Marketplace Technical Foundation For General E-Commerce

Topics

Overview:

This is full flash documentation related to a startup marketplace (type General E-commerce). This is a part of a Hacakthon. In this documentation I have done brainstorming about the technical requirements , Api endpoints, system workflow and what technologies I Am going to use in this hackathon.

Technologies:

For this project I will be using the following technologies:
1-**Next.js :** I will be using next.js as it is a full stack framework.
2- **Shadcn.UI:** I will be using shadcn or other libraries as they make working more easy.
3-**OpenAI :** I will use OpenAI for integration of a simple chatbot in my app.
4-**Shipping:** I will use shipengine for the shipping API.
5-**Vercel:** I will utilize vercel for deployment purposes.
6-**Github:** For storing my code Help Helping me deploy it.

Technical Requirements:

My technical requirements are described as follows:

Frontend Requirements:

My frontend will include sleek design and the following features:
1-user friendly interface for best customer experience.
2-Frontend design should be responsive for all screen sizes.
3-All of the important pages will exist in my design like home,about, product listing, contact ,cart,checkout etc.

Backend Requirements:

I will use sanity cms as my backend for now:
1- I will use sanity methods like delete,patch,fetch etc to manage my backend
2- Using sanity I will manage my cart order shipment and order tracking.

Third-Party APIs:

1- I will integrate the api provided in the hackathon and use it in product listing.
2- I will use shipment order tracking api for shipment and order tracking
2- I will ensure that the APIs provide the necessary data for frontend functionality.

Integration Of AI chatbot:(optional)

1-I will either use open AI or the gemini api for making Ai assistant.
2-I will train it related to my product categories.

Deployment:

1- For Deploying the market place I will use vercel.

API Requirement:

Products

Endpoint Name: /products
Method: GET
Description: Fetch all the product details.
Response:{
_id:"aoeiu 748-9jjd",
productName:"watch",
produductDescription:"This is a simple watch which is highly affordable",
rating:4.3,
price:30 PKR,
image:"image url from sanity"
};

Orders:

Endpoint Name: /orders
Method: POST
Description: Creating a new order in Sanity cms.
Response: {
_id:`1234`,
CustomerAddress:`xyz road karachi pakistan`,
purchasedProduct:`watch`,
payment status:`pending`
}

Shipment:

Endpoint Name: /shipment
Method: GET
Description: Track order status via API.
Response: {
Shipment ID:`xycgdh-22`,
order ID:`1234`,
Status:`shipping`,
delivery date:4
}
This type of Api will be provided through shipengine or can have more or less fields.

System workflow:

Login/signup:

First off all the user will have to sign up to the website without signup/login most of the functions will not proceed forward.

Home page:

After the signup/login process is completed The user will receive a popup or a message that their login/sign up is successful then they will have full access. Now they can navigate back to the home page.

Add to cart:
When a customer selects a product and adds it to cart the functionality will run and the items will be added to cart and stored into the local storage so that the products won't disappear if the user accidentally hits refresh.

Checkout:
After the customer is fully satisfied with their cart they will have the check out feature.

Payment :
Payments will not be real they are just gonna be for show as there are no real products. There will also be a condition check that if the payments are clear then only proceed with the shipping and if the payments are not clear hold the shipping.

Order confirmation:
After the payment is successful a confirmation message will be shown to the customer.

Shipment:
After the payment is cleared shipment starts to proceed and order tracking will be shown to the customers and the order will be delivered to the customers

Lost Order:
In case the order is lost by any means Then there will be retraking for the order and if The order is found it will be reshipped to the customer but in case of loss the customer will be given a full return .

My Products Schema Sample:

```
export default {
  name: 'product',
  type: 'document',
  title: 'Product',
  fields: [
    {
      name: 'name',
      type: 'string',
```

```
    title: 'Product Name',
},
{
    name: 'description',
    type: 'string',
    title: 'Description'
},
{
    name: 'price',
    type: 'number',
    title: 'Product Price',
},
{
    name: 'discountPercentage',
    type: 'number',
    title: 'Discount Percentage',
},
{
    name: 'priceWithoutDiscount',
    type: 'number',
    title: 'Price Without Discount',
    description: 'Original price before discount'
},
{
    name:'rating',
    type:'number',
    title:'Rating',
    description:'Rating of the product'
},
{
    name: 'ratingCount',
    type: 'number',
    title: 'Rating Count',
    description: 'Number of ratings'
},
{
    name: 'tags',
```

```
      type: 'array',
      title: 'Tags',
      of: [{ type: 'string' }],
      options: {
        layout: 'tags'
      },
      description: 'Add tags like "new arrival", "bestseller", etc.'
    },
    {
      name: 'sizes',
      type: 'array',
      title: 'Sizes',
      of: [{ type: 'string' }],
      options: {
        layout: 'tags'
      },
      description: 'Add sizes like S , M , L , XL , XXL'
    },
    {
      name: 'image',
      type: 'image',
      title: 'Product Image',
      options: {
        hotspot: true // Enables cropping and focal point selection
      }
    }
  ]
};
```

Visualization of my system workflow: