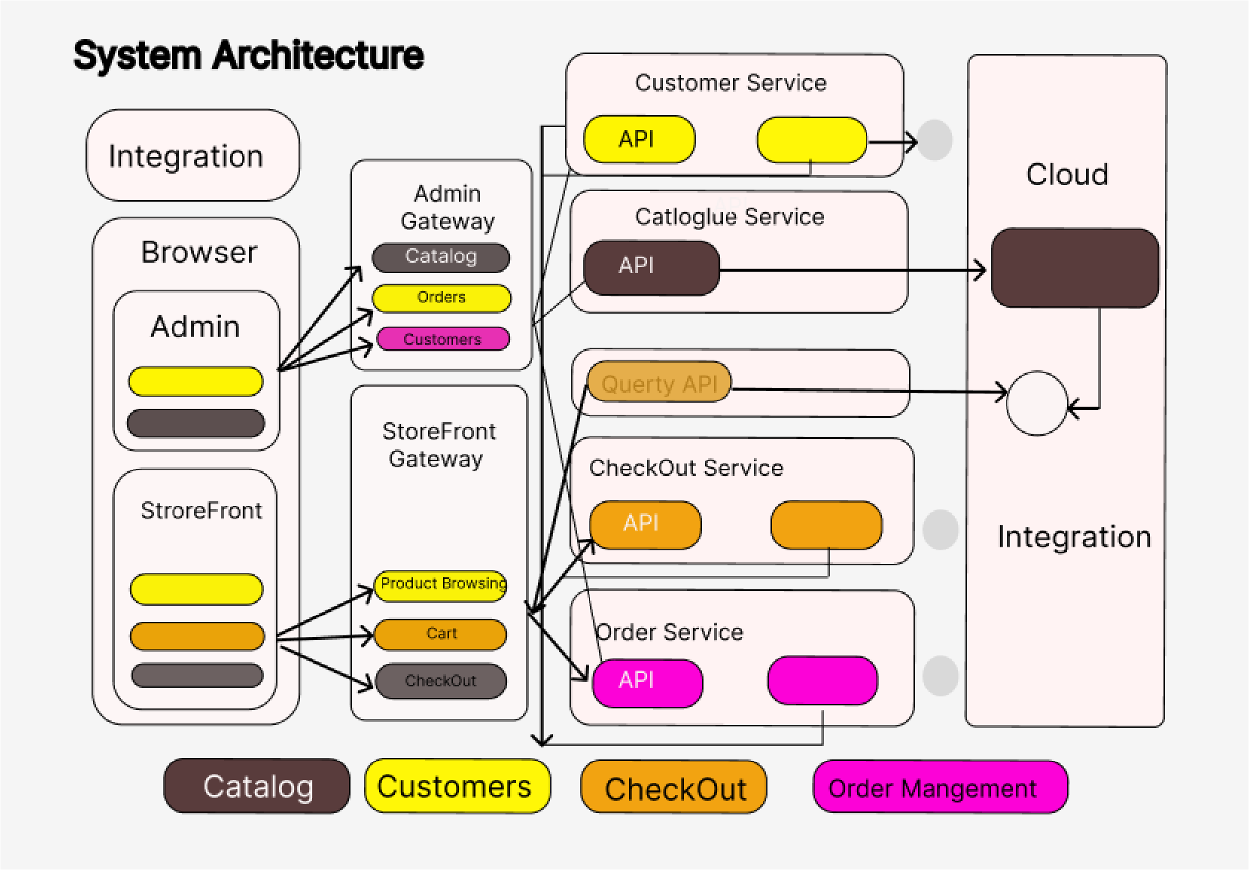
Diagram:

[Frontend (Next.js)]

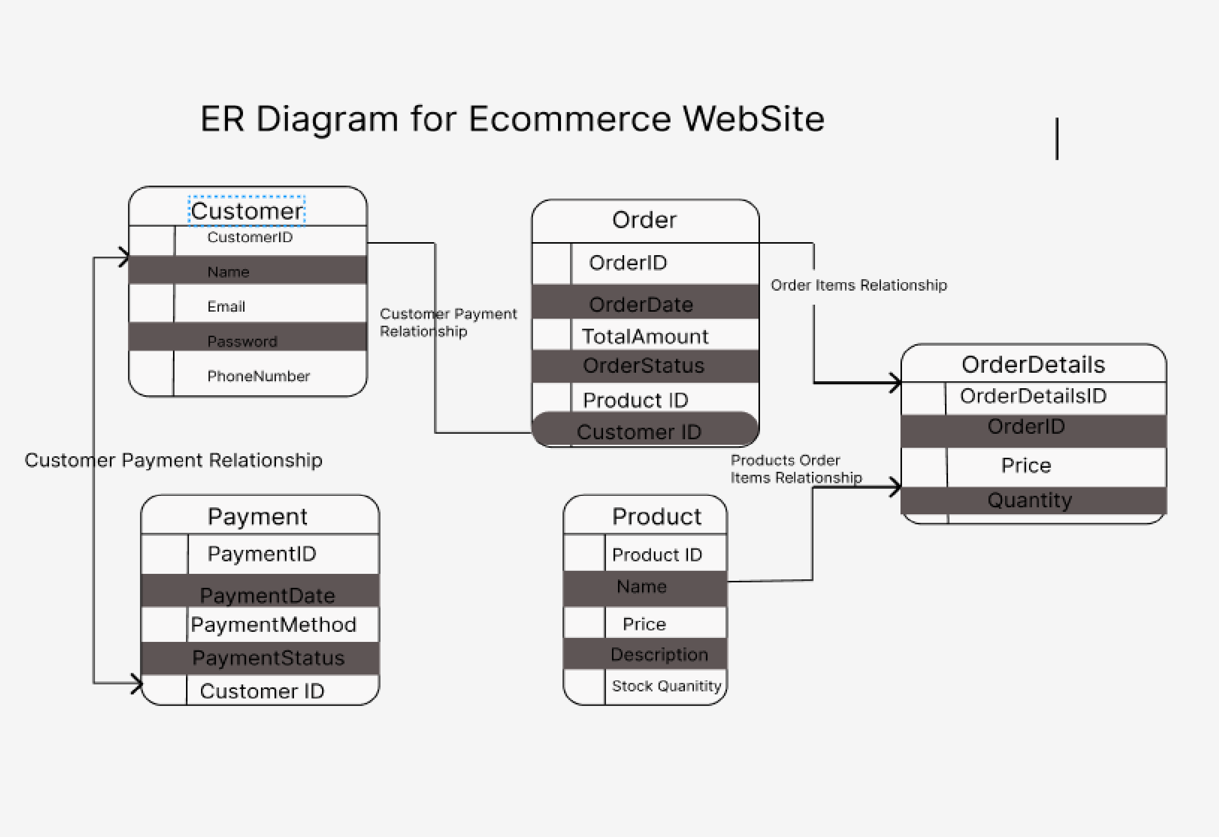
[Sanity CMS] → [Product Data API]

[Third-Party API] → [Shipment Tracking API]

[Payment Gateway]

