Day 2 -Planning the technical foundation.

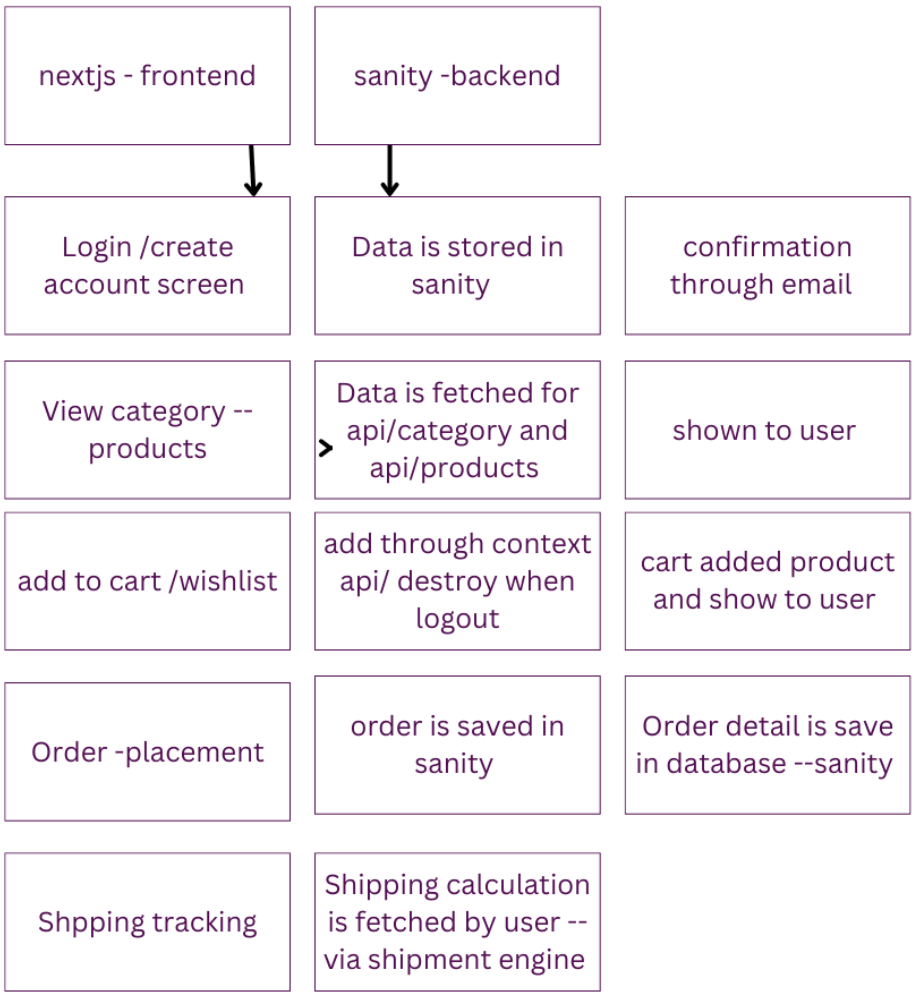
https://www.figma.com/design/nMFNhaol1HtlwWAwpsksuj/Project---Ecom-(An-Ecommerce-Ui-Kit)-(Community)-(Copy)?node-id=1-38&p=f&t=KXjtVjlzQG90EBfr-0

we are making functional requirement for above mentioned figma file.

