

①

Architecture Of Market place

Builder Hackathon - 2025

{ GIAIC Student: "Transformation of Karachi into"
E-commerce market "

DAY-2:

16-1-2025

"Planning The Technical Foundation"

Goal:

~~Transition~~

Completing of Day 1 foundation
from transition to technical
Planning

① Technical Requirement of E-commerce

1. Core Features:

1. Frontend Requirement / User Interface
 - Responsive and intuitive design for web and mobile
 - Personalized recommendation
 - Multi language and currency support

(2)

2. Product Management:

- Admin dashboard for adding, editing and removing products.
- Categories, tags and filter for easy navigation
- Real time updates

3. Search and Filtering:

- Advance Search bar with auto suggestions.
- Filters based on price, categories and deals.

4. User Accounts:

- Registration via email or phone number for regular customer or guest checkout.
- Saved preferences and purchase history
- Address book for delivery

5. Order and payment system:

- Shopping cart and wishlist functionally

- Integration with multiple payment gateways, e.g. Sadap, Debit or Credit Card.
 - Secure check out.
6. Shipping and Delivery:
- Integration with logistics providers for real-time tracking
 - Shipping cost calculation based on location
7. Customer Support:
- Chatbots and live support
 - FAQs and help lines.
8. Analytics and Reporting:
- Sales reports and user behavior analysis
 - Conversion rate tracking.

(4)

2. Technical Stack:

1. Front-End:

- Next.js Framework
- Tailwind CSS styling
- React, shadcn and Framer libraries.
- Next.js Hooks.

2. Back-End:

- Sanity CMS:
 - * For create and manage schema data
 - * GET, POST, Delete method use.
 - * Fetching API
 - * GROQ Queries.
- TypeScript function

3. Third Party API:

- For shipping and tracking
- For payment gateway

4. Deployment:

- Vercel through Git Hub

B. Design System ~~Architect~~ Architecture

1. User initiates a search.

- Main page .tsx
(Landing Page)

- About / Page .tsx

- Product / Page .tsx

2. Product catalog retrieves results

- Product detail .tsx

- View detail .tsx

3. User filters by tag and categories

4. Add item to cart

- cart .tsx

5. Proceeds to checkout

- checkout .ts

6. System validates payment

- Payment .ts

7. Order and get status /
delivery schedule.

- Order .tsx

- Status .tsx

(6)

3. Plan API Requirement:

API End point:

1. Endpoint Name / Product.

- Method: GET

- Description: Fetch all available products from Sanit

- Response:

Product detail

{ - id, name, price, image, }
description

2. Endpoint Name / Order

- Method: POST

- Description: Create a new order from Sanit

- ~~Response~~ detail:

{ - id, customer name, customer info Product id }

- Payload:

{ id, customer info, product detail payment status }

3. Endpoint Name / shipment

- Method: GET

- Description: Track order

⑦

status via third-party
API

• Response:

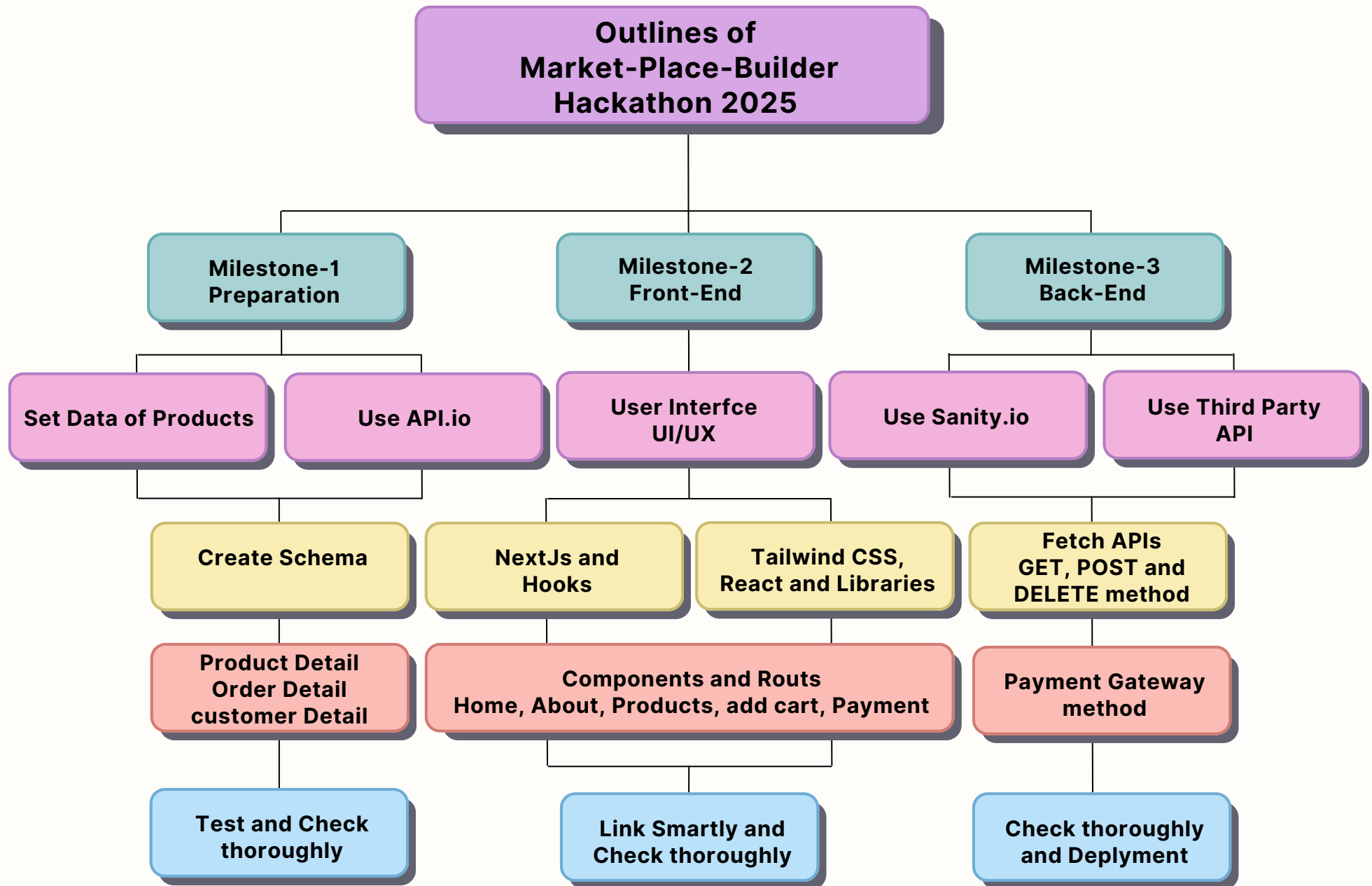
{shipment id, order id
status, expected delivery
time.}