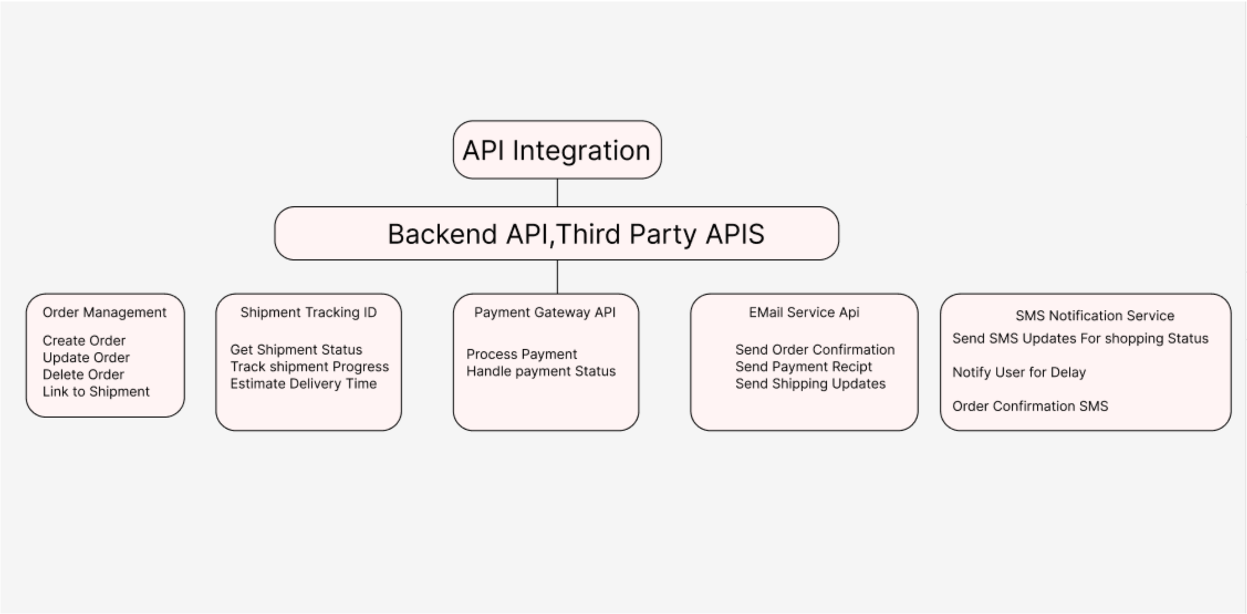

System Architecture
Integration
Browser
Admin
StroreFront
Admin Gateway
Catalog
Orders
Customers
StoreFront Gateway
Product Browsing
Cart
CheckOut
Customer Service
API
Catloglue Service
API
Querty API
CheckOut Service
API
Order Service
API
Cloud
Integration
Catalog
Customers
CheckOut
Order Mangement

EDR DIAGRAM:

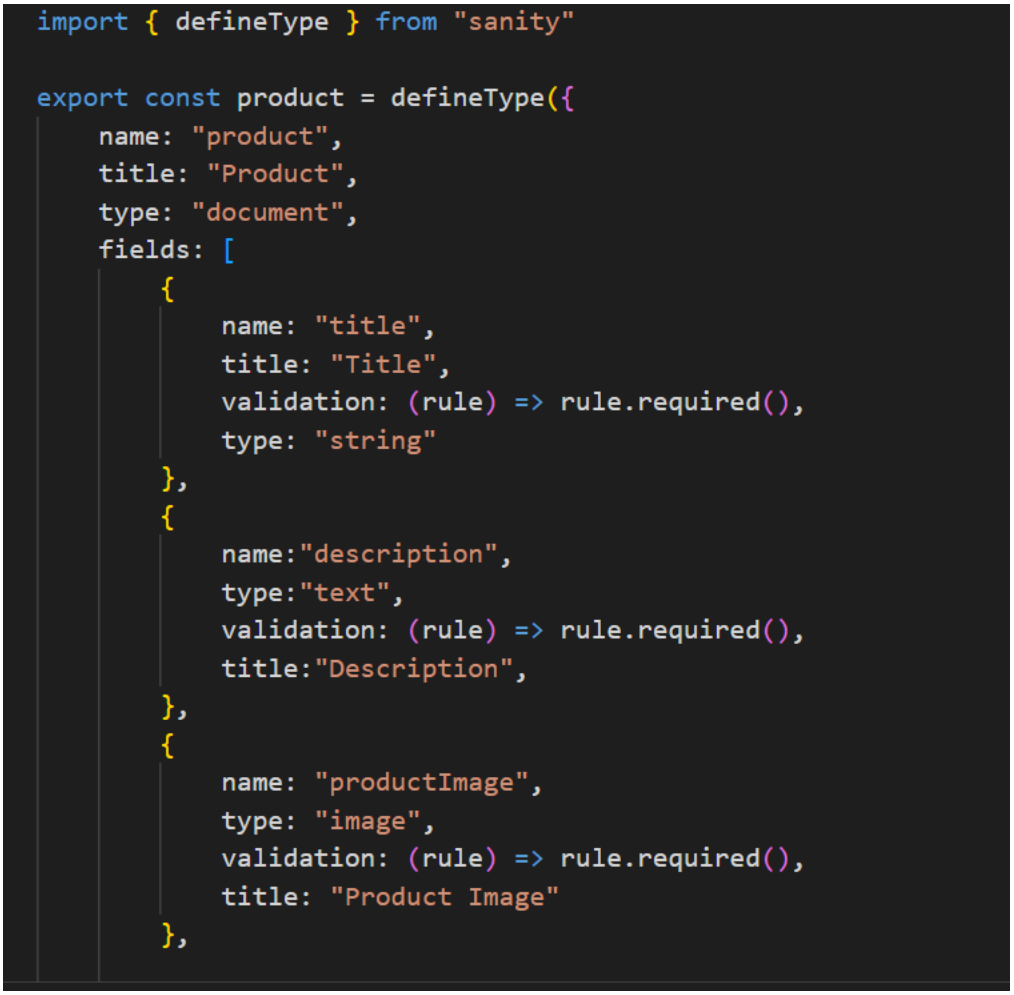


ER Diagram for Ecommerce WebSite

API Integration;

