**Documentation -VMS UI**

**Setup:**

1. Download node js v10.16.3
2. Clone the git repo with the following command:

2.1 git clone <https://github.com/ketankakade/VisitorsManagementSystem.git>

2.2 cd visitors-management-system

3. After Cloning of Repo run the command --🡪 npm install (this command will install all the required dependency for running our project.)

4. After npm install you will be able to see node\_modules folder in project structure.

5. Now, try to run the project using command 🡪 npm start

This command Runs the app in the development mode. Open <http://localhost:3000> to view it in the browser. The page will reload if you make edits.

6. npm test 🡪 Launches the test runner in the interactive watch mode.

7. npm run build 🡪 Builds the app for production to the `build` folder. It correctly bundles React in production mode and optimizes the build for the best performance. The build is minified and the filenames include the hashes.

Your app is ready to be deployed!

**Technical details:**

1.This project was bootstrapped with [Create React App] (https://github.com/facebook/create-react-app).

2.Complete application is developed in REACT using react features. And Used React Redux and Store for the API calls.

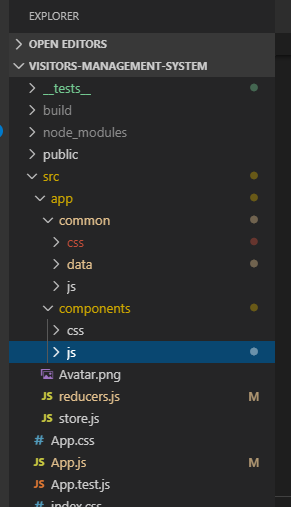
3. Used SCSS for styling of the application.

4. Also Used React Styles to demonstrate the use of styles in react. With Material-UI

5. Used Axios for making an api request in react js.

6. Various Material-ui Components used like AppBar , Sidebar, Dialog,Button.

**Folder Structure:**



In the src folder all the project files are written.

Common- this folder is for common components used in project.

Data – this folder is for all mockup data which is used when apis are not there.

Components- in components folder all the components are there which are used for our project. This is divided again in two folders.

Js- for all the components

**Duck**- for all the redux implementation.

**Actions**- Actions are payloads of information that send data from your application to your store.

**Reducers**- Reducers specify how the application's state changes in response to actions sent to the store.

**Operations**- In this file all the api calls related to that components are made.

**Types**- used for storing all the action types as a constant at 1 place.

**Store-** Holds application state; Allows access to state via getState(); Allows state to be updated via dispatch(action);

*we'll only have a single store in our Redux application.*

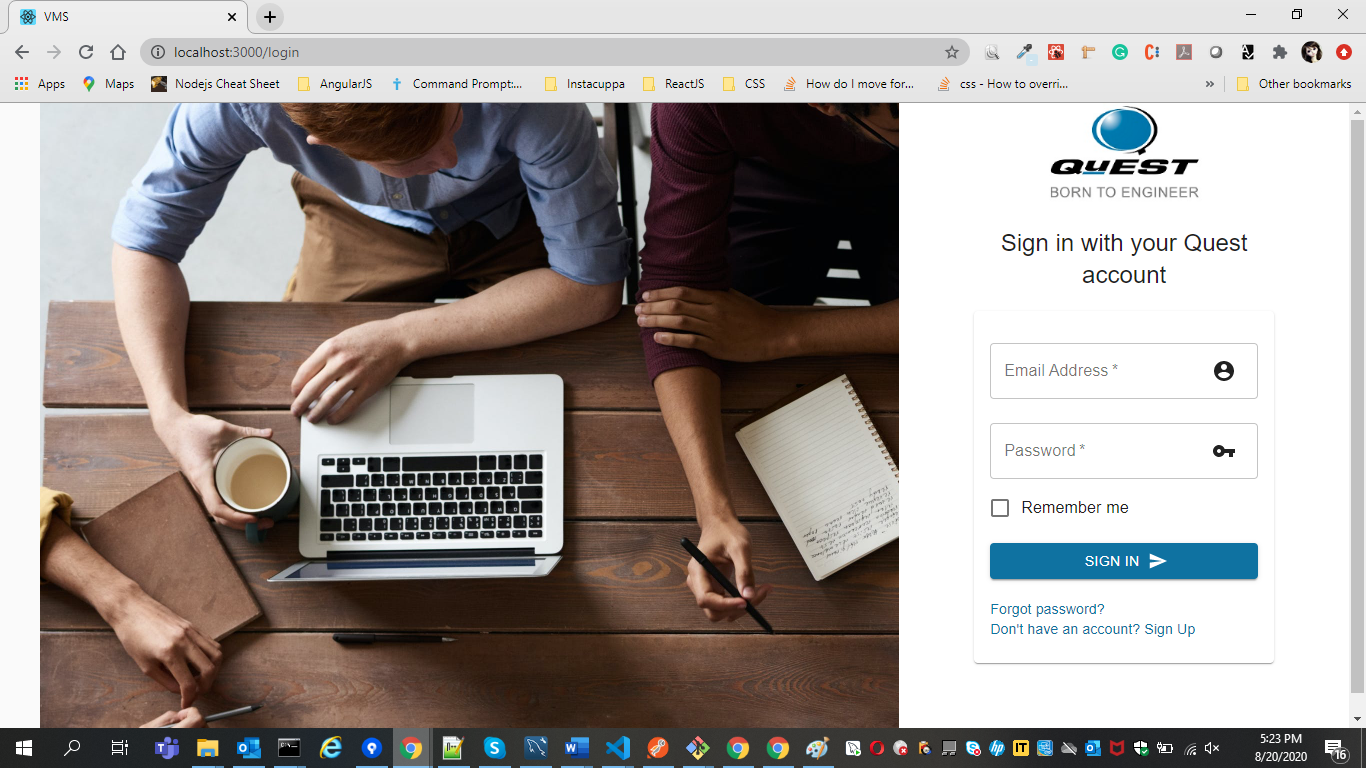
we used combineReducers() to combine several reducers into one. i.e reducers.js We have imported it, and passed it to createStore().

