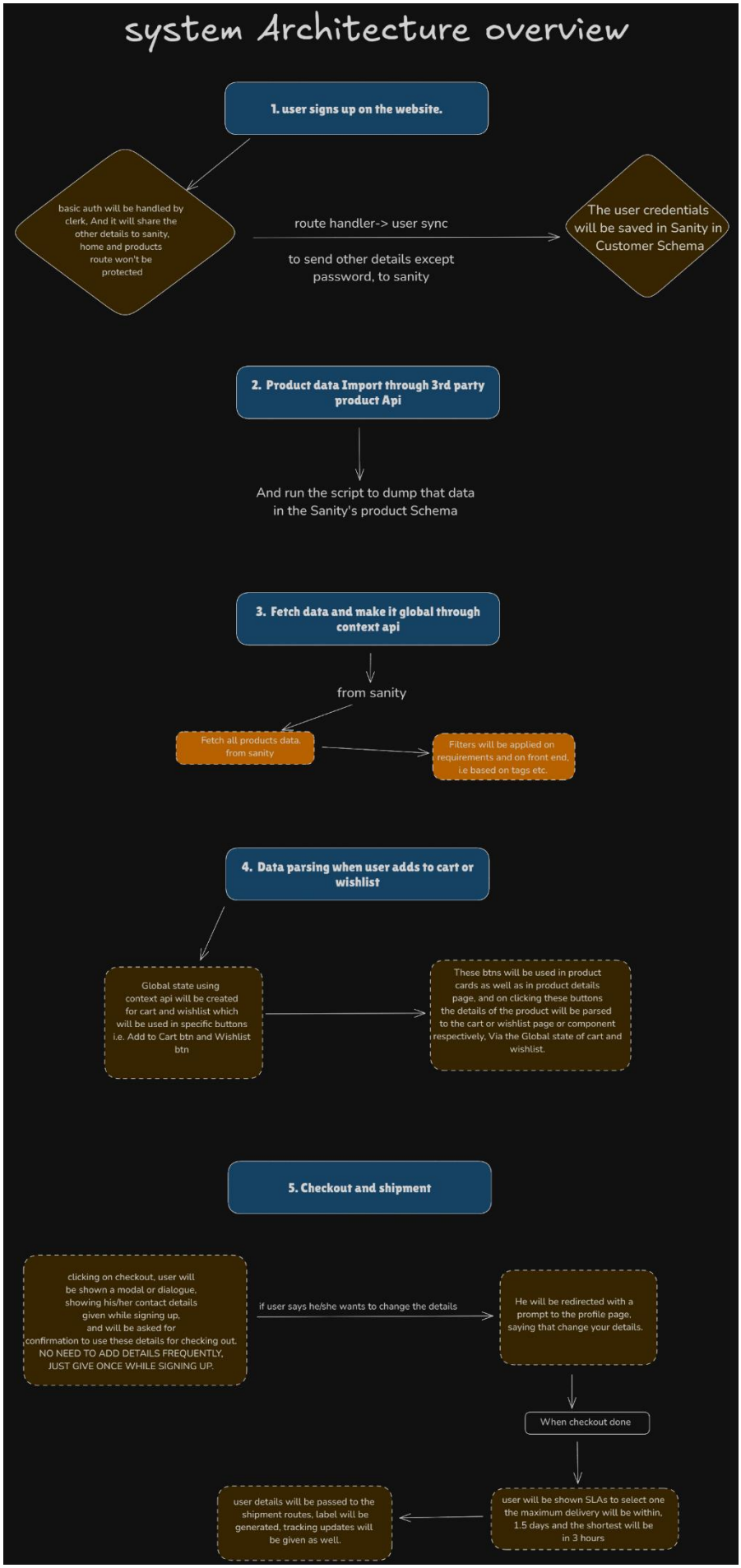
Marketplace Technical Foundation - Hiply



Marketplace Technical Foundation - Hiply

Marketplace Technical Foundation - Hiply

API Specification Document:

Note: This doc is the provisional version, on requirements I'll be making changes in the apis.

ROUTE	METHOD	RESPONSE
• api/userSync	Post (will be used to send contact details of user when, signs up from clerk).	"User details Saved" and an instance showing, details are saved. Status: 201.
• api/shipment	Post (will be getting user contact information for shipment)	"shipment created", and an instance of to and from objects. And shipment price will be returned to the frontend that will be added in the final price of the product Status: 201.

• Api/label	Post (will be getting the shipment id to create final label.
	of order details)
		{
		createdBy: {
		firstName: 'habib',
		lastName: 'Ullah',
		username: 'habibahmed918131@gmail.com'
		},
		parcel: 'e8d5efd6c85e41319aa59a8ea1601748',
		rate: 739d18c5750b42c0a06eb4983aab6417',
		status: 'SUCCESS',
		test: true,
		trackingNumber: '7590135442',
		trackingStatus: 'UNKNOWN',
		id=7590135442'
		}
		
		status: 201

• Api/tracking	GET	The tracking status of product i.e. dispatched or not.

Marketplace Technical Foundation - Hiply

Data Schema Design And relations between Schemas

Product:

- Id
- Name
- Description
- Reviews
- Price
- Available colors
- reviews
- Sizes
- Stock
- Tags
- Categories
- Returnable → whether the product is eligible for returns even.
- Return window → Time within product return will be accepted upon delivery.
- On arrival check guidelines → Specific points customer should check on arrival.

Order:

- Order Id
- Product details
- Customer details (from customer schema this will be a reference)
- Status i.e. (delivered, returned, asked for exchange)
- Timestamp it is ordered.
- Delivery expected time according to the selected SLA.
- Photo on delivery → photo of the product state when delivered (if the user opened and checked it).
- Customer confirmation → Boolean or maybe the signature of the user confirming the he/she accepted the product.

