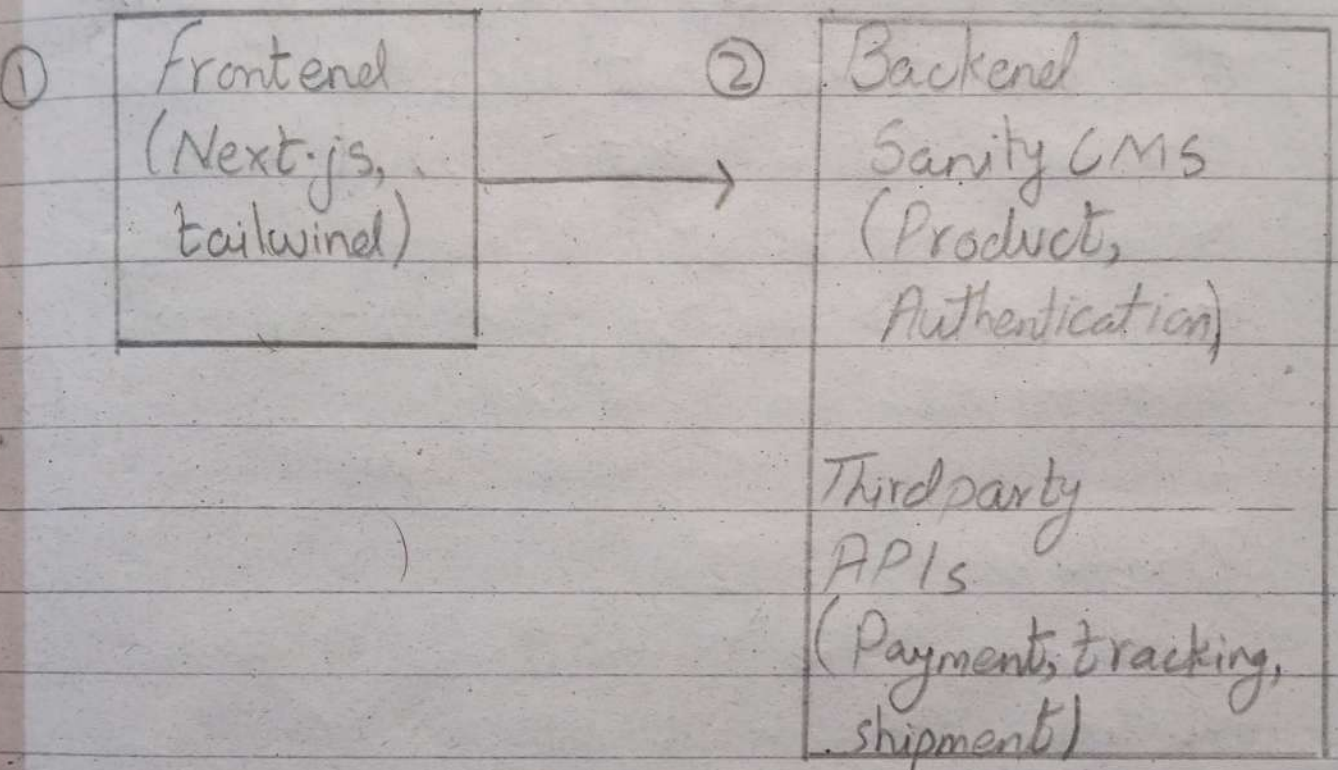
Market place Name= chromacurate

1) System Architecture Overview



Description:

① Front end (Next.js and tailwind)

→ It helps users to route between the items and pick the one they like. The cart, payment and product details will be displayed with the help of Next.js and the styles were given by tailwind

→ Responsiveness

→ Fast loading time

Backend

Sanity CMS:

