**PROBLEM STATEMENT**

We received a small application where we need to update user details according to the token provided

* 1. Read the existing profile given the token provided below
  2. Provide a UI to collect all the elements in the name object
  3. Call the relevant API(s) to add the collected information to the user profile
  4. Display the profile data, including the name object

**SOLUTION**

I created a React Application where I show the existing user’s data and also provide an option for updating the user’s data by using a Pop-up form. After updating data, it will reflect on User’s Profile. User can Add or Update this profile fields by submit a simple popup form.

**JOURNEY MAP**

* **Set-up a React Project:** Here, I create a React application and set-up the React Project.
* **Build a UI design:** By using Material UI Kit, create a user profile page and a pop-up form.
* **Organize Folder Structure**: I create a folder structure according to components, Constant and services used. This is for divided our all code structure into different parts which help us easier to understand and reusability of code.
* **Build API Services:** Build an API Services to get or post the API data. It handles the request and response from API also handle error and data according to API Response. I created a separate file for API where all services call on one component page so that all UI pages can access it. Here I manage All error, Internet Not Connected issues and other issues if data not showing on that page.
* **Show User Data:** I call API on UI Component to get user data via token provided and display it on UI Page. The API Token is added in Constant File and can be modified through one page for all services. Getting Data from API I place into the UI Design of User Profile Page.
* **Update User Data:** Add an Update button on top of user data where a MUI Modal is implemented. I create a form inside it which update the user data. When User add data inside the Fields it Update the Given API and User Updated Data with automatically displayed on User Profile Page.
* **Display Message**: I have added a message component inside component folder where all message will be added and show according to the need. We managed it on separate page because if there will changes in message then a developer will do globally.
* **Responsive:** We make mobile support of both Page and Pop. I check in multiple devices i.e. iPhone, iPad and other small and large devices. I have added Media Queries to handle Responsive for Mobile and Tablet.
* **Test Cases:** I also added a test case to check that is user api is working fine. I wrote two test cases to get API Result and if those API not showing result then test cases will show error.

**DESIGN ARCHICTURE**

* **Application**
  + **SRC**
    - Constant
      * Types: Used to add all static types used in application
      * Message: This File is used to store all static messages
      * Path: It is used to store a constant path of domain, api and other static path which is included globally.
      * Token: This is used to store all static token used in application. We can store token i.e Authorization, Firebase etc.
    - Component
      * Message
        + This folder is used to store all functionally to show html of message
      * Users
        + Here all add, updates and other form related to functionality are used
    - Services
      * API Services
        + I prepare a services which includes the error handling, success and network issues. This services provide data and work as middleware between Frontend and API
      * Fetch Services
        + All API are implemented on this page. Developer can call any one of API by using this file / Folder.
    - Helper
      * This folder is used to create a pure component which used to manipulate data or functionality. We can create those features which used multiple times.
    - UI
      * All UI Component are included on this page i.e Model, Table, Button etc. We can used any one of component from this folder to make our UI Pattern
    - App
      * This is main file of Application where all Component are joined. Here we can check the flow of all application
  + **PUBLIC**