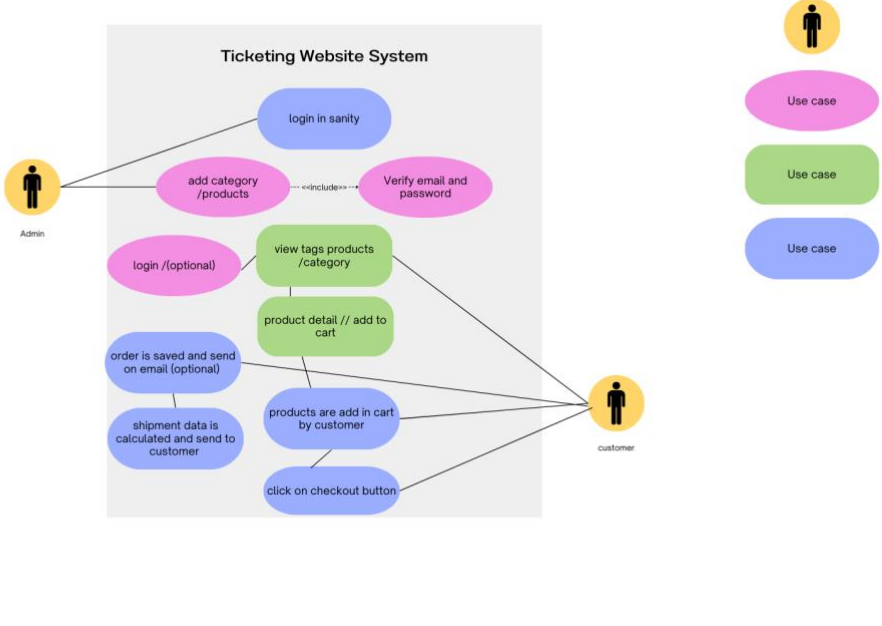
1. Frontend Requirements
    a. TopNavbar: common in all .these two contain dropdown for changing language which can be localization “i8ln” and currency change – these two feature can be done in future. Also it contain cart and wishlish which will done through context api (like adding product in wishlist or cart)
    b. Navbar : These contain multiple pages
    c. Footer : These contain pages link.
    d. Home page : Now on home page there is a section for featured products , latest products , trending product –(these are product which will fetch from database using tags like featured latest and trending and category when on clicking category – it will shifted to product page using dynamic routing.
    e. Product page: Product detail page contains view as ascending or descending order feature also it contains product under category .
    f. Product detail page contains detail information of product and add to cart functionality . On clicking product will be added to cart.On clicking wishlish product will add to wishlist as well
    g. Now clicking on cart button on top we will directed to cart page. Now total and subtotal will be calculated and clicking on calculate shipping button we will be calculating shipping will calculated based on by shipengine.
    h. The product and total and subtotal will be added to orders .
    i. The products are added in cart and wish list . Note they will be cleared when user logout.
    j. Here we are keeping cart and wish list in contextapi for global state management and the reason is that we don’t want to unnecessary on database and also it will time to fetch it.
    k. Since we are using sanity there are two users admin who will add category and product and a another user who will see products and add them so we need to track them so will also provide a contact form to save their info.
    l. After order is save a notification will be given “Your order is saved”.
2. Backend --- sanity is being as database.
    a. Category --- schema design by admin
    b. Product --- schema design by admin
    c. Review --- saved to schema by customer.
    d. Order --- saved to schema by customer.
    e. Payment – saved by customer to schema.
3. Third party API
    a. Stripe can be used for handling payments done by cards
    b. Shipment engine to calculate shipment

c. Bandmay can be use for handling loan because furniture price can be costly.
d. Tab ui – react library
e. Star rating library.

System architecture



ecommer
ce



3- Plan API requirement
In here first the data will be fetch from MockAPI,

End point Name:/category
Method :GET
Description: fetch category _id, name, image from sanity
Response Detail:{id:1,category:chair,category:img_url}

End point: Name:/product
Method : GET
Description:fetch product _id , name ,
colours,beforeprice,afterprice,rating,description,catergories_id,tags from sanity
Response detail:{_id:1,name:plamwoodchair,color:[
Red
,yellow,
Blue],beforeprice:321,afterprice : 123 ,rating:4,description:”abc xx”,
categories_id:1,tags:[latest,featured]}

For review:
End point :/review
Method:post
Description : post for each product
Product description : review :{_id,review,rating,p_id}

For certain product based on featured , latest , trending
End point : product?tags=${tags} tags is featured
Method : get

Description : get certain product based on tags
{_id:1,name:plamwoodchair,color:[
Red
,yellow,
Blue],beforeprice:321,afterprice : 123 ,rating:4,description:”abc xx”,
categories_id:1,tags:[featured],r_id:1}

End point:/order
Method :post
Description : create order in sanity
Payload: order:{
Status, date purchased,price
} ,customer info{
First name
Lastname
Address
Address detail:
{first address , second address and city , country}
}} , product detail :{ img , name , color , size , price},subtotal , total},payment:{card type , name , card number , date}

Endpoint Name: /shipment

Method: GET
Description: Track order status via third-party API.
Response: Shipment ID, order ID, status, expected delivery date.

Write Technical Document

System Architecture Document:

1. Here we will be using context api for global state management so that cart value increased or decreased based on add to cart value which is design in product detail page. Same goes for wish list as well
2. Dynamic routing is being use by linking category -> catgory->[id] ->product->[id]
3. APi confriguation defined above.
4. Use cases
    a. As an admin , I add product inside category or use mock api to store data
    b. As an admin I view orders and shiping of each customer
    c. As an admin I do above activity by login in first using email and password
    d. As a customer I view a product inside category or product display on home page

    e. As customer I click on category and its directs me to products page having lots of products
    f. As a customer I click on add to cart and value in cart goes to 1 I , can update or decreased value by – or +
    g. As admin I add stock inside product ,So when product goes out stock it no add to cart is shown.
    h. As a customer after I have add product on cart , I go checkout where I fill information (email ,password , address) and it provide total value and upon clicking calculate it calculate amount based on address and courier service.
    i. A order is confirmed I receive email regarding order.
5. Technical milestone
    a. First define schema --- refined it
    b. Next define mock api – may be problem because it give paid after two api made
    c. Next migrate mock api on sanity
    d. Next output saved product on sanity.
    e. Next make dynamic routing using category -> category->[id]->page.tsx ->product->[id]
    f. Next create context api --- add to cart
    g. Optional integrate --- bandmay api
    h. Save cart info in order schema
    i. Next use shipping to calculate –
    j. Next email