My Recommendation Summary

Putting together what's best practice, here are my recommendations:

1. **Use redirect to Paystack checkout URL** (via authorization URL) rather than building your own UI for card/M-Pesa input — this reduces PCI scope and uses Paystack's secure UI.
2. **Set and use a callback URL**, but do *not* rely solely on it to determine payment success. Use **webhooks + transaction verification** on the backend as the authoritative source.
3. **Frontend should handle redirect flow and status polling**: i.e., after user comes back, check with your backend for final status before delivering value.
4. **Backend must verify the transaction**, check amount, check status, and only then mark your internal record as paid and enable service.
5. **Handle user experience well**: show loading, show success/failed, handle cancellations, provide fallback if redirect doesn't happen, ensure mobile & desktop flows both work.
6. **Future mobile app compatibility**: Since you plan a mobile interface later, aim for a flow that also works in mobile (WebView or mobile SDK). Paystack's guide for mobile WebView is good precedent. Paystack+1
7. **Security & reliability**: Use HTTPS, secure your secret keys on the backend, use webhooks, handle network errors, ensure you verify transactions.
8. **User context & metadata**: Pass meaningful metadata (order id, user id, payment type) when initializing transaction so backend can tie payment to correct user/service. It makes reconciliation easier.
9. **Smooth UI integration**: Given you are using React + Flowbite-React, integrate the "Make Payment" UI nicely, show progress/status, ensure you don't show the UI or service access until payment is confirmed.