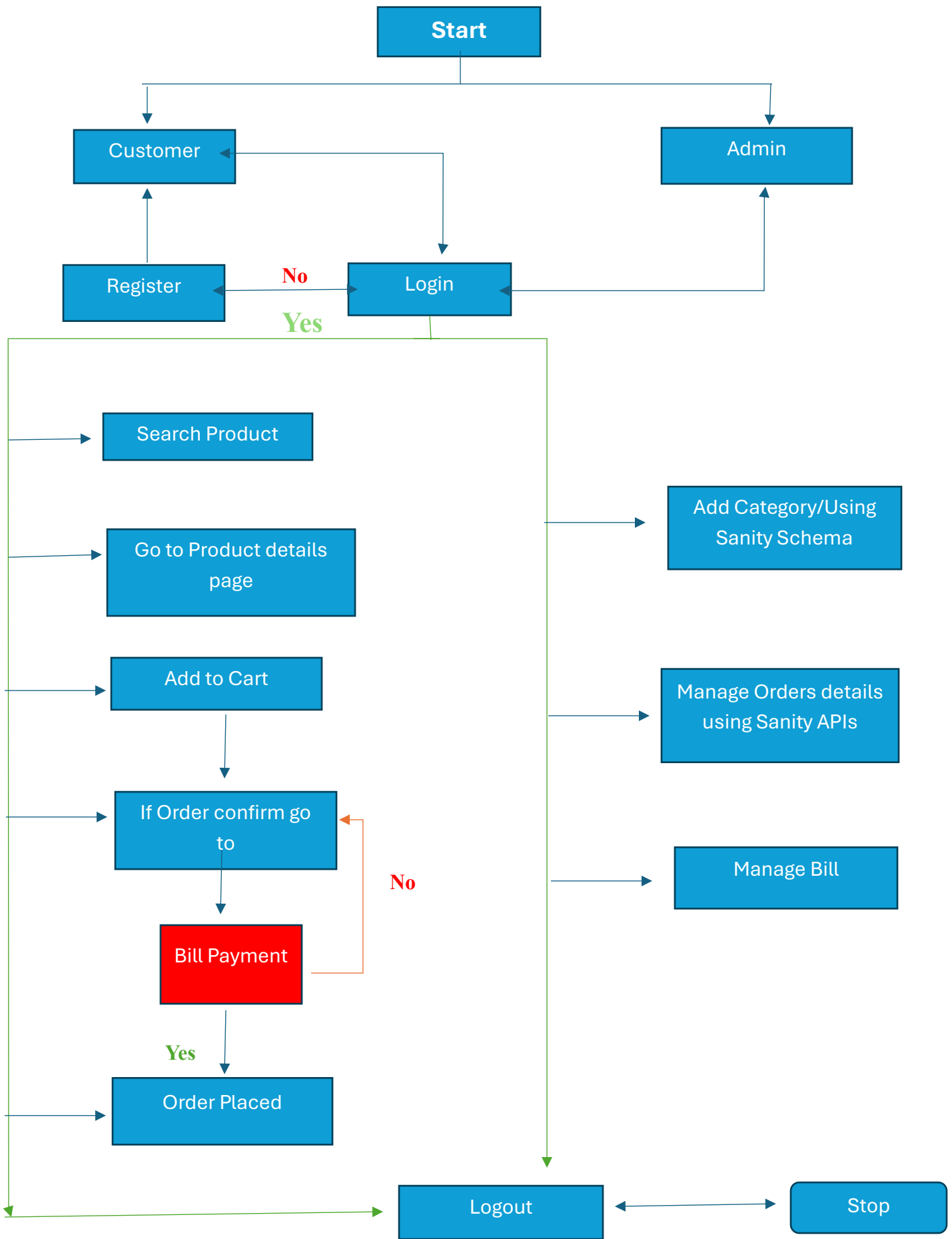
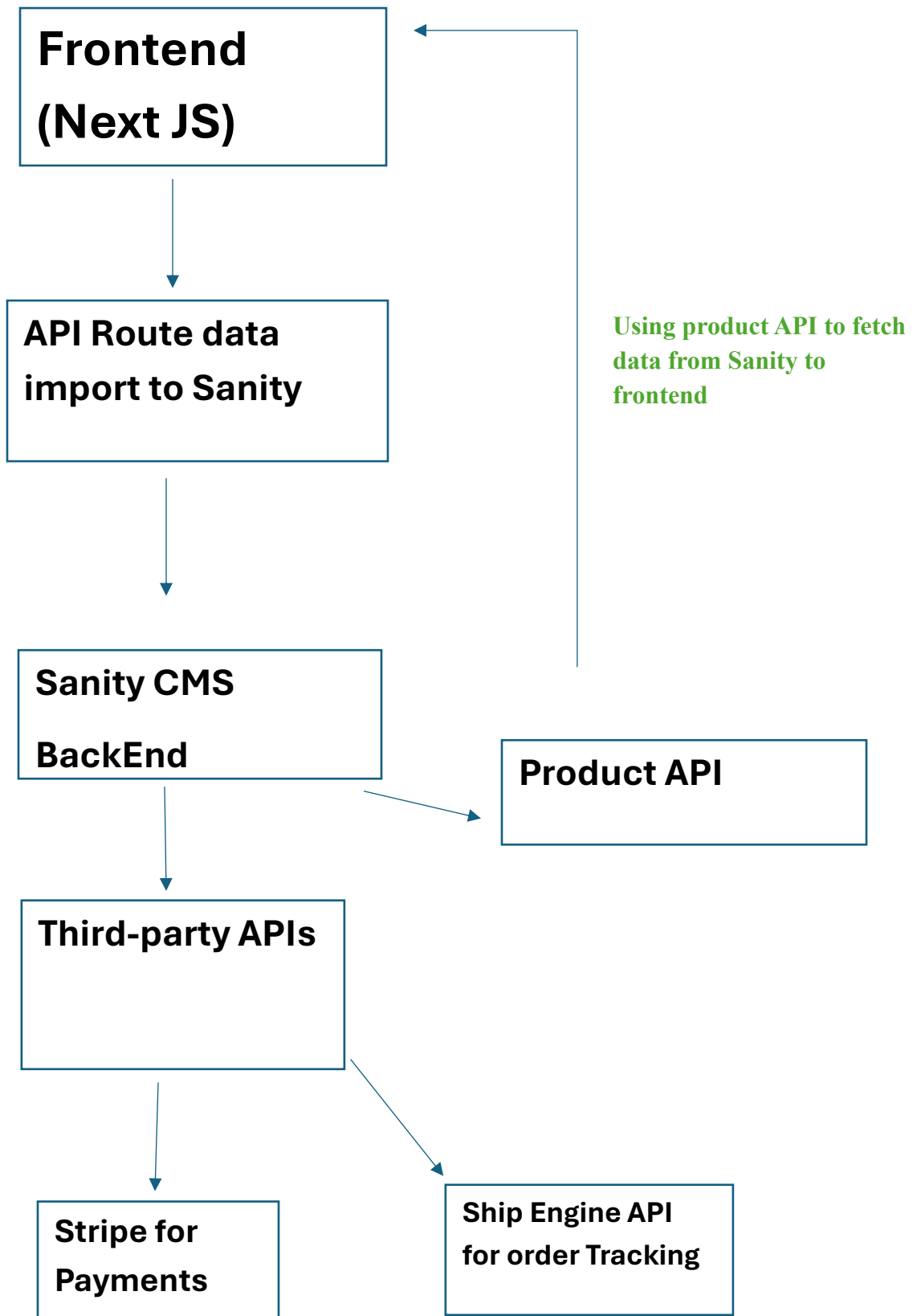