Delivery Zones:

- Gulshan -e- Hadeed
- Steel town
- Port Qasim
- Shah Latif

Customer:

- Customer Id.
- Name
- Contact details
- Address
- Order history
- Return rate ? → percentage of product user returns, will be calculated using "Order history" and will be starting calculating it after customer has successfully ordered at least 1 item.
- Exchange rate ? → percentage of product user exchanged, will be calculated using "Order history" and will be starting calculating it after customer has exchanged at least 1 item.

Return:

- Return id
- Order, a reference to that order customer wanna return the delivery of.
- Customer, reference to that customer's schema doc who wanna return it.
- Product, which will also be reference of the products user wanna return, with the additional information of quantity and Reason for return, from the list.
- Detailed Reason
- Request Status.

Exchange:

- Exchange Id.
- Order, the order wanna change.
- Customer, reference
- Original product, reference to the arrived product.
- Request product, reference of the requested product.
- Exchange reason, with a list of reasons.
- Exchange attempt number, counts the number of attempts at max 3 will be allowed.
- Status

RELATIONS BETWEEN SCHEMAS

Product & Order Relationship:

- Products are referenced within orders through an array of product references (to show how many products user have ordered, if many)
- Each order can contain multiple products with their quantities.

- This allows tracking which products were purchased in each order.

Customer & Order Relationship:

- Each order has a reference to exactly one customer (through customer id)
- Customers have an array of references to their orders (order History, through order ids)
- This creates a two-way relationship where you can:
  - Find all orders for a specific customer
  - Find the customer who placed any specific order

Order & Delivery Zone Relationship:

- Each order references one delivery zone, i.e. the user selected delivery zone.
- Delivery zones can track multiple active orders, i.e. can hold reference to multiple order ids, if multiple users have placed order in the same zone, making it easy to give information to the delivery partner dealing with that specific zone.
- This helps in managing deliveries by geographic area.

Technical Roadmap

Milestone →1: Define Schemas and import product data from 3rd party product Api

- Define all the schemas i.e. product, order, Delivery Zones, Customer, Orders
- When defining schemas add proper references in schemas to make relations in them i.e. order's relation with customer schema and product etc.
- Import the product data from the 3rd party api into sanity.

Milestone →2: Import the data from sanity on frontend, and globalize it using Context api, so that you don't have to fetch data on every route and component repeatedly reducing the api calls.

Milestone → 3: The frontend functionalities. Implement all the functionalities on all pages.

- Product display on Home page and shop pages.
- Different Categories and tags filter in the components i.e. new arrival products, summer collection, differentiate or filter them using filter logics.
- Implement Filters functionality on Shop page on the fetched data which will be available through the global data state, available in context api. The filters will be based on Tags, Categories, price and old to latest and latest to old.
- In product cards make the add to cart button and add to Wishlist button functional by adding this functionality in buttons → on clicking the Cart button it should pass the product's details to the Global state available through the

context api, and those details will be imported or extracted in the cart component to display the products in cart when the button is clicked. Same process will be repeated for wishlist component.

- Add the DELETE functionality in the wishlist and cart component, so, user can remove products as well from the cart and wishlist. Add the increasing and decreasing buttons as well in the cart component to increase/decrease quantity.
- In the cart add the "Total" calculation as well and "Checkout button" to proceed to checkout.
- Create the product's details page, with all necessary functionalities i.e. add to cart, add to wishlist, increase/decrease quantity, size selection, dynamic price which will be updated based on quantity and size (if larger sizes are expensive than the default or small), Buy now button.
- Create a related products component to display related product on the product details page (OPTIONAL OR CREATE IT LASTLY)
- Create the other necessary functionalities on pages like, about, contact, blog, with the contact forms being functional as well.

Milestone →4: Implement Auth using Clerk, and protect the checkout route. Also implement the logic to send the extra user details collected during signup i.e. phone number, address or nearest place, email etc. to Sanity's Customer schema.

Milestone → 5: Create user profile with user information available in the sanity's Customer Schema that you just sent in sign-up also create a session for the signed in user. And add the user's contact/delivery information to the Context Api and it will remain connected to the profile i.e. change and the profile will change the delivery details in context api too.

Milestone → 6: Create the route handlers, for shipment.

Milestone → 7: When user click on checkout he/she should be shown a dialogue displaying their delivery information with 2 buttons continue with this info, or change it. If user clicks on edit, he/she will be redirected to the profile where they can edit their details, and the edited details will be added to the checkout component. And also, there will be the option to select the SLA there will be 3 ideally, one with the fastest delivery and same exchange and return policies and then with a bit of delay respectively. Users also have to select their Zone as well.
PAYMENT OPTION WILL BE CASH ON DELIVERY INITIALLY.
Clicking on Purchase the details will be sent to the shipment route. From there the Labels will be generated for delivery partner and tracking number will be provided.

Milestone → 8: Implement Parcel tracking.

Milestone → 9: Implement the return and exchange policy with their logics, with adding the return and exchange component in the users profile as well in which user have to provide reason for return and if exchanging, the exchanging reason and the id or ids of products they

wanna order instead, i.e. Customer can ask for return with in a limited time frame after delivery of product and customer can return for limited number of times i.e. 3 times etc. If Customer requested exchange the shipment process will be repeated.

Milestone →10: For the delivery partner implement the on-arrival product's image sharing i.e. when the user checks the product on arrival the delivery partner will also click an image of the product as well for its "delivery time state" and also ask for customer confirmation that he/she finalized the reception of the product.