

# Hackathon Day 2

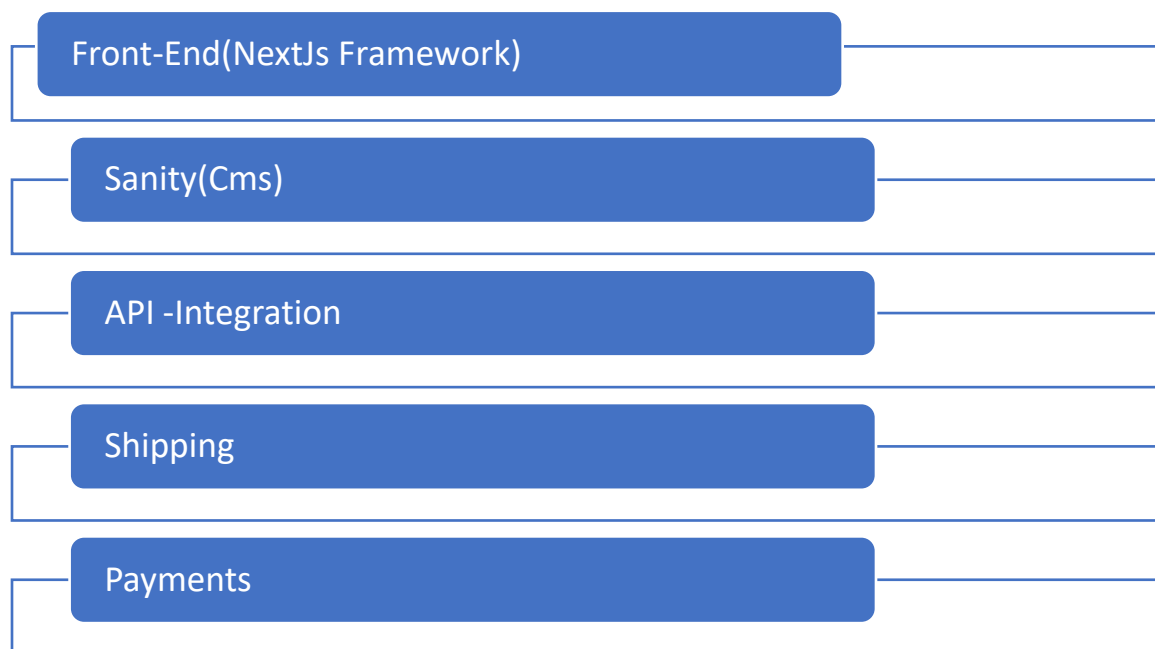
## Technical Requirements:

1.Front-End .

2.Sanity as CMS

3.Integrating APIs      3a. ship Engine APIs   3b. Payment gateways API

Below is a schematic flow diagram



## **Front-end Workflow**

### **1. Browsing Products:**

- Users visit the site and browse through various products.
- They click on any product to visit its detail page, which is powered by Sanity as the backend CMS for updating product details.

### **2. Adding to Cart:**

- Users add their desired products to the cart.

### 3. Payment Methods:

- Users proceed to the payment page where they can choose from various payment methods, including Cash on Delivery (COD), credit cards, and online platforms like PayPal.

### 4. Shipment:

- After the payment, the shipping process is handled by ShipEngine, which tracks the delivery from the warehouse to the customer's address.

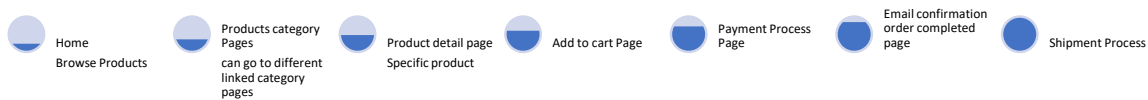
### 5. Confirmation Emails:

- Once the order is confirmed, an email is sent to both the admin and the customer to notify them of the order status.

### 6. Order Delivery Confirmation:

- When the order reaches the customer, a confirmation email is sent to both the admin and the customer to confirm the deliver.