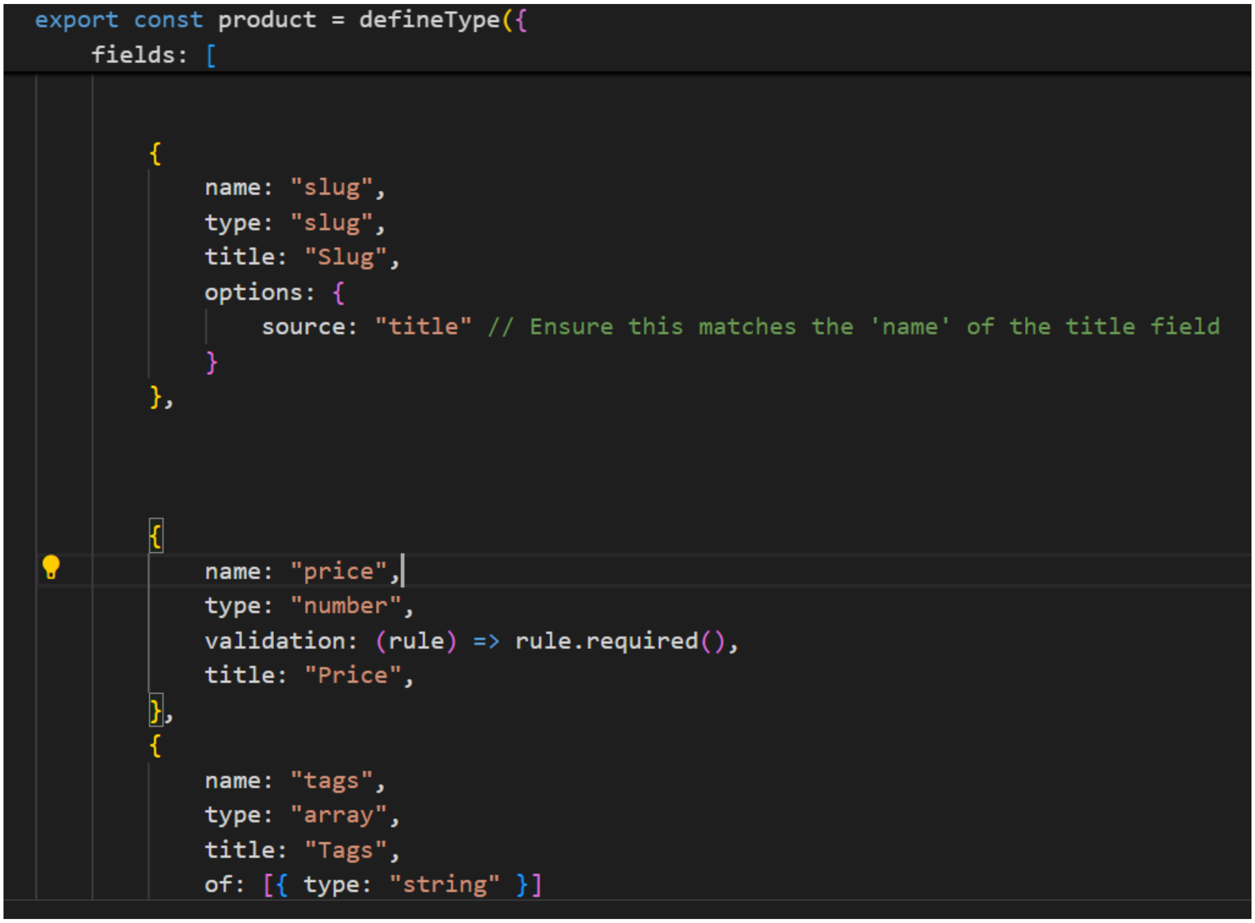


Schema;



```
import { defineType } from "sanity"

export const product = defineType({
    name: "product",
    title: "Product",
    type: "document",
    fields: [
        {
            name: "title",
            title: "Title",
            validation: (rule) => rule.required(),
            type: "string"
        },
        {
            name:"description",
            type:"text",
            validation: (rule) => rule.required(),
            title:"Description",
        },
        {
            name: "productImage",
            type: "image",
            validation: (rule) => rule.required(),
            title: "Product Image"
        },
```