WorkFlow Diagram for E-Commerce Website Avion

Created By Zeeshan Tariq



System Architecture Overview Created By Zeeshan Tariq



Data Schema Design.js

01) Schema Types used in Avion Marketplace Hackathon-2025

Product Schema.

```
export const product = defineType({
  name: "product",
  title: "Products",
  type: "document",
  icon: TrolleyIcon,
  fields: [
    defineField({
      name: "category",
      title: "Category",
      type: "reference",
      to: {
        type: "category"
      }
    }),
    defineField({
      name: "name",
      title: "Title",
      validation: (rule) => rule.required(),
      type: "string"
    }),
    defineField({
      name: "slug",
      title: "Slug",
      validation: (rule) => rule.required(),
      type: "slug"
    }),
    defineField({
      name: "image",
      type: "image",
      validation: (rule) => rule.required(),
      title: "Product Image"
    }),
    defineField({
      name: "price",
      type: "number",
```

```

        validation: (rule) => rule.required(),
        title: "Price",
    })),
    defineField({
        name: "quantity",
        title: "Quantity",
        type: "number",
        validation: (rule) => rule.min(0),
    })),
    defineField({
        name: "tags",
        type: "array",
        title: "Tags",
        of: [{
            type: "string"
        }]
    })),
    defineField({
        name: 'description',
        title: 'Description',
        type: 'text',
        description: 'Detailed description of the product',
    })),
    defineField({
        name: 'features',
        title: 'Features',
        type: 'array',
        of: [{ type: 'string' }],
        description: 'List of key features of the product',
    })),
    defineField({
        name: 'dimensions',
        title: 'Dimensions',
        type: 'object',
        fields: [
            { name: 'height', title: 'Height', type: 'string' },
            { name: 'width', title: 'Width', type: 'string' },
            { name: 'depth', title: 'Depth', type: 'string' },
        ],
        description: 'Dimensions of the product',
    })),
],
preview: {

```

```

    select: {
      title: 'name',
      media: 'image',
      price: 'price',
    },
    prepare(select) {
      return {
        title: select.title,
        subtitle: `$$${select.price}`,
        media: select.media,
      };
    },
  },
}
})

```

02. Order Type 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",
    }),
    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(),
  }),
  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: "total",
    title: "Total",
    type: "number",
    validation: (Rule) => Rule.required(),
}),
defineField({
    name: "currency",
    title: "Currency",
    type: "string",
    validation: (Rule) => Rule.required(),
}),
defineField({
    name: "amountDiscount",
    title: "Amount Discount",
    type: "number",
    validation: (Rule) => Rule.min(0),
}),
defineField ({
    name: "status",
    title: "Order Status",
    type: "string",
    options: {
        list: [

```

```

        {title:"Pending", value:"pending"},
        {title:"Paid", value:"paid"},
        {title:"Shipped", value:"shipped"},
        {title:"Delivered", value:"delivered"},
        {title:"Cancelled", value:"cancelled"},
    ],
    },
  )),
  defineField({
    name:"orderDate",
    title:"Order Date",
    type:"datetime",
    validation: (Rule) => Rule.required(),
  )),
],
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} (${orderIdSnippet})`,
      subtitle: `${select.amount} ${select.currency}, ${select.email}`,
      media: BasketIcon,
    };
  }
},
})

```


03. Categories Schema Type

```
export const Category = defineType({
  name: "category",
  title: "Category",
  type: "document",
  icon: TagIcon,
  fields:[
    defineField({
      name: "name",
      title: "Name",
      type: "string",
      validation: (rule) => rule.required(),
    }),
    defineField({
      name: "slug",
      title: "Slug",
      type: "slug",
      validation: (rule) => rule.required(),
      options: {
        source: "name",
      }
    })
  ]
})
```