## Visual Diagram of Work flow:



## API structure:

End Points	Methods	Description	Parameters	Response Example
/api/products	GET	Fetch All products	none	
/api/products/id	GET	Fetch Single product	Path id	{Id: ,name:}
/api/foods	POST	Add a new item	Name ,price ,category	{success: true, id:5}
/api/foods/id	PUT	Update an item	Id, name, price	{success: true, id:5}
/api/foods/id	DELETE	Delete an item	id	{success: true}

/api/categories	GET	Fetch all products	none	{categories: ["Dresses"]}
-----------------	-----	--------------------	------	------------------------------

## Schemas For a Website:

### Product schema:

```

export default {
  name: 'product',
  title: 'Product',
  type: 'document',
  fields: [
    {
      name: 'name',
      title: 'Product Name',
      type: 'string',
      validation: Rule => Rule.required(),
    },
    {
      name: 'description',
      title: 'Description',
      type: 'text',
    },
    {
      name: 'category',
      title: 'Category',
      type: 'reference',
      to: [{ type: 'category' }],
    },
    {
      name: 'price',

```

```
    title: 'Price',
    type: 'number',
    validation: Rule => Rule.required().min(0),
  },
  {
    name: 'stock',
    title: 'Stock',
    type: 'number',
    validation: Rule => Rule.required().min(0),
  },
  {
    name: 'image',
    title: 'Image',
    type: 'image',
    options: {
      hotspot: true,
    },
  },
],
};
```

**Category schema:**

```
export default {
  name: 'category',
  title: 'Category',
  type: 'document',
  fields: [
    {
      name: 'name',
      title: 'Category Name',
```

```
    type: 'string',
    validation: Rule => Rule.required(),
  },
  {
    name: 'description',
    title: 'Description',
    type: 'text',
  },
],
};
```

#### **User schema:**

```
export default {
  name: 'user',
  title: 'User',
  type: 'document',
  fields: [
    {
      name: 'username',
      title: 'Username',
      type: 'string',
      validation: Rule => Rule.required(),
    },
    {
      name: 'email',
      title: 'Email',
      type: 'string',
      validation: Rule => Rule.required(),
    },
  ],
}
```

```
    name: 'passwordHash',
    title: 'Password Hash',
    type: 'string',
    hidden: true,
  },
  {
    name: 'address',
    title: 'Address',
    type: 'text',
  },
  {
    name: 'phoneNumber',
    title: 'Phone Number',
    type: 'string',
  },
  {
    name: 'userRole',
    title: 'User Role',
    type: 'string',
    options: {
      list: [
        { title: 'Customer', value: 'customer' },
        { title: 'Admin', value: 'admin' },
      ],
    },
  },
],
};
```

**Orde schema:**

```
export default {  
  name: 'order',  
  title: 'Order',  
  type: 'document',  
  fields: [  
    {  
      name: 'user',  
      title: 'User',  
      type: 'reference',  
      to: [{ type: 'user' }],  
      validation: Rule => Rule.required(),  
    },  
    {  
      name: 'totalAmount',  
      title: 'Total Amount',  
      type: 'number',  
      validation: Rule => Rule.required(),  
    },  
    {  
      name: 'paymentMethod',  
      title: 'Payment Method',  
      type: 'string',  
      options: {  
        list: [  
          { title: 'Cash on Delivery', value: 'COD' },  
          { title: 'Credit Card', value: 'CreditCard' },  
          { title: 'PayPal', value: 'PayPal' },  
        ],  
      },  
    },  
  ],  
}
```

```
    },
  },
  {
    name: 'shippingAddress',
    title: 'Shipping Address',
    type: 'text',
    validation: Rule => Rule.required(),
  },
  {
    name: 'orderStatus',
    title: 'Order Status',
    type: 'string',
    options: {
      list: [
        { title: 'Pending', value: 'Pending' },
        { title: 'Shipped', value: 'Shipped' },
        { title: 'Delivered', value: 'Delivered' },
        { title: 'Cancelled', value: 'Cancelled' },
      ],
    },
    initialValue: 'Pending',
  },
  {
    name: 'orderItems',
    title: 'Order Items',
    type: 'array',
    of: [{ type: 'orderItem' }],
  },
],
```



```
};
```

**Order item schema:**

```
export default {  
  name: 'orderItem',  
  title: 'Order Item',  
  type: 'object',  
  fields: [  
    {  
      name: 'product',  
      title: 'Product',  
      type: 'reference',  
      to: [{ type: 'product' }],  
    },  
    {  
      name: 'quantity',  
      title: 'Quantity',  
      type: 'number',  
      validation: Rule => Rule.required().min(1),  
    },  
    {  
      name: 'price',  
      title: 'Price',  
      type: 'number',  
      validation: Rule => Rule.required().min(0),  
    },  
  ],  
};
```