Marketplace Technical Foundation – VogueAura

Workflow Overview:

Frontend Setup:

- Build a modern, scalable frontend using Next.js for server-side rendering and enhanced SEO.
- Implement Tailwind CSS for consistent styling and full responsiveness across devices.
- Ensure cross-browser compatibility and optimized loading speeds.

Data Management:

- Configure Sanity CMS for flexible content management, focusing on:
  - Product catalog (ID, name, price, stock, categories, images).
  - Customer information (name, email, address).
  - Order history (products purchased, timestamps, payment status).
  - Shipping data and reviews with customizable fields for expansion.

Product Browsing Workflow:

- User visits the homepage and can browse products by category.
- Product data fetched dynamically from Sanity CMS via API.
- Selected product details, including multiple images, description, pricing, availability, and customer reviews, are displayed on the frontend using Sanity API responses.

Order Placement Workflow:

- User adds items to the cart with a dynamic cart summary.
- If the user is logged In , then he can place his order , if not then the order will failed
- User proceeds to a secure checkout page and submits the order.
- Order data, including product details, customer information, and payment confirmation, is stored in Sanity CMS.
- Payments processed through multiple gateways:
  - **International Market:** Payoneer, Wise, Stripe.
  - **Local Market:** All major banks, JazzCash, NayaPay.

Shipment Tracking Workflow:

- Upon order confirmation, a unique shipment ID is generated and stored.
- Real-time shipment tracking data is fetched via a third-party shipment API.

- Users can view shipment progress directly on their order summary page.

API Endpoints:

- `/products` (GET): Fetches all products with filters for categories and availability.
- `/orders` (POST): Creates a new order and triggers payment processing.
- `/shipment` (GET): Retrieves real-time shipment status based on shipment ID.

Sanity Schema Definition:

```
export default {
    name: 'product',
    type: 'document',
    fields: [
      { name: 'Id', type: 'number', title: 'Product Id' },
      { name: 'name', type: 'string', title: 'Product Name' },
      { name: 'description', type: 'text', title: 'Description' },
      { name: 'price', type: 'number', title: 'Price' },
      { name: 'category', type: 'string', title: 'Category' },
      { name: 'color', type: 'array', title: 'Product color' },
      { name: 'stock', type: 'number', title: 'Stock Level' },
      { name: 'images', type: 'array', of: [{ type: 'image' }],
      title: 'Product Images' },
      { name: 'reviews', type: 'array', of: [{type: 'reference',
      to: [{type:   'review'}]}]}]
      ]
      };
```

8. **Key Outcome:**
   - Complete, detailed technical documentation with system architecture diagrams and API specifications.
   - Sanity CMS schema designed for long-term scalability and easy content management.
   - Clear and structured API endpoint definitions for seamless frontend-backend communication.
   - Documented peer feedback, version control strategies, and continuous improvement processes.