Great! Since you already have **mobile number + OTP login** ready, your next steps should be focused on **role selection, backend setup, and printing flow**. Here’s a detailed roadmap:

**1. Role Selection**

* On login, ask: **“Are you Owner or User?”**
* Store the role in your app state (React Native Context or Redux) or locally.
* Depending on the role, load different screens:
  + **Owner** → Owner Dashboard
  + **User** → User Dashboard

**2. Owner Mode Setup**

1. **Start backend server** (Express) on owner device or cloud:
   * Create endpoints:
     + /upload-print-job → accepts PDFs from users
     + /verify-payment → checks payment status
     + /print-job → sends file to printer
2. **Generate QR code** for users to connect:
   * QR contains backend URL and a temporary token for security
3. **Printer Integration**:
   * USB printer → use Node.js package printer or system commands
   * Wi-Fi printer → send PDF via IPP or printer SDK
4. **Job Dashboard** (optional):
   * Show pending print jobs, payment status, and history

**3. User Mode Setup**

1. **QR Scanner**:
   * Scan owner’s QR code → retrieve server URL + token
2. **File Picker**:
   * Let user select PDF using react-native-document-picker
3. **Print Options & Price Calculation**:
   * Number of pages, B/W or color → calculate amount
4. **Payment Integration**:
   * UPI / Razorpay SDK
   * Send payment confirmation to backend
5. **Upload PDF**:
   * After payment, upload PDF to owner’s server
6. **Print Confirmation**:
   * Backend triggers printing and responds with success

**4. Backend (Express + MongoDB)**

* **PrintJob Schema**: userId, filename, filePath, pages, color, amount, paymentStatus, printed
* **Endpoints**:
  + POST /upload-print-job → receives file, stores temporarily, triggers printing
  + POST /verify-payment → confirms payment success
  + GET /jobs → optional, for owner dashboard

**5. Recommended Development Order**

1. **Owner backend endpoints** → test manually with Postman
2. **Owner dashboard** → start server, generate QR, connect to printer
3. **User QR scan + file selection** → connect to owner backend
4. **Payment integration** → Razorpay / UPI
5. **File upload → printing** → verify end-to-end
6. Optional: real-time updates, multi-user handling

If you want, I can **make a step-by-step diagram showing the flow from login → role selection → QR code → payment → printing**, so you know exactly what happens in each step.

Do you want me to make that diagram?