NewTemplate.js- For loading admin components which are login based this template is used for routing as well.

ViiTemplate- For loading VII related components this template is loaded.

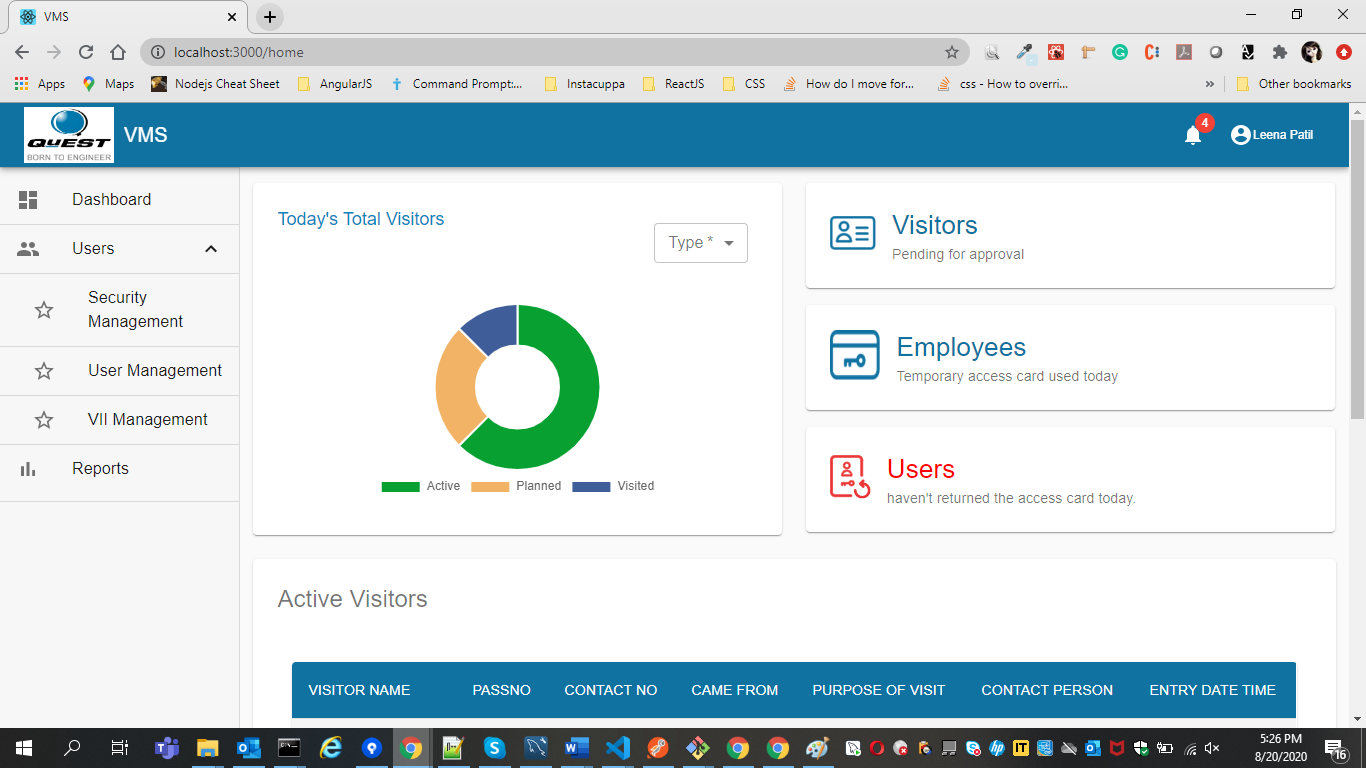
