

## Execution of Day 2 Tasks: Planning the Technical Foundation

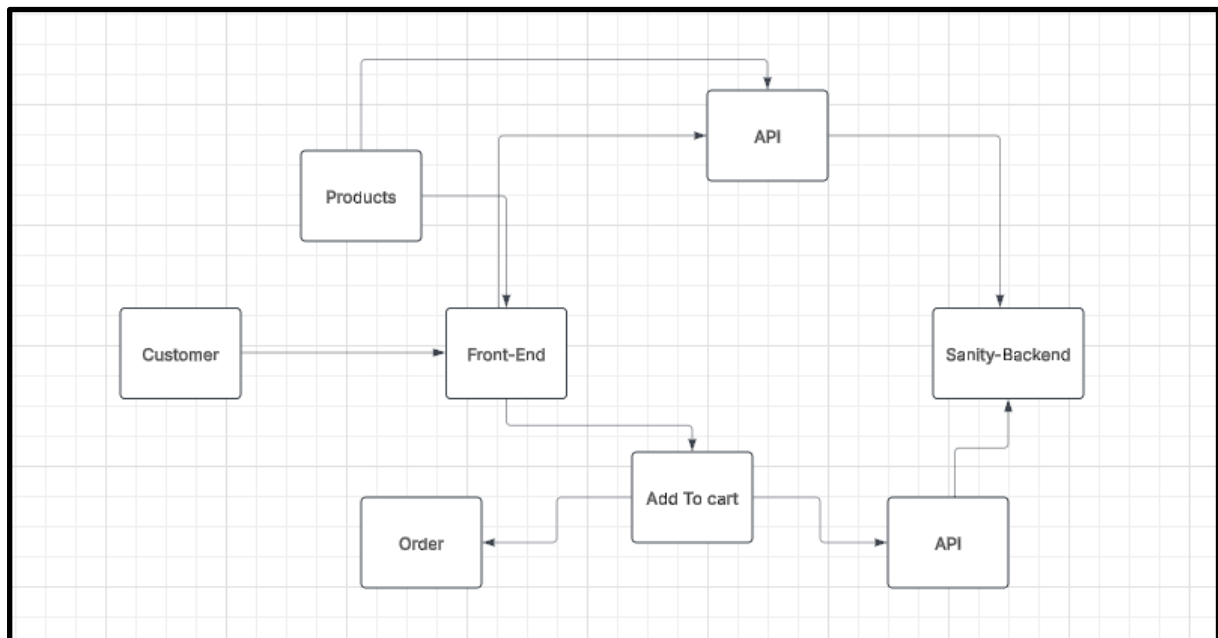
---

### 1. Technical Plan Aligned with Business Goals

For a **General E-Commerce Marketplace**, the technical plan includes:

- **Frontend:**
    - A responsive web application using **NextJS** for intuitive product browsing, cart management, and order placement.
    - Integration of advanced search and filtering capabilities (e.g., by category)..
  - **Sanity CMS:**
    - Centralized content management for product details, customer reviews, and promotional banners.
  - **Third-Party APIs:**
    - Integration with payment gateways (e.g., Stripe, PayPal) for secure transactions.
    - Order APIs for Orders and Checkouts success.
- 

### 2. System Architecture Visualized



### 3. API Requirements :

#### 1. Endpoint Name: **/products**

- **Method:** **GET**
- **Description:** Fetch all available products from Sanity CMS.

- **Response:** Returns a list of product details, including ID, name, price, stock, and image.

```

▼ 2: {...} 15 properties
  _id: a20a75fa-6f2f-4603-a16e-d7013f1ae7ae
  ▶ categories: [...] 1 item
  _type: product
  description: WEB DEVELOPMENT ARTIFICIAL INTELLIGENCE
  _updatedAt: 2025-02-05T15:42:29Z
  ▶ image: {...} 2 properties
    _rev: XbbB8rplnhWhePlT6odi9n
    label: 256GB
    name: Apple Tablet Book 1
    status: NEW
    discount: 25
    price: 115
    _createdAt: 2025-02-05T15:41:28Z
    stock: 55
  ▼ slug: {...} 2 properties
    current: apple-tablet-book-1
    _type: slug

```

## 2. Endpoint Name: `/api/webhook`

- **Method:** **POST**
- **Description:** Create a new order in Sanity CMS.
- **Payload:** Includes customer information, product details, and payment status etc.

```

const order = await backendClient.create({
  _type: "order",
  orderNumber,
  stripeCheckoutSessionId: id,
  stripePaymentIntentId: payment_intent,
  customerName,
  stripeCustomerId: customerEmail,
  clerkUserId: clerkUserId,
  email: customerEmail,
  currency,
  amountDiscount: total_details?.amount_discount ? total_details.amount_discount / 100 : 0,
  products: sanityProducts,
  totalPrice: amount_total ? amount_total / 100 : 0,
  status: "paid",
  orderDate: new Date().toISOString(),
});

```

## 3. Endpoint Name: `/orders`

- **Method:** **GET**
- **Description:** Fetch all placed Orders from Sanity CMS.

