Day 2 Assignment: Planning the Technical Foundation for the Sit & Style Studio;

Web-System Design Overview:

The system is built to be smooth and easy to scale. The frontend works well with the backend to give users a simple and enjoyable shopping experience. Here's a quick overview of how it works:

Frontend:

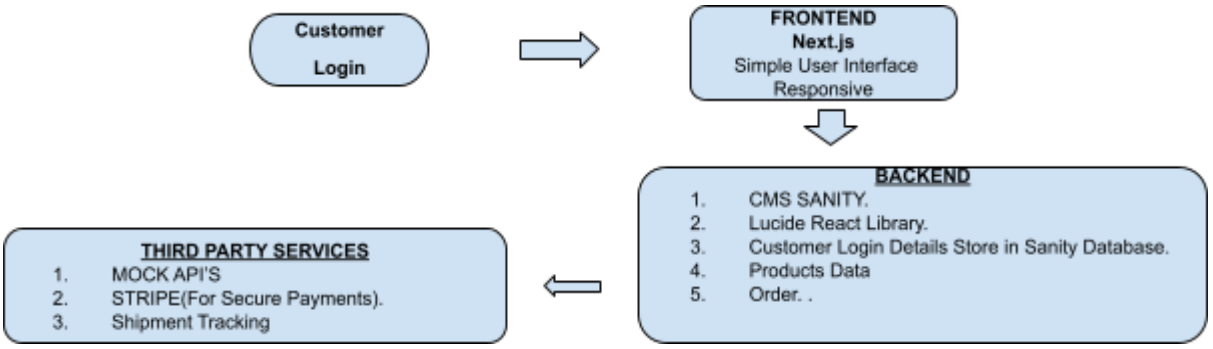
- Built with **Next.js** for a fast and efficient frontend.
- Styled using **Tailwind CSS** and enhanced with UI component libraries like **ShadCN UI** and **Lucide React**.
- **Context API** is used for state management.
- Backend data is fetched through **API calls**.

Backend:

- Data handled via Sanity.io
- Third-party Services e.g. Product Data APIS, Shipment Tracking, Payment
- The frontend connects to

Sanity.io through —------>**HTTP requests** (GET, POST, PUT, DELETE).

Workflow:



Sanity Schema Details Example

1. GET /api/products:


```
[ {
    "id": 1,
    "name": "Wooden Chair",
    "price": 49.99,
    "stock": 15,
    "category": "Furniture"
  },]
```


2. POST /api/orders:

Schema (Request):


```
{
  "customerId": 123,
  "items": [
    {
      "productId": 1,
      "quantity": 2
    },
    {
      "productId": 2,
      "quantity": 1
    }
  ],
  "totalPrice": 299.97
}
```


Schema (Response):


```
{
  "orderId": 456,
  "status": "success",
  "message": "Order created successfully."
}
```


3. PUT /api/users:

Schema (Request)

{ "name": "John Doe",

  "email": "john.doe@example.com",

  "phone": "+1234567890"}

Schema (Response):

{ "userId": 123,

  "status": "success",

  "message": "User details updated."}

4. DELETE /api/orders:

Schema (Response):

{ "orderId": 456,

  "status": "success",

  "message": "Order deleted successfully."}

Day 2 Summary:

"I've outlined how the system will work, how components will interact, and how third-party APIs will be integrated. The next step is to bring this design to life by building a fully functional marketplace."