**Screens Developed.**

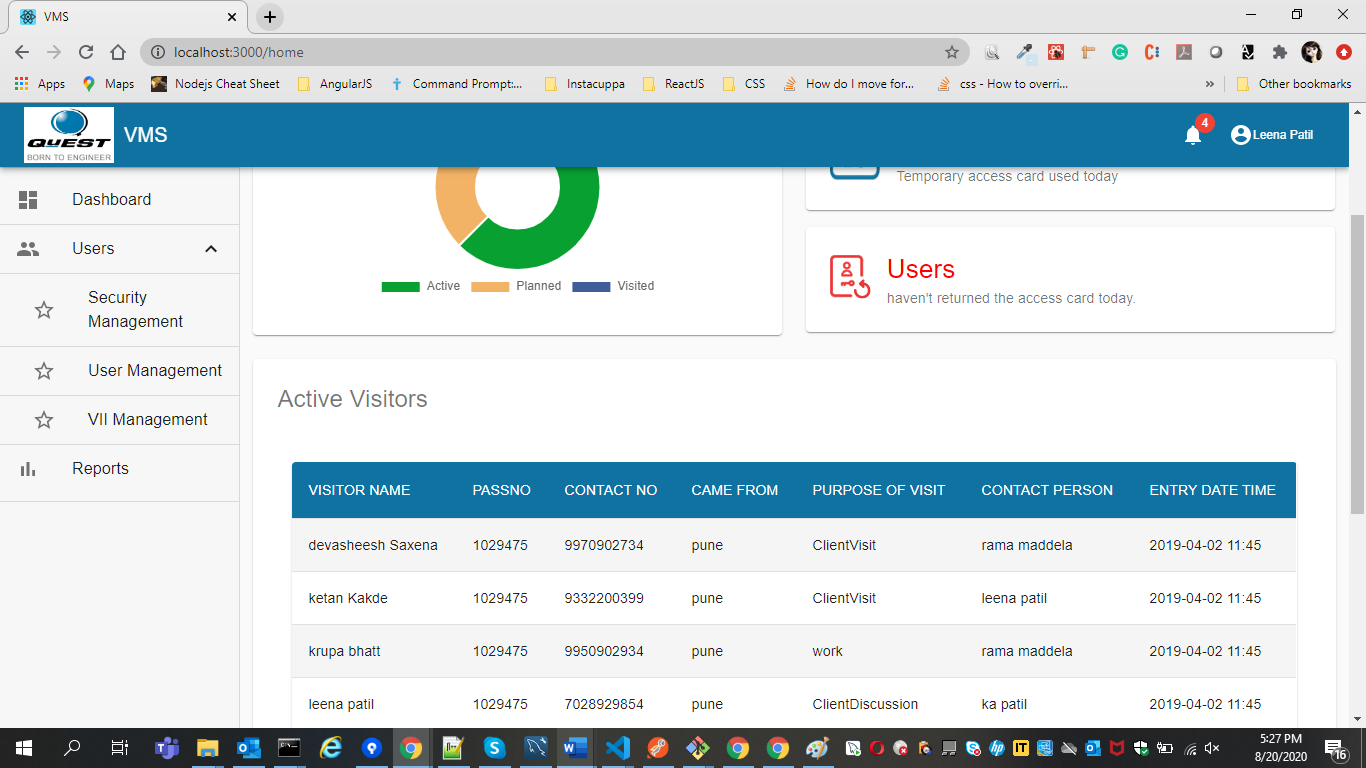
1. ***Admin Login Screen*** – for now you can use credentials from UserData.json file .



All the login validations are fulfilled here for valid email and valid password.