- **Response:** Returns a list of orders , including ID, name, price, status, and image.

```
{
  products: [ [Object], [Object] ],
  totalPrice: 154,
  stripeCheckoutSessionId: 'cs_test_b1z7BJVvVHLpxok30kvPIhoCSK40JeJrweACO0KjOBE75eCE46mj2gRpTo',
  stripeCustomerId: 'rameezaziz2001@gmail.com',
  _updatedAt: '2025-02-06T14:32:43Z',
  _originalId: 'b8Q1vNZ2RcdM6Ir5Bk7gyo',
  clerkUserId: 'user_2sZ1sn18qWQI6nU92qvyNNGqE81',
  _rev: 'b8Q1vNZ2RcdM6Ir5Bk7gue',
  amountDiscount: 0,
  orderNumber: '42df71ae-7752-4e4b-9567-8a2925973d52',
  email: 'rameezaziz2001@gmail.com',
  _type: 'order',
  status: 'paid',
  stripePaymentIntentId: 'pi_3QpW8CG2wRcpc91d10LzW1jP',
  currency: 'usd',
  customerName: 'Rameez Aziz',
  _createdAt: '2025-02-06T14:32:43Z',
  _id: 'b8Q1vNZ2RcdM6Ir5Bk7gyo',
  orderDate: '2025-02-06T14:32:43.304Z'
}
```

## 1. Endpoint Name: `/product/[slug]`

- **Method:** GET
- **Description:** Fetch product based on its name from Sanity CMS.
- **Response:** Returns a product details, including ID, name, price, stock, and image.

```
▼ 2: {...} 15 properties
  _id: a20a75fa-6f2f-4603-a16e-d7013f1ae7ae
  ► categories: [...] 1 item
    _type: product
    description: WEB DEVELOPMENT ARTIFICIAL INTELLIGENCE
    _updatedAt: 2025-02-05T15:42:29Z
  ► image: {...} 2 properties
    _rev: XbbB8rplnhWhePlT6odi9n
    label: 256GB
    name: Apple Tablet Book 1
    status: NEW
    discount: 25
    price: 115
    _createdAt: 2025-02-05T15:41:28Z
    stock: 55
  ▼ slug: {...} 2 properties
    current: apple-tablet-book-1
    _type: slug
```

## 1. Endpoint Name: `/categories/[slug]`

- **Method:** GET
- **Description:** Fetch all available products related to category from Sanity CMS.
- **Response:** Returns a list of product details, including ID, name, price, stock, and image.

```

▼ 2: {...} 15 properties
  _id: a20a75fa-6f2f-4603-a16e-d7013f1ae7ae
  ▶ categories: [...] 1 item
    _type: product
    description: WEB DEVELOPMENT ARTIFICIAL INTELLIGENCE
    _updatedAt: 2025-02-05T15:42:29Z
  ▶ image: {...} 2 properties
    _rev: XbbB8rp1nhWhePlT6odi9n
    label: 256GB
    name: Apple Tablet Book 1
    status: NEW
    discount: 25
    price: 115
    _createdAt: 2025-02-05T15:41:28Z
    stock: 55
  ▼ slug: {...} 2 properties
    current: apple-tablet-book-1
    _type: slug

```

## 1. Schema Designs:

### 1. Category Schema :

```

export const categoryType = defineType({
  name: 'category',
  title: 'Category',
  type: 'document',
  icon: TagIcon,
  fields: [
    defineField({
      name: 'title',
      type: 'string',
    }),
    defineField({
      name: 'slug',
      type: 'slug',
      options: {
        source: 'title',
      },
    }),
    defineField({
      name: 'description',
      type: 'text',
    }),
  ],
  preview: {
    select: {
      title: 'title',
      subtitle: 'description'
    }
  }
})

