Absolutely — here's a **comprehensive, organized list of all frontend components** you would need for your REGKYC-inspired prototype. I'll focus on **functional components / UI sections** rather than code.

---

1 Wallet & Authentication Components

- **WalletConnect / MetaMask component**
  - Connect/disconnect wallet
  - Display connected address
- **User status display**
  - Shows whether wallet is KYC-verified or not

---

2 KYC Form Components

- **User Input Form**
  - Fields for age, nationality, optional extra attributes
  - Input validation (e.g., age ≥ 0, nationality format)
- **Local Verification Indicator**
  - Shows if the input passes basic checks before proof generation

---

3 Proof Generation Components

- **ZK Proof Builder / Trigger**
  - Button to generate proof based on user input
  - Status indicator: proof generation in progress / completed / failed
- **Proof Summary / Hash Display**
  - Shows commitment hash, proof hash, public signals (optional for demo)

---

4 IPFS Interaction Components

- **Upload KYC JSON to IPFS**
  - Status indicator: uploading / success / failed
  - Displays resulting IPFS CID
- **View KYC Record on IPFS**
  - Optional: link to IPFS JSON for demo purposes (no PII)

---

5 Blockchain Interaction Components

- **Submit Proof / Store KYC on-chain**
    - Calls smart contract to store IPFS CID + zkProof
    - Displays transaction hash / confirmation status
- **KYC Status Checker**
    - Query the contract to see if user's KYC is valid
    - Show whether NFT has been minted (if applicable)
- **Revoke / Update KYC** (optional for admin/demo purposes)

---

6 NFT Components (Optional)

- **Mint KYC NFT Button**
    - Trigger minting of soulbound KYC badge after proof verification
- **NFT Status Display**
    - Shows token ID, metadata link (IPFS)
    - Indicates verified attributes (ageVerified, nationalityVerified)
- **NFT Viewer / Badge Display**
    - Simple card view of KYC NFT for UX/demo

---

7 Feedback & Status Components

- **Global Notification / Alerts**
    - Transaction success/failure
    - Proof generation errors
    - IPFS upload errors
- **Progress Steps / Wizard**
    - Optional: guide user through form → proof → IPFS → contract → NFT

---

8 Optional Admin / Auditor Components

- **View All Verified Users**
    - List of wallet addresses + KYC status (hashes only)
- **Audit Proofs**
    - Display proof hashes + IPFS CID
- **Revocation Controls**

- Burn NFT or mark KYC as revoked

9️⃣ Optional Documentation / Help Components

- **Instructions / How It Works Panel**
  - Explain proof generation, IPFS, and NFT integration for demo
- **Developer / Test Info Panel**
  - Display testnet info, transaction logs, or snarkjs outputs

---

☑ Summary

At a high level, your **frontend should have components in these categories**:

1. Wallet / Authentication
2. KYC Form / Local Validation
3. Proof Generation & Display
4. IPFS Upload & Display
5. Blockchain Interaction (store proof, query status)
6. NFT Display & Minting
7. Feedback / Progress Indicators
8. Admin / Auditor Tools (optional)
9. Documentation / Help (optional)

This structure will make your **UX intuitive**, and it covers **all critical functionality** without backend.

If you want, I can also **draw a clear component diagram showing how each component interacts** for your report or presentation — very helpful for academic submission.

Do you want me to do that?