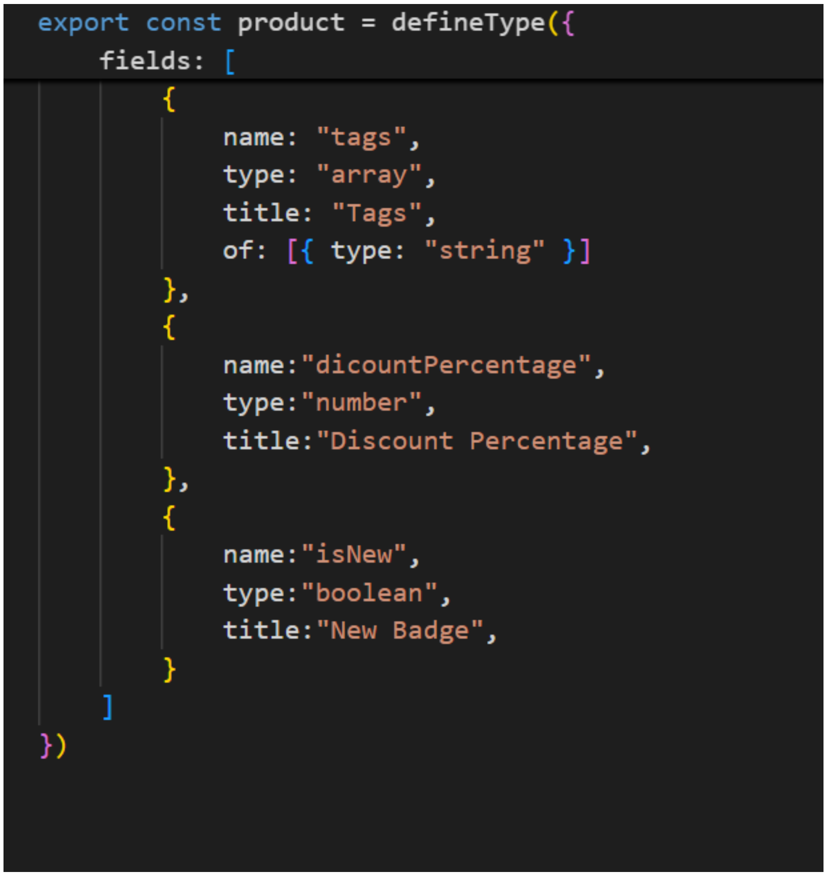


```
export const product = defineType({
    fields: [


        {
            name: "slug",
            type: "slug",
            title: "Slug",
            options: {
                source: "title" // Ensure this matches the 'name' of the title field
            }
        },



        {
            name: "price",
            type: "number",
            validation: (rule) => rule.required(),
            title: "Price",
        },
        {
            name: "tags",
            type: "array",
            title: "Tags",
            of: [{ type: "string" }]
```



```
export const product = defineType({
    fields: [
        {
            name: "tags",
            type: "array",
            title: "Tags",
            of: [{ type: "string" }]
        },
        {
            name:"dicountPercentage",
            type:"number",
            title:"Discount Percentage",
        },
        {
            name:"isNew",
            type:"boolean",
            title:"New Badge",
        }
    ]
})
```

APIENDPOINT.xlsx

Endpoint	Method	Description	Parameters	ResponseExample
/api/foods	**GET**	Fetch all products items	No	{id:1, name: “pizza”}
/api/foods/:id	**GET**	fetch a single product item	Id(path)	{id: 1, name: “pizza”}
/api/foods	**POST**	Add a new product item	Name, price, category, (Body)	{success: true ,id : 5}
/api/foods/:id	**PUT**	Update a product item	Id(path) name,price (Body)	{success,true}
/api/foods/:id	**DELETE**	Delete a product item	Id(path)	{success,true}
/api/categories	**GET**	Fetch all products categories	None	{categories:[“Drinks”]}