

---

# KYC App

## OVERVIEW

Build a Super-Modular Digital KYC [SPA](#) using [React.js](#) which can be used by any kind of financial institution to do the KYC after the user registration is complete.

## BACKGROUND

KYC means “**Know Your Customer**”. It is a process by which financial institutions obtain information about the identity and address of the customers. This process helps to ensure that financial services are not misused. Financial institutions are now required by the RBI norms to have their customers KYC process completed before allowing them complete access to all of their services.

### Documents required to do the KYC:

1. Basic customer details like name, DOB and gender
2. Latest photo
3. ID and Address Proofs like Aadhaar, Voter ID, Driving License, Passport and PAN.

## PRODUCT SPECIFICATIONS

The flow of app consists of **4 modules**, authentication, customer details, selfie, and govt. ID.

- **Authentication:** Design and develop a minimal login only flow where users should be able to login if they already have registered with us.
- **Customer Details:** Take the full name, date of birth & gender of the customer.
- **Selfie:** Open *front camera* inside the browser to take a photo. Show an oval overlay on top of the camera interface to guide the user where they have to put their face.
- **Government ID:** Ask the user to select the government ID from Aadhaar, Voter ID, Driving License, Passport and PAN. After selecting, ask the user to click the photo of front and back(if applicable) side of their selected ID. Switch to the *back camera* for clicking the photos of ID. Show a landscape or portrait overlay, based on the ID selected, on top of camera interface to guide the user in taking a clear photo of their IDs.

Show a success page, after completion of the KYC Flow.

---

## FLOW

Default flow:

*Authentication => Customer details => Selfie => Govt. ID => Success*

## FEATURES

1. You are free to change the design elements and build your own UX
2. Only eligible customers should be allowed to log in. Show error if any ineligible customer tries logging in.
3. Post login pages should only open in mobile browsers where a camera supported inside the browser. Show error message in case the browser is not supported
4. On clicking refresh after logging in, state of the page should be persisted
5. All input fields should have a proper validation
6. Save the customer details in session storage
7. Prompt the permission box to access camera before selfie page, proceed to selfie module only when the camera permission is accepted
8. If permission is denied, show a page to assist the customer in giving the camera access (Build the page specific to Chrome)
9. On selfie page, show tips to take a better photo
10. Show overlay on top of the camera interface according to the mode selected (Selfie / ID)
11. Add edit functionality on each page

## TECHNICAL SPECIFICATIONS

**Technologies to be used** to create this application: HTML, CSS, Javascript, Browser Web APIs, ReactJS, Redux, Material UI, Node, NPM, Webpack, JSONServer.

- Use [Visual Studio Code](#) IDE to create this project.
- Use [create react app](#) as base React app with the latest version of [React.js](#)
- Build and deploy the react app on [Heroku](#) or somewhere else
- Fetch or post any data, use [JSON Server](#) and create dummy APIs.
- Use [web storage](#) options to persist data on page
- Show camera inside the browser using [getUserMedia web API](#)
- You are free to use any CSS framework like [Bootstrap](#) or [Material UI](#)
- Use [CSS variables](#) wherever it can be used
- No usage of jQuery
- Encrypt data when saving information in the [session or local storage](#)

- 
- Decrypt data when fetching information from the session or local storage
  - Use [Redux](#) if you want to use a state manager

## ★ GOOD TO HAVE FEATURES

List of features which are not mandatory but will give you extra points if built.

- Usage of Typescript in places of Javascript
- Test cases and flow charts for each module
- Click best resolution photo with minimal size
- Validation feedback for Selfie and ID photos
- Dark theme support
- Make modules in such a way that the order of pages between Authentication and Success can be shuffled. For example, Auth. => G. ID => Selfie => C. Details => Success

## 🏆 MILESTONES & EVALUATION

This problem statement carries a total of **100 points**.

1. Learn the basic web technologies like HTML, CSS, Javascript and React.
2. Create react app and deploy the production build to Heroku/AWS **(12 pts)**
3. Set up a simple login page by mocking API response **(10 pts)**
4. Build the customer details page with form validation **(8 pts)**
5. Create a modular camera interface in the browser for selfie and govt ID screens **(20 pts)**
6. Build the selfie screen and integrate the camera module **(5 pts)**
7. Build a basic govt ID page with ID selector slider **(5 pts)**
8. Extend the govt ID page to show the front and backside of the selected ID **(5 pts)**
9. Integrate the camera module in govt ID and build the final preview page **(10 pts)**
10. Create error and success pages and show them wherever they are required **(5 pts)**
11. Beautify the entire flow and fix UX issues **(20 pts)**
12. Earn up to **20** bonus pts if you are able to include features from good to have features list

**Note: Maximum points earned by any team cannot exceed 100 points**

---

## WIREFRAMES

Use the below wireframes as a reference, you are free to change the theme or any design element.

