

Author Name: Uzma Khan

Slot Detail: Friday (morning 9:00am to 12 pm)

Roll No:00485300

Hackathon DAY 2 PLANNING THE TECHNICAL FOUNDATION

Installation:

- * next js
- * typescript
- * tailwind css
- * react Icons
- * lucide Icons
- * shadcn.ui
- * Project Introduction

App Functionality

- * Build Start
- * Shadcn Setup
- * Header
- * Clerk Authentication Setup
- * Footer
- * Sanity CMS Setup
- * Sale Banner Schema
- * Creating Helper Functions
- * Banner
- * Product Schema
- * ProductList
- * Font Configuration
- * ProductCard
- * Single Product Page
- * Search Page
- * CategorySelector
- * Order Schema
- * Zustand Store Setup and Functions dispatch
- * Cart Page
- * Stripe Payment Configuration Setup
- * Checkout Function
- * Success Page
- * Orders Page
- * Webhook Setup and Functionality
- * Orders Component
- * Coupons Setup
- * Shipping
- * Sanity Live Presentation Setup

***** Work flows *****

User land on the homepage.

* Product browsing:

Uses select a category or use the search bar

Product are display with options to filter by price, brand or ratings

*Product details:

User click on a product to view its details

Page includes product description availability and price.

* Add to card:

User add the product to the card

Card update dynamically with quantity and price.

* Checkout:

Uses proceeds to the checkout page.

* Payment:

User enters payment details

After the payment method user will be navigated to success page.

* Order confirmation:

User visit the order history section

* Delivery:

Product deliver to the use's address

***** <Sanity Schema> *****

product Schema:

```
export const productType = defineType({
  name: "product",
  title: "Products",
  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,
    },
}),
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.min(0),
}),
defineField({
    name: "label",
    title: "Label",
    type: "string",
    validation: (Rule) => Rule.min(0),
}),

```

```

defineField({
  name: "status",
  title: "Product Status",
  type: "string",
  options: {
    list: [
      { title: "New", value: "new" },
      { title: "Hot", value: "hot" },
      { title: "Sale", value: "sale" },
    ],
  },
}),
],
preview: {
  select: {
    title: "name",
    media: "image",
    subtitle: "price",
  },
  prepare(selection) {
    return {
      title: selection.title,
      subtitle: `${selection.subtitle}`,
      media: selection.media,
    };
  },
},
});
*****

```

Sales Schema:

```

export const salesType = defineType({
  name: "sale",
  title: "Sale",
  type: "document",
  icon: TagIcon,
  fields: [
    defineField({
      name: "title",
      title: "Sale Title",
      type: "string",
    }),
    defineField({
      name: "description",
      title: "Sale Description",
      type: "text",
    }),
    defineField({

```

```
    name: "badge",
    title: "Discount Badge",
    type: "string",
    description: "Discount Badge Ratio",
  }},
  defineField({
    name: "discountAmount",
    title: "Discount Amount",
    type: "number",
    description: "Amount off in percentage or fixed value",
  }),
  defineField({
    name: "couponCode",
    title: "Coupon Code",
    type: "string",
  }),
  defineField({
    name: "validFrom",
    title: "Valid From",
    type: "datetime",
  }),
  defineField({
    name: "validUntil",
    title: "Valid Until",
    type: "datetime",
  }),
  defineField({
    name: "isActive",
    title: "Is Active",
    type: "boolean",
    description: "Toggle to activate/deactivate the sale",
    initialValue: true,
  }),
  defineField({
    name: "image",
    title: "Product Image",
    type: "image",
    options: {
      hotspot: true,
    },
  }),
],
preview: {
  select: {
    title: "title",
    discountAmount: "discountAmount",
    couponCode: "couponCode",
    isActive: "isActive",
  },
}
```

```
},  
prepare(select) {  
  const { title, discountAmount, couponCode, isActive } = select;  
  const status = isActive ? "Active" : "Inactive";  
  return {  
    title,  
    subtitle: `${discountAmount}% off - Code: ${couponCode} - ${status}`,  
  };  
},  
},  
});
```