```

### 2. Order Schema:

```

export const orderType = defineType({
  name: "order",
  title: "order",
  type: "document",
  icon: BasketIcon,
  fields: [
    defineField({
      name: "orderNumber",
      title: "Order Number",
      type: "string",
      validation: (Rule) => Rule.required(),
    }),
    defineField({
      name: "stripeCheckoutSessionId",
      title: "Stripe Checkout Session ID",
      type: "string",
      validation: (Rule) => Rule.required(),
    }),
    defineField({
      name: "stripeCustomerId",
      title: "Stripe Customer ID",
      type: "string",
      validation: (Rule) => Rule.required(),
    }),
    defineField({
      name: "clerkUserId",
      title: "Store User ID",
      type: "string",
      validation: (Rule) => Rule.required(),
    }),
    defineField({
      name: "customerName",
      title: "Customer Name",
      type: "string",
      validation: (Rule) => Rule.required(),
    }),
    defineField({
      name: "email",
      title: "Customer Email",
      type: "string",
      validation: (Rule) => Rule.required(),
    })
  ]
}),

export const orderType = defineType({
  fields: [
    defineField({
      name: "stripePaymentIntentId",
      title: "Stripe Payment Intent ID",
      type: "string",
      validation: (Rule) => Rule.required(),
    }),
    defineField({
      name: "products",
      title: "Products",
      type: "array",
      of: [
        defineArrayMember({
          type: "object",
          fields: [
            defineField({
              name: "product",
              title: "Product Bought",
              type: "reference",
              to: { type: "product" },
            }),
            defineField({
              name: "quantity",
              title: "Quantity Purchased",
              type: "number",
            })
          ]
        })
      ],
      preview: {
        select: {
          product: "product.name",
          quantity: "quantity",
          image: "product.image",
          price: "product.price",
          currency: "product.currency",
        },
        prepare(select) {
          return {
            title: `${select.product} x ${select.quantity}`,
            subtitle: `${select.price * select.quantity}`,
            media: select.image,
          };
        }
      }
    })
  ]
}),

defineField({
  name: "totalPrice",
  title: "Total Price",
  type: "number",
  validation: (Rule) => Rule.required().min(0)
}),
defineField({
  name: "currency",
  title: "Currency",
  type: "string",
  validation: (Rule) => Rule.required(),
}),
defineField({
  name: "amountDiscount",
  title: "Amount Discount",
  type: "number",
  validation: (Rule) => Rule.required(),
}),
> defineField({
}),
> defineField({
}),
],
preview: {
  select: {
    name: "customerName",
    amount: "totalPrice",
    currency: "currency",
    orderId: "orderNumber",
    email: "email",
  },
  prepare(select) {
    const orderIdSnippet = `${select.orderId.slice(0,5)}...${select.orderId.slice(-5)}`;
    return {
      title: `${select.name} ${select.orderIdSnippet}`,
      subtitle: `${select.amount} ${select.currency} , ${select.email}`,
      media: BasketIcon
    };
  }
}
}
)
)

```

### 3.Product Schema:

```

export const productType = defineType({
  name: "product",
  title: "product",
  type: "document",
  icon: TrolleyIcon,
  fields: [
    defineField({
      name: "name",
      title: "Product Name",
      type: "string",
      validation: (Rule) => Rule.required()
    }),
    defineField({
      name: "slug",
      title: "Slug",
      type: "slug",
      options: {
        source: "name",
        maxLength: 96
      },
      validation: (Rule) => Rule.required()
    }),
    defineField({
      name: "image",
      title: "Product Image",
      type: "image",
      options: {
        hotspot: true
      },
      validation: (Rule) => Rule.required()
    })
  ],

```

```

    defineField({
      name: "description",
      title: "Description",
      type: "string",
    }),
    defineField({
      name: "price",
      title: "Price",
      type: "number",
      validation: (Rule) => Rule.required().min(0)
    }),
    defineField({
      name: "discount",
      title: "Discount",
      type: "number",
      validation: (Rule) => Rule.required().min(0)
    }),
    defineField({
      name: "categories",
      title: "Categories",
      type: "array",
      of: [{ type: "reference", to: { type: "category" } }]
    }),
    defineField({
      name: "stock",
      title: "Stock",
      type: "number",
      validation: (Rule) => Rule.required().min(0)
    }),
    defineField({
    }),
    defineField({
    })
  ],
  preview: {
    select: {
      title: "name",
      media: "image",
      subtitle: "price"
    },
    prepare(selection) {
      return {
        title: selection.title,
        media: selection.media,
        subtitle: `${selection.subtitle}`
      }
    }
  }
})

```