NAME: RUBA HAROON

ROLL NO: 00441882

SLOT: FRIDAY 9 TO 12

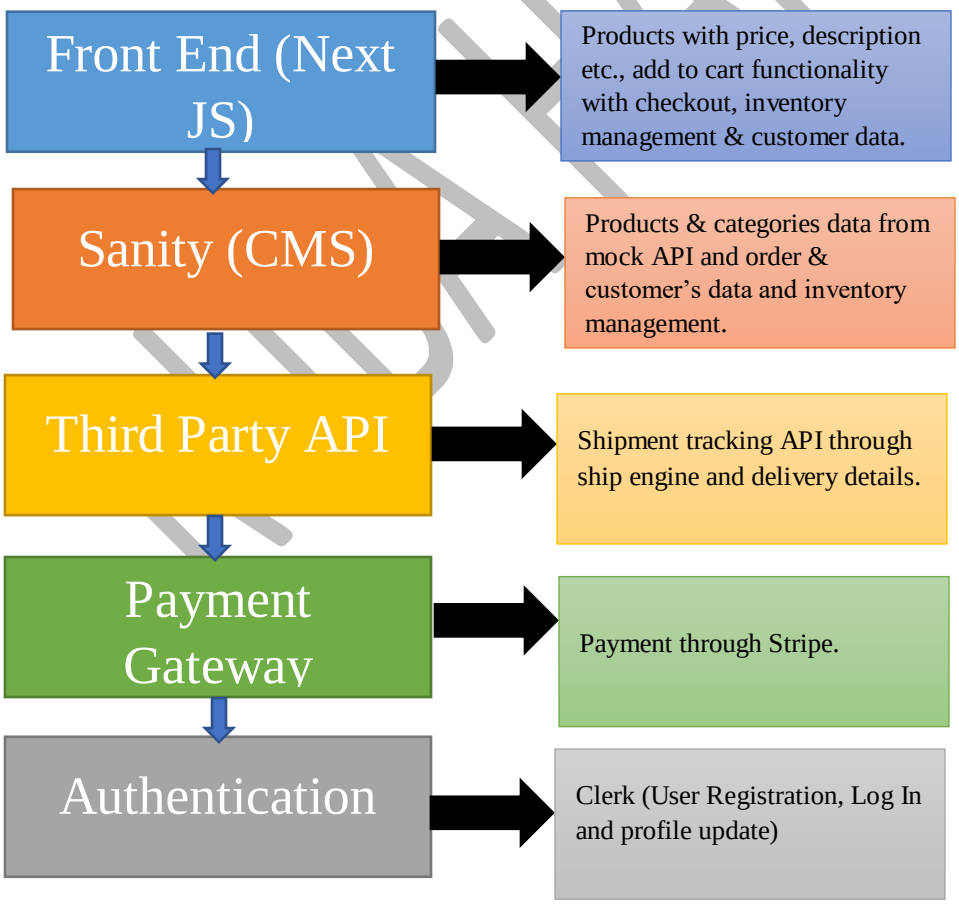
TEACHER: SIR HAMZAH SYED

AVION ECOMMERCE PLATFORM:

OBJECTIVE:

The primary purpose of Avion is to provide a convenient and user-friendly platform where customers can discover, customize Chairs, Sofa, Vase sets and Lamps in very affordable prices. This platform builds a trustworthy brand through delivering high quality products and excellent customer support and ensures a smooth shopping journey from browsing to checkout to delivery.

1: SYSTEM ARCHITECTURE OVERVIEW:



2) Components Description:

- **Frontend (Next.js)**:

  - ✓ Display the responsive and attractive UI for browsing products of Chairs, Sofa, Vase Sets and Lamps. Managing cart through use state, inventory and orders. This is made from Next-JS, which makes the process super-fast and easy to use. Styling is done through Tailwind CSS, which gives your website looks stylish, professional, attractive and responsive for every screen.
  - ✓ Fetches and displays data from the backend APIs in real-time.
  - ✓ Contact Form through ZOD library and Shad CN UI.

- **Backend (Sanity CMS)**:

- ✓ Act as a primary backend to manage product data such as listing of products, details and inventory stats, customer details, and inventory handling and user data. This process makes the data up to date.
- ✓ Provide API for the front end in order to fetch and update data.