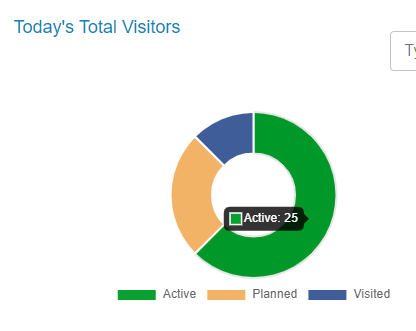
1. ***Admin Dashboard***-once user logs in successfully you can have following dashboard screen.



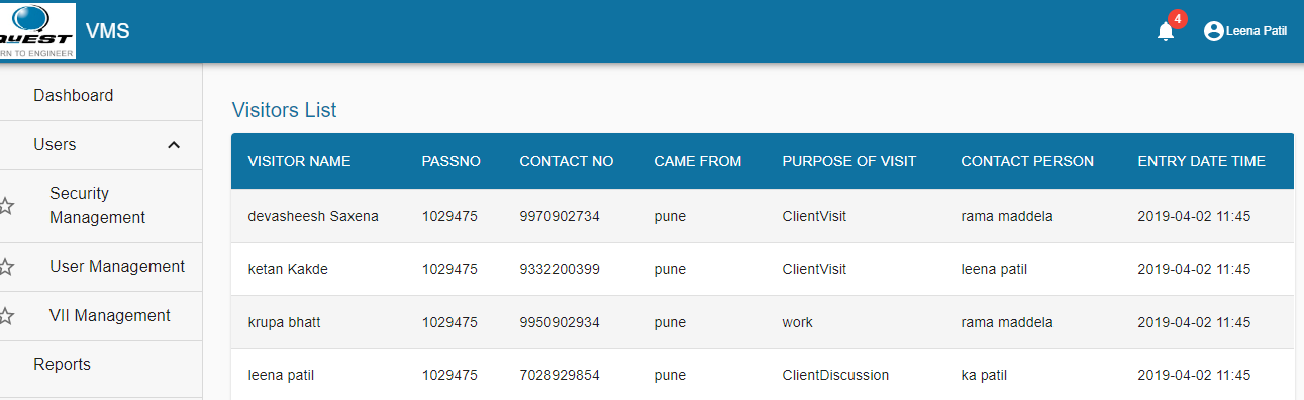


Dashboard is having the active list of visitors. This visitors list is coming through api. – getAllVisitors

Pie chart will show the visitors data as follows after hovering on that category of color.

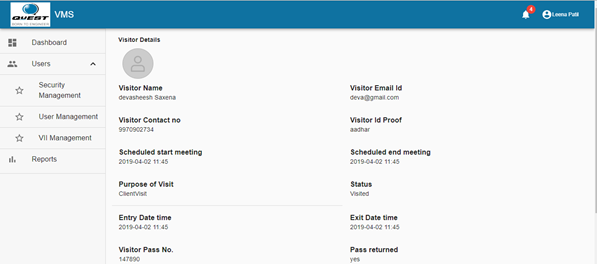


After clicking on the card visitors, you will get list of visitors visited to office.

