

Type of Project

This is a **Hybrid Web-Based POS System** with **Offline-First Architecture**, intended to automate and streamline operations in a **bar or club setting**.

It's both a:

- **Transactional System:** Handling orders and payments.
 - **Real-Time Communication System:** Between waiters, bartenders, and cashiers.
-

Key Roles & Responsibilities

1. Waiter Interface (Tablet/Web App)

- Takes customer orders.
- Sends orders directly to the bartender system.
- Sees order status (e.g., prepared, in queue).

for now waiter takes order and tell the cashier, the cashier then place the order. The order will then show up in either bartender. or kitchen terminal. this is the priority but if you have time add waiter order. If food kitchen, drink => bartender

2. Bartender Interface

- Sees incoming orders from waiters.
- Marks drink orders as "Ready" or "Delivered".

As a suggestion: you can allow both waiters and cashier to place order. For waiters they can place the order using a unique 6 digit pin code and for cashier using login (better if phone number and password)

3. Cashier Interface

- Views all orders placed.
- Authorizes and finalizes orders.
- Handles payments (cash, card, mobile).
- Prints receipts.

bartender interaction should be minimal, add marking option also let the system mark it after a couple of minutes or....

No need for payment options for now

4. Admin (Optional)

- Manage users, products (drinks, food), inventory, reports.

No inventory (like tracking the available item...), just allow the admin to add food, drinks.. their price... then show the sales based on a selected date range...

How the System Should Work

1. **Waiter takes order** → Sends to **barman** and **cashier system**.
2. **Barman** gets the drink order and marks it ready.
3. **Cashier** confirms order, accepts payment, prints receipt.
4. **Admin** can monitor and configure the system.

print is via cash register machine

Key Features

- Role-based login (waiter, bartender, cashier, admin).
- Real-time synchronization between devices.
- Offline functionality using local storage + sync when online.
- Digital receipts + print support.
- Secure and auditable transaction flow.

The system should transfer data in LAN (between the local computers in cashier, kitchen and barman) also should sync online

Technical Requirements

1. **Frontend:** React.js (or Vue.js) with PWA support (for offline capabilities).
2. **Backend:** Node.js / Django / Laravel – RESTful API with offline data sync support.
3. **Database:**
 - **Local (Offline):** IndexedDB or SQLite (for caching offline data).
 - **Server (Online):** PostgreSQL or MySQL.
4. **Sync Mechanism:**

- Service Workers + Background Sync (PWA pattern).
- Conflict resolution for updates when reconnected.

5. **Receipt Printing:**

- Integrate with thermal printers using WebUSB or a lightweight Electron bridge.
-

Functional Requirements

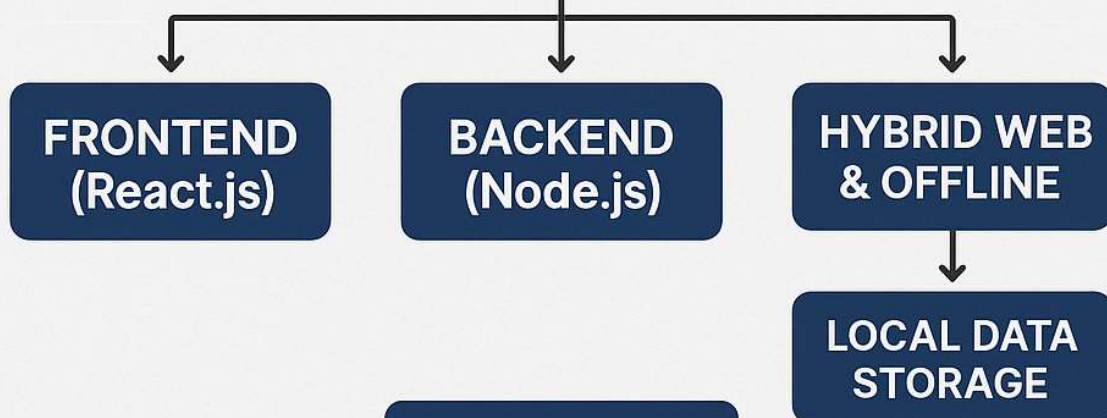
- User Authentication and Role Management
 - Order Placement by Waiters
 - Order View/Process by Barmen
 - Payment Handling by Cashier
 - Receipt Generation
 - Offline Order Queuing and Sync
 - Daily Sales Reports
 - Product/Menu Management
-

Non-Functional Requirements

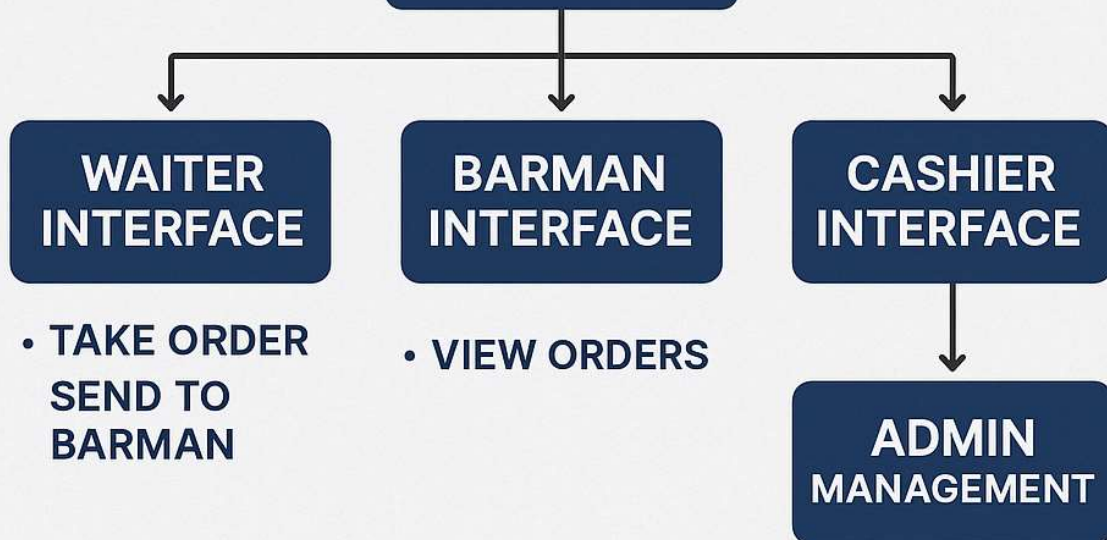
- Must work on low-end tablets.
- Fast local response (offline mode).
- Secure data transmission.
- Sync without data loss.
- Easy to deploy/update (possibly via Docker or Electron for hybrid desktop apps).

POS SYSTEM FOR BAR/CLUB

ARCHITECTURE



FEATURES



- TAKE ORDER
- SEND TO BARMAN