KYC Application Architecture

# 1. Tech Stack

- Frontend: React for both mobile (React Native) and desktop (React.js).  
- Backend: Node.js with Express.js, using MongoDB or PostgreSQL for database, JWT for authentication.  
- Deployment: AWS Amplify/Netlify for web, Expo/React Native CLI for mobile, AWS/Google Cloud for backend.

# 2. Features

- User authentication (Sign-up, login, 2FA).  
- User profile with document upload (KYC documents).  
- Verification system (manual or third-party API).  
- Admin dashboard for verifying KYC submissions.  
- Notifications to inform users about their KYC status.

# 3. Steps

- Set up project environment using `create-react-app` for web and `React Native` or `Expo` for mobile.  
- Build UI components, including KYC forms, profile dashboard, and admin panel.  
- Create backend REST API with Express.js and handle file uploads with Multer.  
- Integrate manual or automated KYC verification systems like Jumio or Onfido.  
- Ensure proper security measures including HTTPS, encryption, and secure document storage.