Day 2: Technical Planning Outcome Checklist

Objective: Transition to the technical foundation for your marketplace.

Checklist:

1. Technical Plan:

Frontend:

- **Framework**: Next.js (React-based for fast rendering and SEO optimization).
- **Styling:** Tailwind CSS for a responsive and modern UI.
- **State Management:** React Context API for managing global state.
- **Deployment:** Vercel for seamless deployment with GitHub integration.
- **Authentication:** JWT (JSON Web Tokens) for user authentication.

Backend:

- **Database:** Sanity for content management (product listings, customer details, etc.).
- **API Routes:** Next.js API routes for server-side functionality (handling orders, products, etc.).
- **Payment Integration**: EasyPaisa and JazzCash APIs for payment processing.

2. Workflows:

Customer Workflow:

- Browse products by category or search.
- Add products to cart.
- Checkout with payment methods (COD, EasyPaisa, etc.).
- Receive order confirmation with delivery details.
- Track order status via shipment API.

Admin Workflow:

- Login to the admin panel for product and order management.
- Add/edit/remove products from the marketplace.

- Process orders (mark as shipped, manage payments).
- Monitor shipment status and update customers.

3. API Requirements:

Products API:

- `GET /api/products`: Fetch all products.
- `POST /api/products`: Add new products.
- `PUT /api/products/{id}`: Update product details.
- `DELETE /api/products/{id}`: Delete a product.

Orders API:

- `GET /api/orders:` Fetch all orders.
- `POST /api/orders:` Place a new order.
- `PUT /api/orders/{id}:` Update order status (shipped, delivered).

Customer API:

- `POST /api/customers/register`: Register new customer.
- `POST /api/customers/login`: Authenticate user for login.

Payment API:

- Integration with EasyPaisa, JazzCash, and COD payment methods.

Shipment API:

- `GET /api/shipment/{orderId}`: Fetch shipment status for an order.
- `POST /api/shipment`: Create a new shipment for an order.

4. Sanity Schema:

```
// schemas/product.js
export default {
  name: 'product',
  type: 'document',
  title: 'Product',
  fields: [
    { name: 'name', type: 'string' },
    { name: 'price', type: 'number' },
    { name: 'stock', type: 'number' },
    { name: 'category', type: 'string' },
    { name: 'material', type: 'string' },
    { name: 'image', type: 'image' }
  ]
}

// schemas/order.js
export default {
  name: 'order',
  type: 'document',
  title: 'Order',
  fields: [
    { name: 'customerId', type: 'reference', to: [{ type: 'customer'
}] },
    { name: 'products', type: 'array', of: [{ type: 'reference', to:
[{ type: 'product' }] }] },
    { name: 'total', type: 'number' },
    { name: 'status', type: 'string', options: { list: ['Pending',
'Shipped', 'Delivered'] } }
  ]
}

// schemas/customer.js
export default {
  name: 'customer',
  type: 'document',
  title: 'Customer',
  fields: [
    { name: 'name', type: 'string' },
    { name: 'phone', type: 'string' },
    { name: 'address', type: 'string' },
    { name: 'email', type: 'string' }
  ]
}

// schemas/shipment.js
export default {
  name: 'shipment',
  type: 'document',
  title: 'Shipment',
  fields: [
    { name: 'orderId', type: 'reference', to: [{ type: 'order' }] },
    { name: 'trackingNumber', type: 'string' },
    { name: 'carrier', type: 'string' },
    { name: 'status', type: 'string', options: { list: ['In Transit', 'Out for Delivery', 'Delivered'] } },
    { name: 'estimatedDelivery', type: 'datetime' }
  ]
}
```

5. Collaboration Notes:

Collaborated with team leader Shan–e–ali on the whole project for rough briefing (notes are not included)

6. Submission from Day 2:

Click here to see the day 2 work.