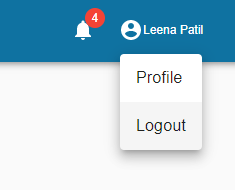


**VisitorDetails.js**

After clicking on visitors, you will be redirected to Visitor Details page

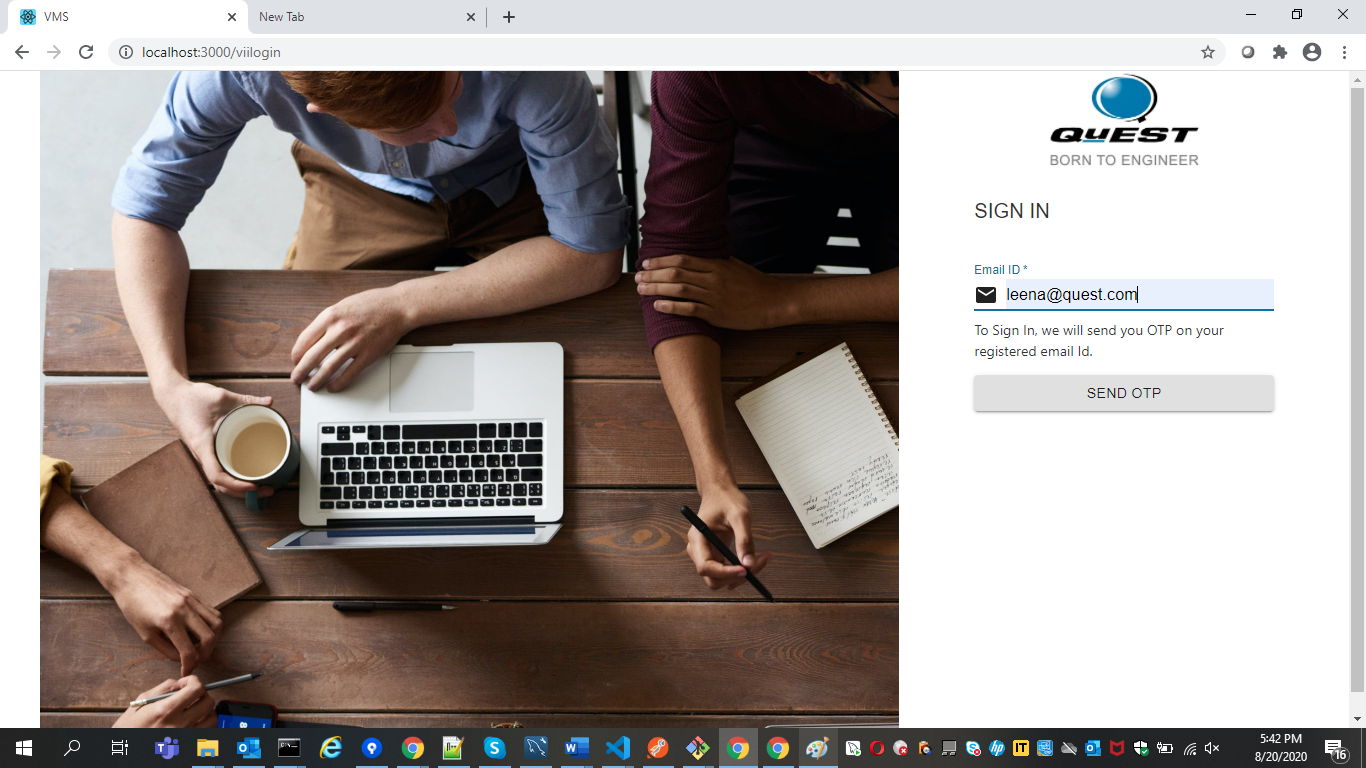


After Clicking on right side user Icon, you will get following two menus: After clicking on logout you will be redirected to login page for admin. You will be able to see from user name who is logged in currently.