- **Third-Party APIs**:

- ✓ Integrate services like shipment tracking through Ship Engine. This API keeps track of your order and gives updates about inventory status.
- ✓ Payment process through stripe. It make sure that your transaction is safe and secure. It gives details related to the confirmation of your order placement.

- **Authentication (Clerk)**:

- ✓ Handles User registration, Log In and admin dashboard, which make sure that only you can access your account.
- ✓ Integrates with Sanity CMS to store user data securely.

- **Hosting & Deployment: (Vercel)**

- ✓ For hosting use Vercel platform as it provides automatic deployments and better performance.

3) I) Workflow related to UI:

- **Home Page:**

- Displays a responsive and professional Header, featuring New Ceramics products, popular products, newsletter section and an attractive Footer.
- A hero section featuring a chair image with some inspiring text related to my platform with a view collection button.

- **About Page:**

- Displays about platform that what services offer what is the company about, what is the mission and vision of Avion**.**

- **Product Page:**
- Displays products in a grid format with sorting and filtering options for a better user experience.

- **Product Details Page:**

- Displays product details, description, quantity update options, an add-to-cart button, color and size selection and dimensions with height, width.

- **Cart Page:**

- Shows cart items, quantity, subtotal, and total price with checkout button.

- **Checkout Page:**

- Provides delivery details and payment options.

- **Order Confirmation Page:**

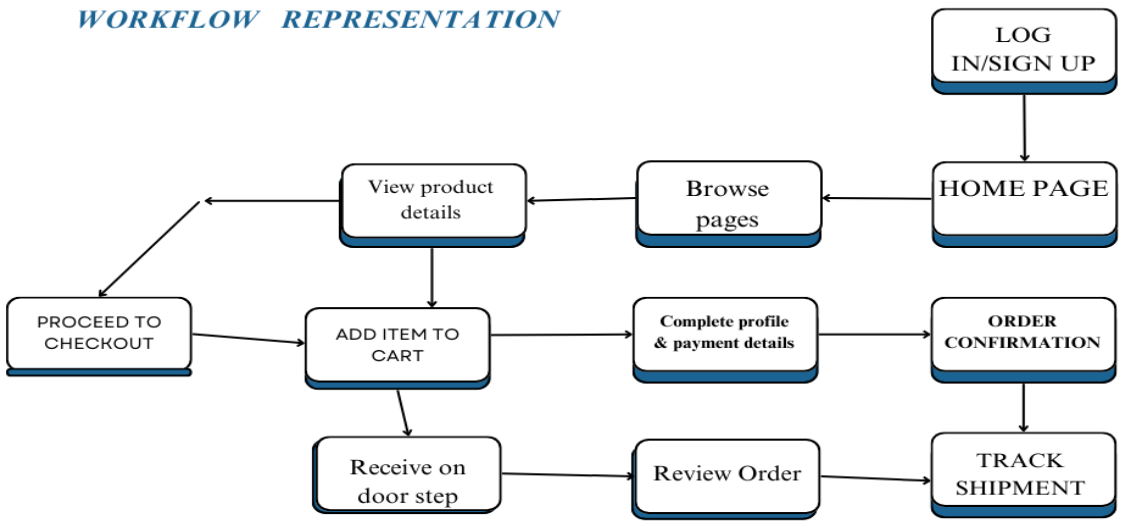
- Displays order ID and delivery details.

- **Contact Page:**

- Display a contact page with customer email, phone number, address and message.

- Validation through ZOD library which is best for website as sometimes customer fills incomplete information.

3) 2) Workflow:



1) User Registration:

- User signs up through clerk and make their account and can update their profile.
- Registration details data should be saved in sanity.

2) Product Browsing:

- User go in categories and can browse each category and when they click on each category they see the chairs, sofa, and vase set and lamps products.
- Through Sanity the products description, dimension, images, size and color data fetch.
- Dynamic product listings are displayed on the frontend in which there is product id of every product through which product details can display on website.

3) Order Placement:

- User adds products to the cart and then proceeds to checkout.
- In checkout there is a billing information form in which customer details should be provided.
- Order details includes products, quantities, size, color and shipping details are saved in Sanity.
- Payment is processed through stripe and then a confirmation dialog pops on customer screen and the details of their order can send them on the user email and order data is saved in sanity.