### Authentication Screen

**Hi! Welcome to ZestMoney**

One stop KYC solution

**GET OTP**

### Customer Details Screen



**Help us setup your account**  
We'll verify it with your KYC documents

**Your full name**

Ensure it matches name on your PAN

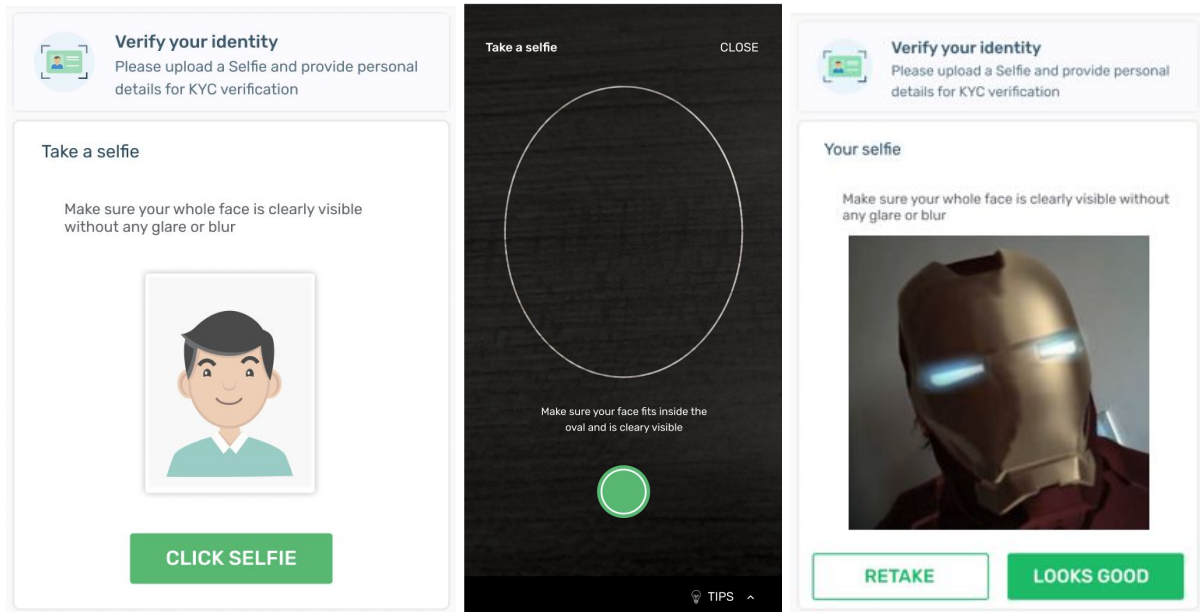
**Your date of birth**

**Your gender**

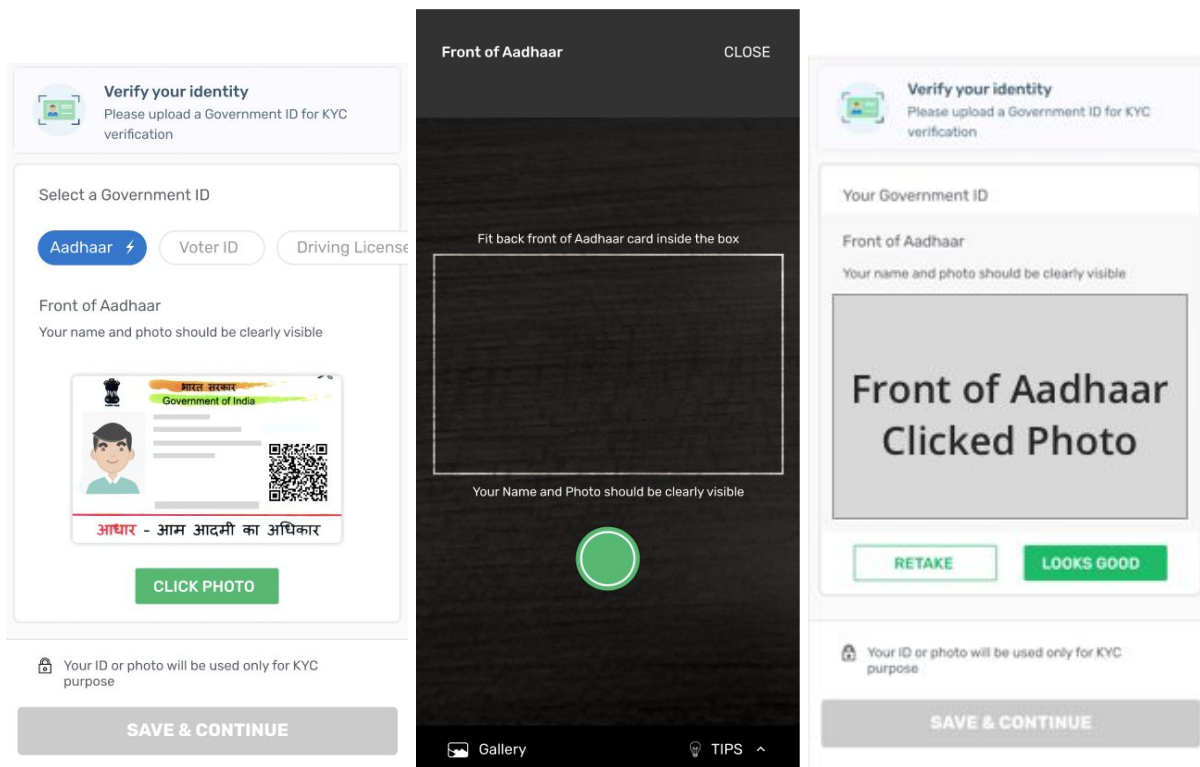
☐☐☐

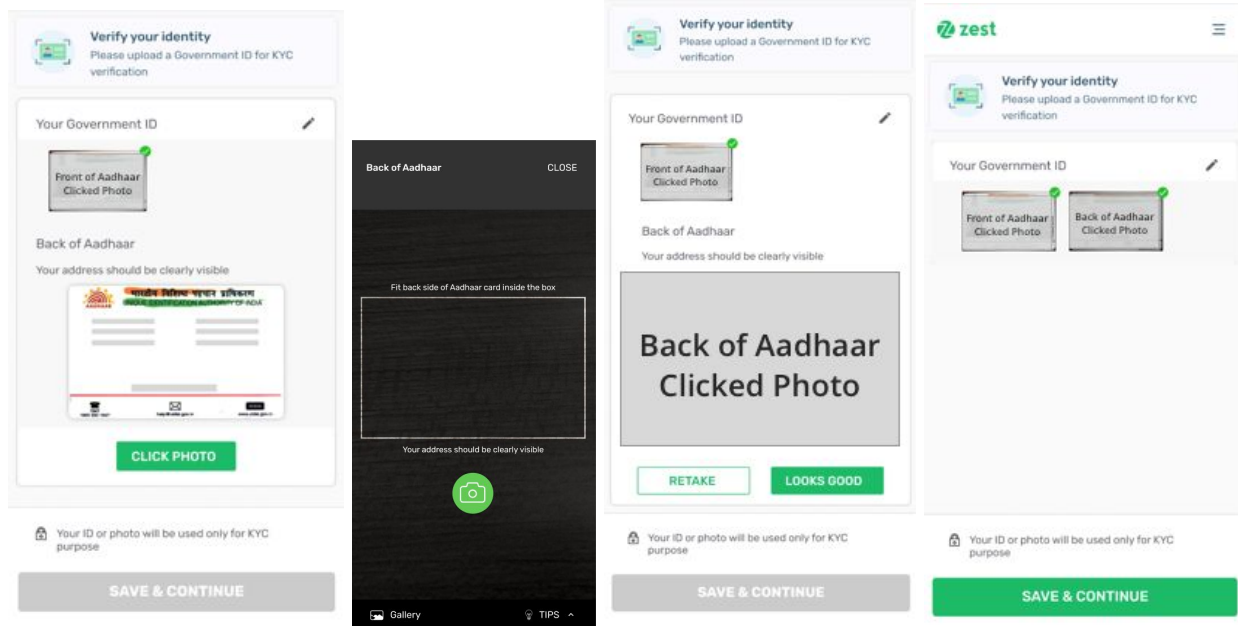
**CONTINUE**

## Selfie Screens



## Govt ID Screens





## Error Screens and Success Page

Use your own creativity to design error scenarios and terminal success page

### **ADVISE FROM PROFESSOR**

Spend the first 2-3 days in learning and brushing up your JS skillset. If you are already proficient in JS, you can directly start working on the problem.

Best resources to learn JS and React:

1. [Intro to Javascript series](#)
2. [React JS Tutorials Series for beginners](#)

### **QUESTIONS**

In case of any concerns, reach out to me at [professor@zestmoney.in](mailto:professor@zestmoney.in)