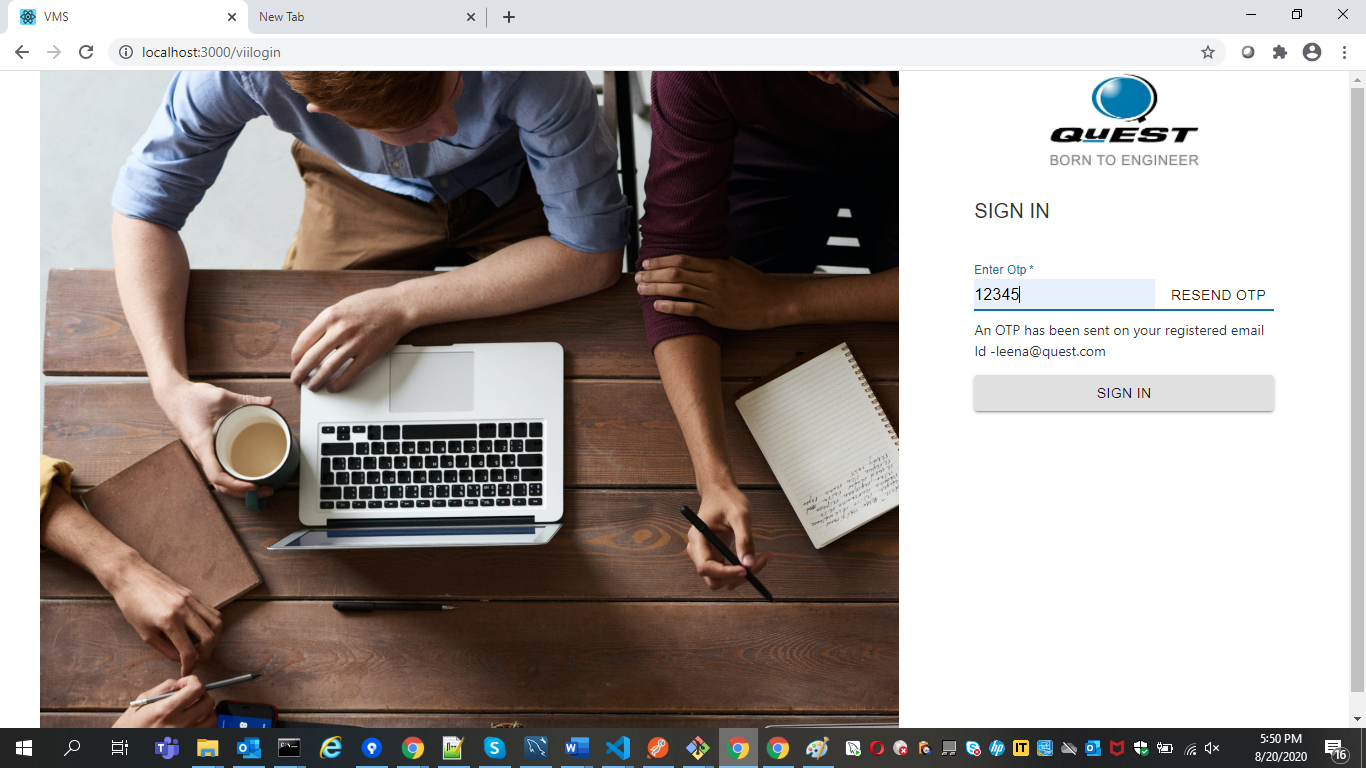


**Visitor Interaction Interface Screens:**

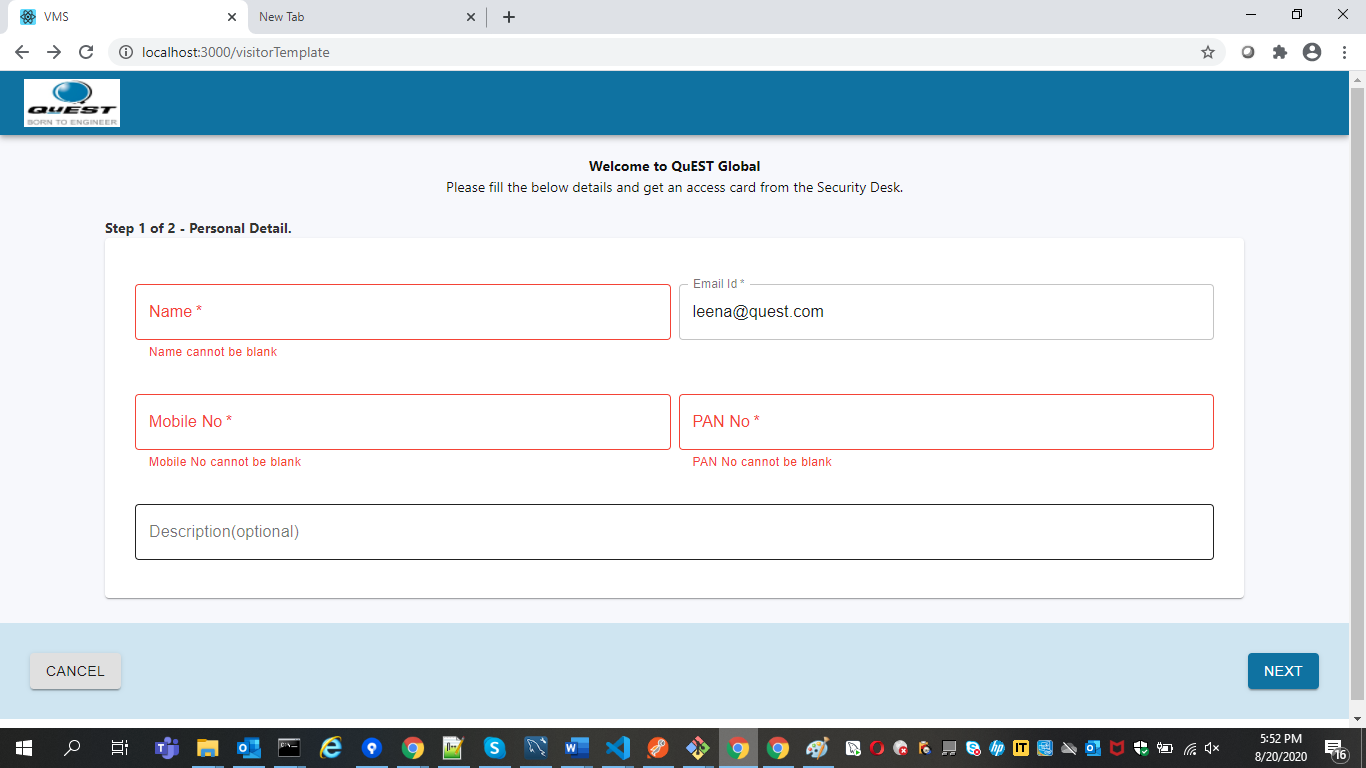
1. **VII First Screen:** in below screen you need to enter valid email address. After this it will ask for otp that was sent to your email address. (but for now it is from UserData.json till the apis are ready.)