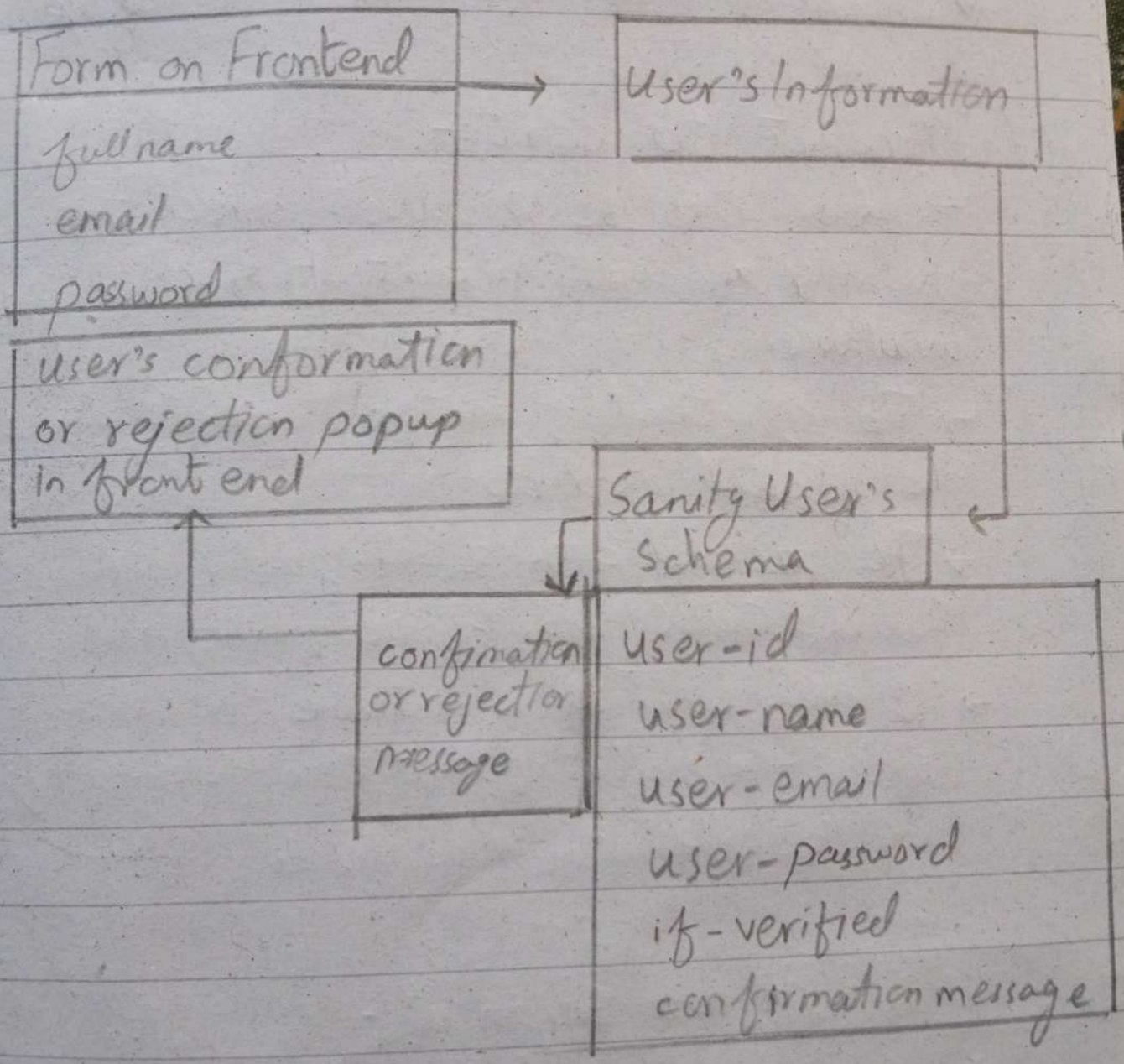
→ Sanity is a control management system that can take input or show output for information like:- Product Name, Stock, rating etc. It helps developer to fetch by writing grock query and display data on front end.

Third-party APIs:

→ Handles the payment process when the user pays. shows realtime tracking of order and shipment details.

Diagram Of System Architecture

① Front end (For login OR signup)



Work flow Of System Architecture

* Expecting that the user is signed up

1) Product to cart:-
→ User browse products select the quantity and adds to cart. If stock is available.
→ The frontend of cart will update due to the selected products.
→ The backend checks Sanity CMS for Product details (PN, PS, PC), size, quantity and stock.
→ If every parameter is available the cart will show the product if Not then the error of available product will be shown.

2) Payment and Shipment:-
→ Upon clicking checkout button user will be required to fill a form containing his payment method. selecting one of many options, his address for shipment and zone.
→ The process of payment will be done by payment API like Jazz cash, easypaisa

→ Or we can use tool like snipcart.io to handle cart management and delivery.

→ The user will be update on the delivery status.

API Requirements:

1) Authentication

·) End-point Name = /authentication

Method: POST

Description: Sign in OR To login users.

Example: {"email": "user1@gmail.com", "password": "123user", "username": "firstuser"}

2) Product

·) End-point Name = /products

Method: GET

Description: To fetch details of products

Example: {"id": 001, "name": "hoodie", "size": "L", "category": "casual wear", "price": 1000}

3) Product Details:

.) End point Name= /productname

Method: GET

Description: To fetch the detail of a single product.

Example: {'id': 001, "name": "hoodie", "size": 2', "price": 1000, "colour": "blue", "sticker": "batman logo", "stock": "only 5 available", "Rating": "4.7 out of 5.0", "rating No": "39", "Description": "A blue coloured hoodie of size large with a batman logo on back"

4) Order/cart:

.) Endpoint Name= /cart

Method: GET

Description: To place New orders.

Example: { "customer Id": 1, "cartItems": [{ "product Id": 001, "Quantity": 2 }]

5) Sanity Schema Example:

This will be an example schema so all fields won't be included

① Product:

```
export default {
  name: 'product',
  type: 'document',
  Feilds: [ {name: 'name', type: 'string', title:
  'product Name'},
  {name: 'Price', type: 'number', title: 'Price'},
  { name: 'description', type: 'text', title: 'description'},
  {name: 'category', type: 'string', title: 'category'},
  ]};
```

② Cart:

```
export default {
  name: 'cart',
  type: 'document',
  Fields: [
{name: 'custamer Id', type: number, title: 'kamlesh'},
{name: 'Products', type: 'Array',
  fields: [{name: 'ProductId', type: 'number', title: 'hoodie'},
           {name: 'ProductId', type: 'number', title: 'shirt'},]
  title: 'Products'}, ]};
```

3) Order:

```
export default {
name: 'order',
type: 'document',
feilds:[
{name: "customerId", type: 'Number',},
{name: 'cartItem", type: 'array',
feilds [{name: 'Product ID', type: 'number'},
        {name: 'quantity', type: 'number'},
       title: "cart Item"},
      {name: 'status', type: 'string', title: 'orderstatus'},
      {name: 'payment', type: 'string'; title: "Payment"},
      ]};
```

Diagram