Hackathon

DAY 2 PLANNING THE TECHNICAL FOUNDATION

Planning the Technical Foundation

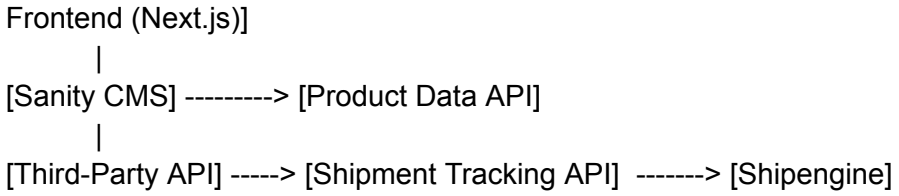
1. **Define Technical Requirements:-**

- **Frontend Requirements:**
  Utilize Next.js, HTML, TypeScript, and Tailwind CSS to build a responsive dynamic, and user-friendly interface.

- **Sanity CMS as Backend:**
  Utilize **Sanity CMS** for streamlined and efficient management of backend operations, including product listings, order data, and other dynamic content.

- **Third-Party APIs:**
  MockAPI:
  For managing and simulating product data.
  ShipEngine:
  To handle the shipment process efficiently.
  Sanity CMS:
  For dynamic content management and backend operations.

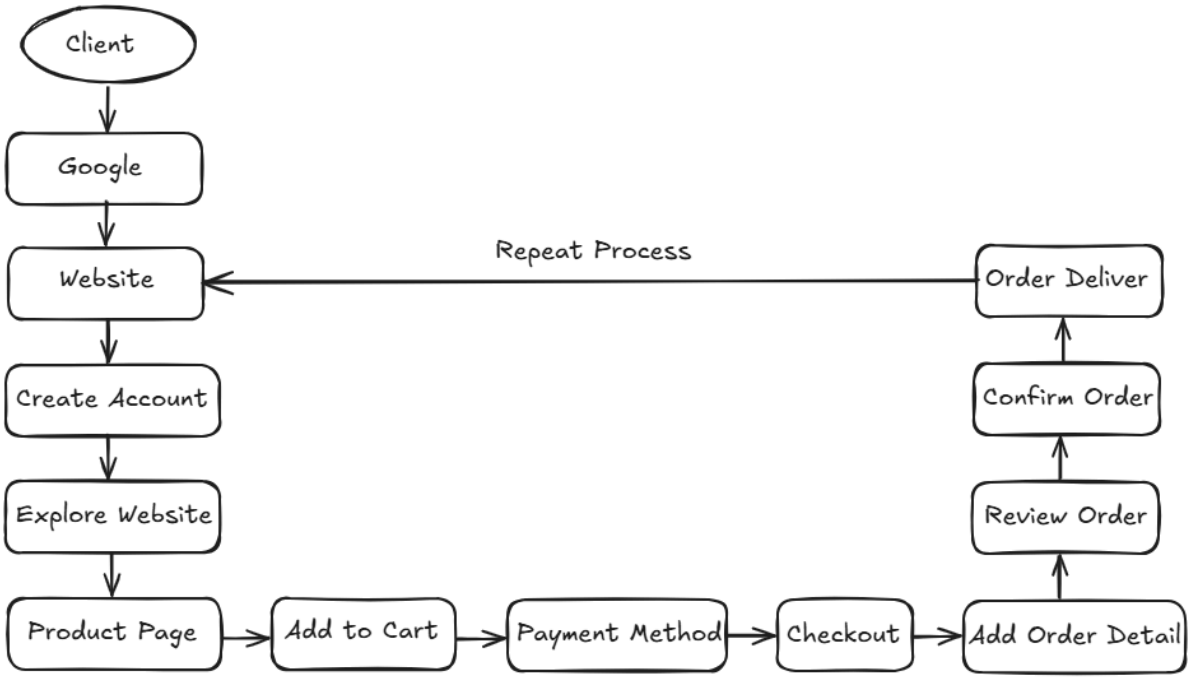
2. **Design System Architecture:-**



3. **Website Overflow:-**

How Our Website Works:

- **Image**



- **Detail:**
  Here's a short description of the user flow for a client confirming their order on the sofa website:

1. **Client searches on Google**: The client uses Google to find the website or hears about the website from someone and decides to visit it.
2. **Visit the website**: They land on the homepage and start exploring.
3. **Account Creation/Login**: New users create an account while returning users login.
4. **Explore Website**: The client browses through the various categories and products.
5. **Product Page**: They navigate to a product page to view details about the selected item.
6. **Add to Cart**: The client adds the product to their cart.
7. **Select Payment Method**: They choose their preferred payment method.
8. **Checkout**: The client proceeds to the checkout page.
9. **Add Order Detail**: They enter necessary order details like shipping address.
10. **Review Order**: Before finalizing, the client reviews their order to ensure accuracy.
11. **Confirm Order**: The client confirms the order.
12. **Order Delivered**: The product is delivered to the client.

4. **Data Schema:-**

Product Schema:

```
import { defineType,defineField } from "sanity";



export const product = defineType ({

    name: 'product',

    type: 'document',

    title: 'Product',

    fields: [

        defineField(

      {

        name: 'title',

        type: 'string',

        title: 'Product_name',

        validation: Rule => Rule.required(),

      },

       validation:Rule => Rule.required()

      ),

      defineField(

      {

        name: 'description',

        type: 'string',

        title: 'Description'

      },

       validation:Rule => Rule.required()
```

```
    ),
    defineField(
    {
      name: 'price',
      type: 'number',
      title: 'Product Price',
      validation:Rule => Rule.required()
    },

    defineField(
    {
      name: 'stock_availability',
      type: 'number',
      title: 'Stock Availability',
    },
     validation:Rule => Rule.required()
     ),
    defineField(
    {
      name:'rating',
      type:'number',
      title:'Rating',
      description:'Rating of the product'
    },
      validation:Rule => Rule.required()
```

```
    ),
    defineField(
    {
      name: 'tags',
      type: 'array',
      title: 'Tags',
      of: [{ type: 'string' }],
      options: {
        layout: 'tags'
      },
      description: 'Add tags like "new arrival", "bestseller", etc.'
    },
      validation:Rule => Rule.required()
    ),
    defineField(
    {
      name: 'image',
      type: 'image',
      title: 'Product Image',
      options: {
        hotspot: true // Enables cropping and focal point selection
      },
    },),
    defineField(
      {
```

```
            name:'slug',
            type:'slug',
            title:'Slug',
            options:{
                source:'title',
                maxLength:96
            },
            validation:Rule => Rule.required()
        },
    ),
```