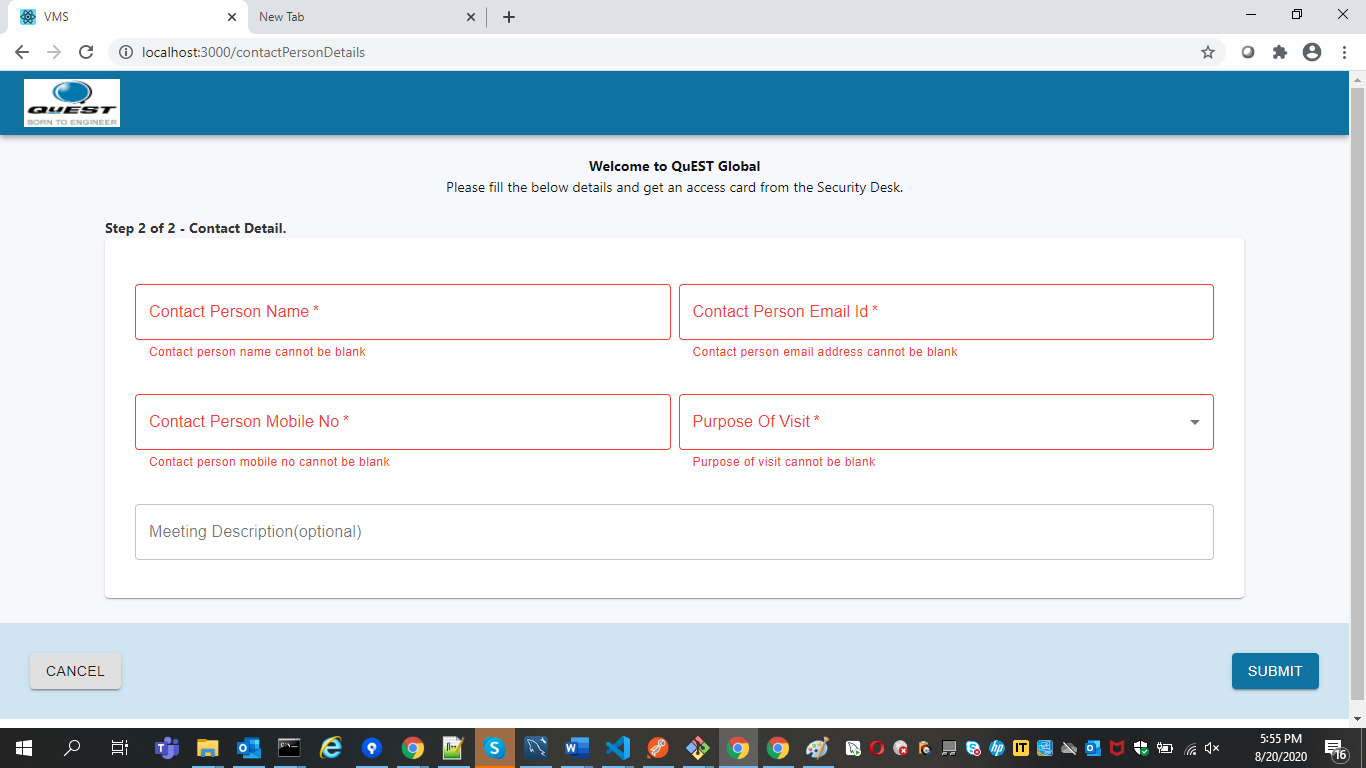
1. **Otp screen for visitor.-** after entering otp visitor can proceed to register.



