day 2 of MarketPLace Hackhtahon

Goal of Ferniro

"Join us in reshaping the furniture shopping experience. Whether you're a customer looking for your dream piece or a seller wanting to reach more buyers, FURNIRO has you covered!".

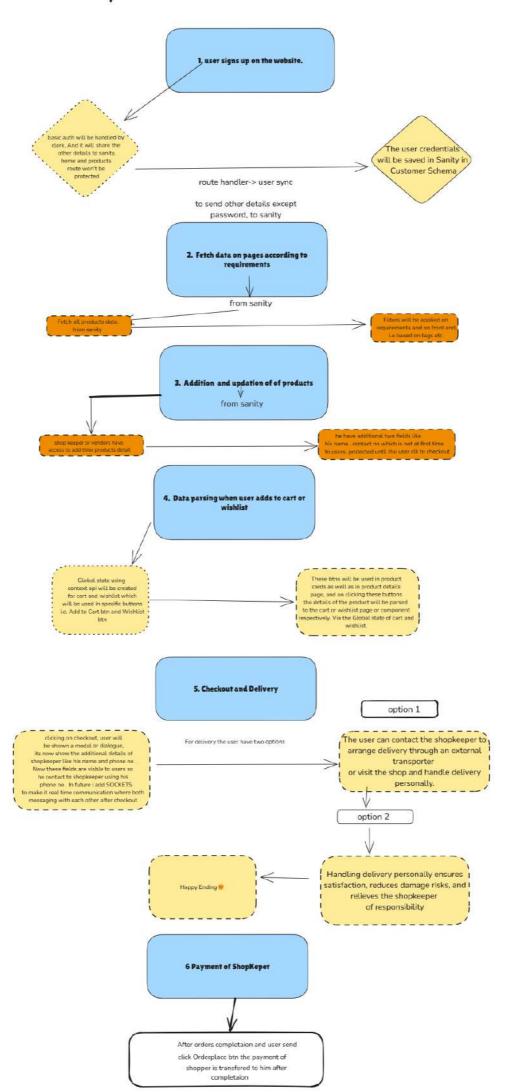
Technical Requirements OF Furniro

- 1) Frontened
- a) I am using Next js with type script as a ui framework for frontened and for styling use tailwind css with Shaden ui for Components library.
- 2) Backend
- a) I am using Sanity for data management as part of the backend. Product details, order details, customer details, and vendor details, such as contact numbers, are key aspects of my planning.
- 3) Third Party Api
- a) Clerk for authentication
- b) stripe for payment if possible

Part of future plans

- c) sockets for real time communication
- d) tuilo or any other providers for sending notificational sms to both vendors and users . and also the admin

system Architecture overview



WORKFLOW DIAGRAM



2. Clicks on the product

based on the product id further details will be filtered from the data fetched from sanity.

3. If clicked on Add to cart or Add to Wishlist

The global states will be updated with the product details that will be displayed using that same Global state that will be available on cart or wishlist route or component.

4. User Clicks on the checkout to finalize the purchase

user will be shown a Dialogue, showing the contact information user about shopkeeper name and contact no For confirmation of product

5. Delivery

For delivery user have two options

1)"The user can contact the shopkeeper to arrange delivery through an external transporter or visit the shop and handle delivery personally."

2)"Handling delivery personally ensures satisfaction, reduces damage risks, and relieves the shopkeeper of responsibility."

Payment

After orders completaion and user send click Orderplace btn the payment of shopper is transfered to him after completaion

ROADMAP

Tasks

Day

Day 1 (%)	Configure Sanity and Clerk. Initialize schemas for products, users, and orders.
Day 2 (2)	Build product-related APIs: Add, fetch, update, and delete products. Test using Postman and integrate with the frontend.
Day 3 (*)	Advanced product features: Search, filters, wishlist, and cart. Implement and test their respective APIs and frontend functionality.
Day 4 (=)	Integrate payments. Add payment-related APIs, configure the payment gateway, and implement the checkout flow. Test all payment workflows (success, failure).
Day 5 ()	Final touches: Test all API integrations, refine the frontend, optimize queries, and deploy. Ensure seamless order creation, checkout, and payment confirmation.