

- 1- Sign Up
Full Name, Email, Mobile, Password, Confirm Password
- 2- Login
Email, Password.
- 3- Profile(Update profile information)
Full name, Email, mobile (should have confirm email/mobile when updating both), profile picture, etc.
- 4- Change Password
Current Password, New Password, Confirm Password.
- 5- My Orders(tab1)
 - a. Orders(Order Number, Order Date, Transaction Id, Total, Grand Total)
 - b. Order Details(Order Number, Item, Picture, Quantity, Price Total)
For digital products like documents, there should be a document viewer, and video player for videos also it should be downloadable if allowed by the app owner.
- 6- My Subscriptions(tab2) **(for video classes or livestream sessions)**
 - a. Subscriptions(Order Number, Order Date, live Stream(Yes/No), Transaction Id, Total, Grand Total)
 - b. Subscription Details(Order Number, Order Date, Start Date, End Date, Life time(Yes/No), live Stream(Yes/No), Start time, Price, Quantity, Total).
For Live Stream Sessions there should be JOIN button (mic on/off, video on/off and End session buttons)
- 7- Cart
Picture, Item, Price, Quantity, Total, Cart total, Cart Items
- 8- Checkout
Order Summary (Picture, Item, Price, Quantity, Cart total, Cart Items) → Optional (since app is small)
Shipping Information (Full Name, Email, Address, City, State, Zip Code, Country)
- 9- Store
Items Display in boxes (Picture, Item Name, Price, Add to Cart (button), View (button))
- 10- Search Product: Search and display results → ((Picture, Item Name, Price, Add to Cart (button), View (button))
- 11- Search Between Dates(mainly for subscriptions based products with dates) → (Picture, Item Name, Price, Add to Cart (button), View (button))
- 12- Users Management: Users can be customers or system owners/administrators → (give admin privileges, permission to specific pages if necessary, active/deactivate, etc.)

13- Inventory Control

- Warehouse management ➔ this is used to manage the warehouses for flexible inventory control.
- Products Management: This is used to add new products to the store, or update existing product information. (Item Name, Listing Price, Buy Price, Sale Price, Quantity, Picture, Display on Store (Yes/No), downloadable (Yes/No) ➔ for digital products, etc.) (We should be able to mark any item in store as “Damaged”).
- Inventory List ➔ this is the inventory list report that also show the re-order levels.
- Product return ➔ this is to manage the previously ordered items that has been returned by customers, this should be reflected in the total inventory in store.
- Damaged items ➔ this is to manage the damaged items in the store which in turn could be repaired or disposed.
- Disposed items ➔ this is used to manage unrecoverable items that is to be disposed which will be identified as loss transaction. (N.B: loss transaction should be generated)

14- Promotion Management (Packages) ➔ (Item Name, Listing Price, Buy Price, Sale Price, Quantity, Picture, Display on Store (Yes/No), downloadable (Yes/No) ➔ for digital products, Package(yes/no: if yes only overall price will be shown otherwise each item price will be shown and the grand total),etc.

15- Order Delivery & Tracking ➔ this is to manage the delivery of items and tracking its movements until delivered and signed.

16- Sales Management ➔ This is used to manage all sales related tasks like Sales, Sales returns, etc.,

17- Procurement management ➔ this is module is used to manage all the procurement related tasks like making quotations, purchase orders and loading the items into warehouse after delivery and reflecting the inventory list.

18- Accounting reports (Balance sheet(detailed & Summarized), Profit & Loss report/income Statement, Trail Balance, General Leger, Cash Flow, etc.,).

19- Chart of Accounts ➔ this is used to manage a chart of accounts list as well as (add/Edit/Delete) accounts. There should be Standard account and Subaccount numbers generation.

N.B

1. All admin pages like the users, products, promotions management should be separate from the Django admin panel.
2. Some clients may not want to provide subscriptions, mobile application services we should be able to activate/deactivate this services from the clients.
3. This should be designed using Django Rest Framework, and Native React from mobile Application. The Admin Panel should be separate from the web application and mobile applications.