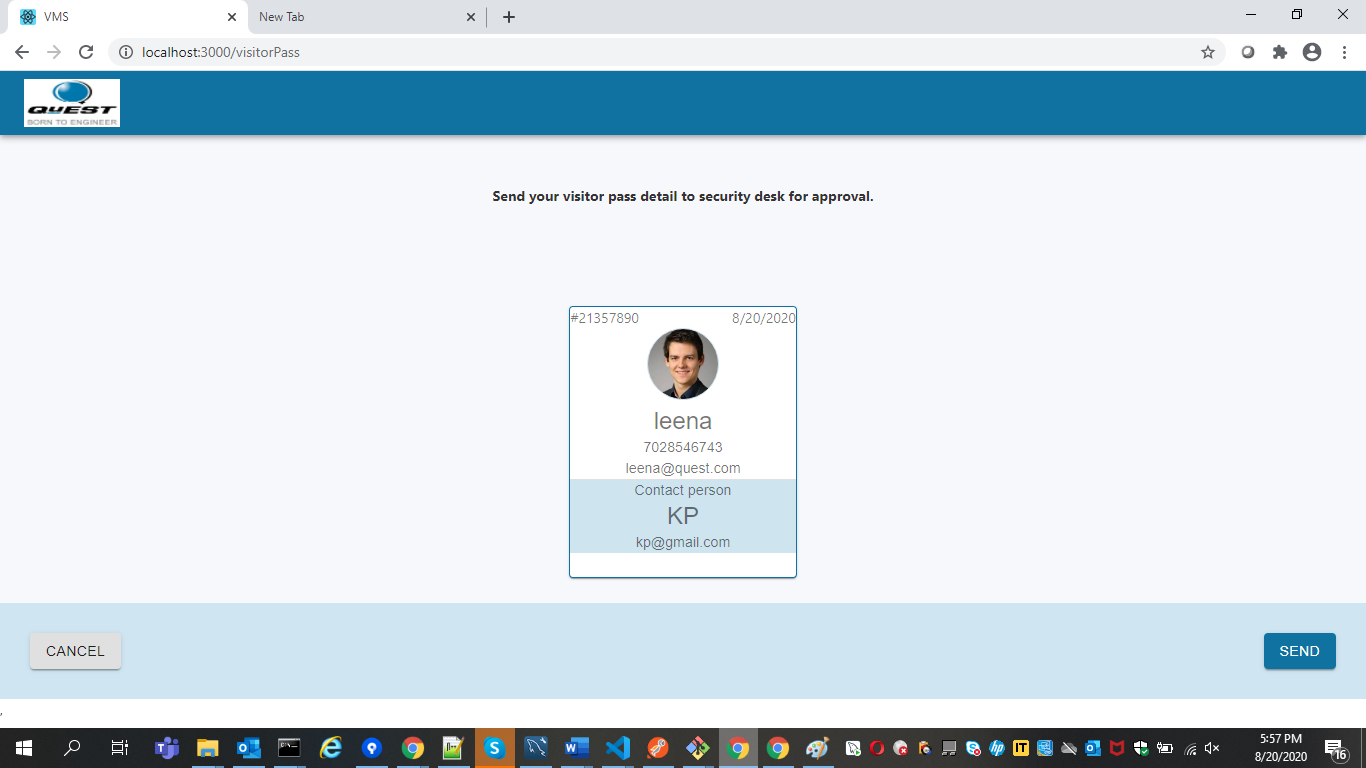
1. **VII Registration page** – visitor need to register himself**.**



1. **Contact Person Details page –** visitor need to enter contact person details here to whom he came to meet.



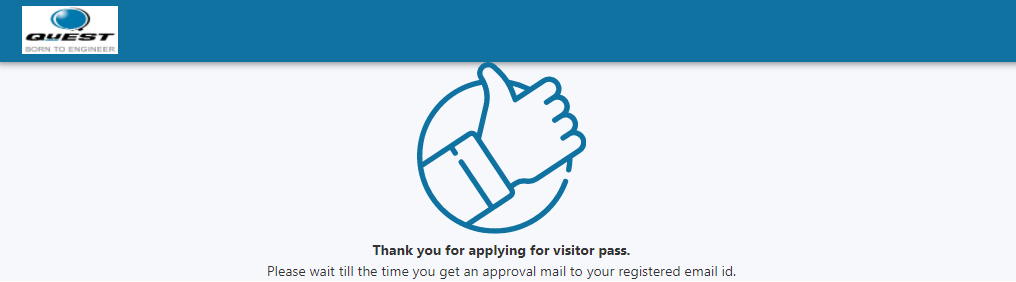
**5.Visitor Pass –** after successful Registration visitor pass is generated.



On clicking of send button the data is submitted, and visitors pass goes for printing.



After taking photo of visitor it redirects to Thank you page.



For VII Redux implementation and API calls it is maintained in VII folder itself. VII is having its own redux implementation having its api called in operations file.