4) Shipment Tracking:

- After placement of order, shipment details is saved in ship engine.
- After data saved in ship engine then tracking information display on customer screens.

5) Inventory Management:

- Product stock details can be managed through sanity.
- .Users can see the stock availability on screen which was fetch through sanity in front end.
- Out of stock products can be saved in wish list and the popup appears on user screen that product is out of stock.
- In stock products are added to cart and then the user can proceed to checkout.

6) API Endpoints:

Endpoint	Method	Response Example
/products	GET	[{ "name": "Product Name", "slug": "product-slug", "price": 200 }]
/order	POST	{ "order Id": 123, "status": "success" }
/shipment-tracking	GET	{ "tracking Id": "AB123", "status": "In Transit" }
/delivery-status	GET	{ "order Id": 456, "delivery Time": "30 minutes" }
/inventory	GET	{ "product Id": 789, "stock": 50 }
/cart	POST	{“cart Id": 101, "items": [...]}
/wish list	POST	{“wish list Id": 202, "items": [...]}

7) Sanity Schema Example

Product.ts:

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

export const product = defineType({
  name: "product",
  title: "Product",
  type: "document",
  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",
    }),
    defineField({
      name: "stock",
      title: "Stock",
      type: "number",
      validation: (rule) => rule.required(),
    }),
    defineField({
      name: "inStock",
      title: "In Stock",
      type: "boolean",
      validation: (rule) => rule.required(),
    }),
    defineField({
      name: "price_id",
      title: "Stripe Price ID",
      type: "string",
    }),
  ],

  preview: {
    select: {
      title: "name",
      media: "image",
      subtitle: "price",
      inStock: "inStock",
      stock: "stock",
    },
    prepare(selection) {
      const { title, subtitle, media, inStock, stock } = selection;
      return {
        title,
        subtitle: `${subtitle} | ${inStock ? `In Stock (${stock})` : "Out of
Stock"}`,
        media,
```

```
      };
    },
  },
});
```

8) Technical Roadmap:

This document outlines the technical roadmap for the Avion. It covers the development, testing, and launch phases, along with key features and workflows.

❖ Development Phase

Authentication

- Implement user registration and login using **Clerk**.
- Integrate Clerk with **Sanity CMS** for user data storage.

Product Management

- Create mock API for product data.
- Store product data in **Sanity CMS**.
- Fetch and display product data on dynamic frontend pages.

Cart and Wish list

- Implement add-to-cart functionality with real-time stock checks.
- Allow out-of-stock products to be added to the wish list.
- Display total bill and a "Proceed to Checkout" button on the cart page.

Payment Integration

- Integrate **Stripe** for secure payments.
- Use Stripe test account for development.
- Handle payment success and failure scenarios.

Shipment Tracking

- Integrate **Ship Engine** for shipment tracking.
- Generate tracking numbers and display them on the frontend.
- Allow users to track their orders in real-time.

Inventory Management

- Create API for real-time stock updates in **Sanity CMS**.
- Update stock levels upon order placement.
- Prevent out-of-stock products from being added to the cart.

❖ Testing Phase

End-to-End Testing

- Test all workflows, including:
  - ✓ User registration.
  - ✓ Product browsing.
  - ✓ Cart management.
  - ✓ Checkout process.
  - ✓ Shipment tracking.
- Validate API responses and ensure data accuracy.
- Security Audits
- Conduct security audits for sensitive data handling, including:
  - ✓ User authentication.
  - ✓ Payment processing.

❖ Launch Phase

➢ Deployment

- Deploy the platform on a cloud hosting service (e.g., **Vercel**, **Netlify**).
- Monitor user feedback and optimize for performance.

❖ Post-Launch

- Collect user feedback for continuous improvement.
- Optimize API performance and frontend loading times.

- Scale infrastructure based on traffic and demand.

Conclusion

This technical foundation outlines the architecture, workflows, and API endpoints for the **Avion Platform**. The platform will provide a seamless e-commerce experience with:

- Robust authentication.
- Efficient inventory management.
- Real-time shipment tracking.