REQUIRED INSTALLATION

- npx create-next-app@latest
- npm create-sanity@latest
- npm shadcn@latest init
- npm i react
- npm i react-icons
- npm i framer
- npm i framer-motion

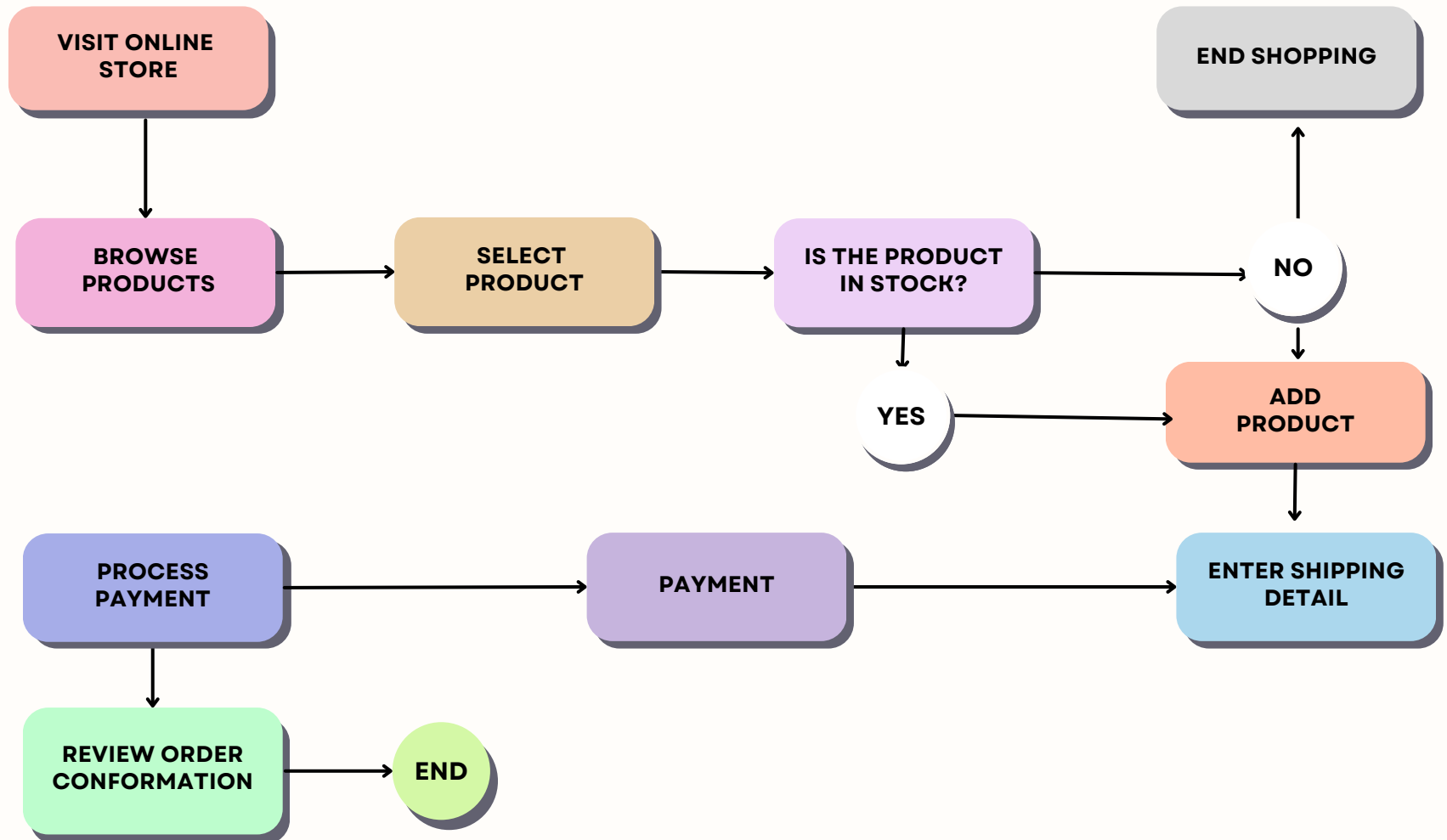
— x —

ROAD MAP CHART



User Workflow Diagram

ONLINE SHOPPING



WORK FLOW CHART

