Projects Plan Of Hackathons:-

Q1) Choose your Marketplace Type:-

Ans) General is Quick commerce (q-commerce) is a marketplace model that focuses on fast delivery of goods, typically within hours or less, catering to customer needs for convenience and immediacy. It usually involves quantities of groceries, food or daily essentials delivered from local resturrent.

Q2) Define your business Goals?

Ans) Business Goals:- To provide fast, reliable food delivery from local resturrent for groceries or meal kits, ensuring convenience quality and costumer, satisfaction with deliveries in under an hours-

§ Target audiences:-

Ans) Resturrent delivery (bsy, Professionals, families Student, friend surban residents for looking meals.

2) Grocery delivery (Parents health conscious individuals costomers interested in healthly diet-specific, or organic food options.

3) Mealkits for delivery (busy, Professional this looking for quick meal delivery options 10. food enthusiasts.

To Create a DATA SCHEMA:-

Products
- ProductID
- Product Name
- Product description
- Product Stock
- Product Price

Order
- Order ID
- Customer ID
- Date
- Status
- Total Amount
- Payment method
- Delivery Status

Customers
- Customer ID
- Name
- email
- Phone
- Address

Payement
- Payement ID
- Order ID
- Amount
- Status

Resturrent
- Name
- Address
- menu
- Rocting

Delivery
- delivery ID
- order ID
- delivery status
- delivery Time
- Driver ID

ExPLAINATIONs:-

1) A customers can have many orders.
2) An order can have many products.
3) Each order is linked to one Resturrent
4) Each order is tied to one Payement.
5) Each order has one delivery.

6) Deliveries are assigned to Delivery zones with driver responsible for the delivery.

$\downarrow$
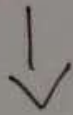
Sanity CMS

- create Schemas for food Product and orders.

// Product Schemas //

```
{
"ProductID": "string",
"name": "string",
"description": "string",
"price": "string",
"category": "string",
"Image": "string",
"Sizes": "~~array~~" => "[small, Large]",
"Stock": "number",
}
```

// Order Schemas //

```
{
"orderID": "string",
"customerID": "string",
"customer name: "string",
"customer contact: "string",
"items": [
{ "Product ID": "string",
```

$\downarrow$

Build user Authentication
- Implement registration and login
  Using Secure in Next Authentication.js

$\downarrow$

Animation
. to create animation full
website like Add to card Login
  a lot of

$\downarrow$

install MongoDB Database
. Set up MongoDB Atlas or Local MongoDB
  . connect using Database.

$\downarrow$

continue to Sanity cms
    Next Page full Schemas

Day 16

HACKAThON Day 2 Assignment

Documentss :-

```
┌─────────────────────────────┐
│ Foodstuck                   │
│ Q-commerce food Delivery    │
└─────────────────────────────┘
              │
              ↓
      ┌──────────────────┐
      │ Start Project    │
      │ Initialization   │
      └──────────────────┘
              │
              ↓
┌──────────────────────────────┐
│    Install Dependencies      │
│  • Tailwind css              │
│  • Next Auth.js              │
│  • Animation - css           │
│  • Sanity client Library     │
│  • ShipEngine API            │
│  • mongoose                  │
│  • Axios                     │
│  • Icons                     │
└──────────────────────────────┘
              │
              ↓
┌──────────────────────────────────┐
│ Tailwind css Config              │
│ • intall Tailwindcss in Next.js  │
│ Project define responsive and    │
│   color                          │
└──────────────────────────────────┘
```

```
    "quantity": "number",
    "sizes": "string",

},
{
    "Total Amount": "number",
    "payement status": "string",
    "shipping status": "string",
    "creatcalender": "Date",
},
],
```

↓

Develop Homepage
• Display featured food products
and categories > Includes search
functionality to filter products
based on categories, Ingredients
and price.

↓

Create Product Page

Shows detail information for each food items like includes Product name, Price, description, Stock Status and high-Quality Images. for Example

- " Total Burgers: 5 "
- After purchase of 1 burger
- Remaning Burgers: 4
- Update Stock in database

↓
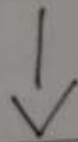
implement Cart functionality

- Users can add, remove, and modify items
- The cart presists user selections until checkout, stock management- when a pizza is purchased, reduce stock by 1 Notify user of remaining stock update

↓
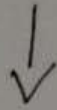
Cheakout Process

- Collect user information and payement details.
- User authorize, shipping address, order confirmation -

↓
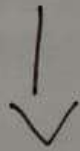
Implement order History

• customers can view their past order
• includes products purchased, order total and status -
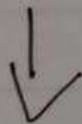
↓

future improvements

• User Reviews and Ratings: Allow customers to rate and review products, Enhanced search functionality: Implement full-text-search with filtering by Price range, and customer ratings, Mobile App ratings: Develop a mobile app with React Native for a better on the go Shopping experience

↓
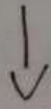
API Endpoints

- Get /api/Products: Retrieve a list of products

- Get /api/products: Retrieve product Details

- Get /api/products/id/: update product stock - Decrease stock when a purchase is made

- Post /api/orders: Place an order.

- Get /api/orders/id: Retrieve order details

- Get/api/users/id: Retrieve user Profile info

↓

End Project