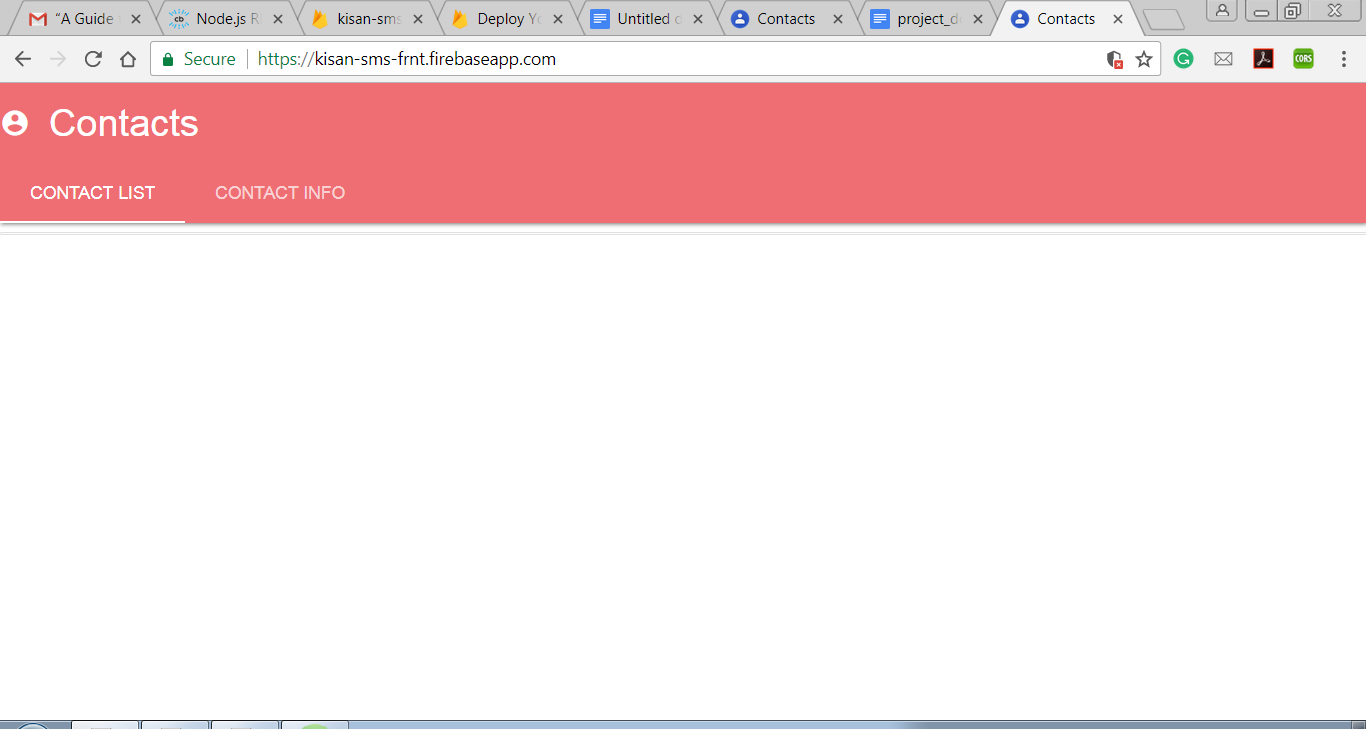
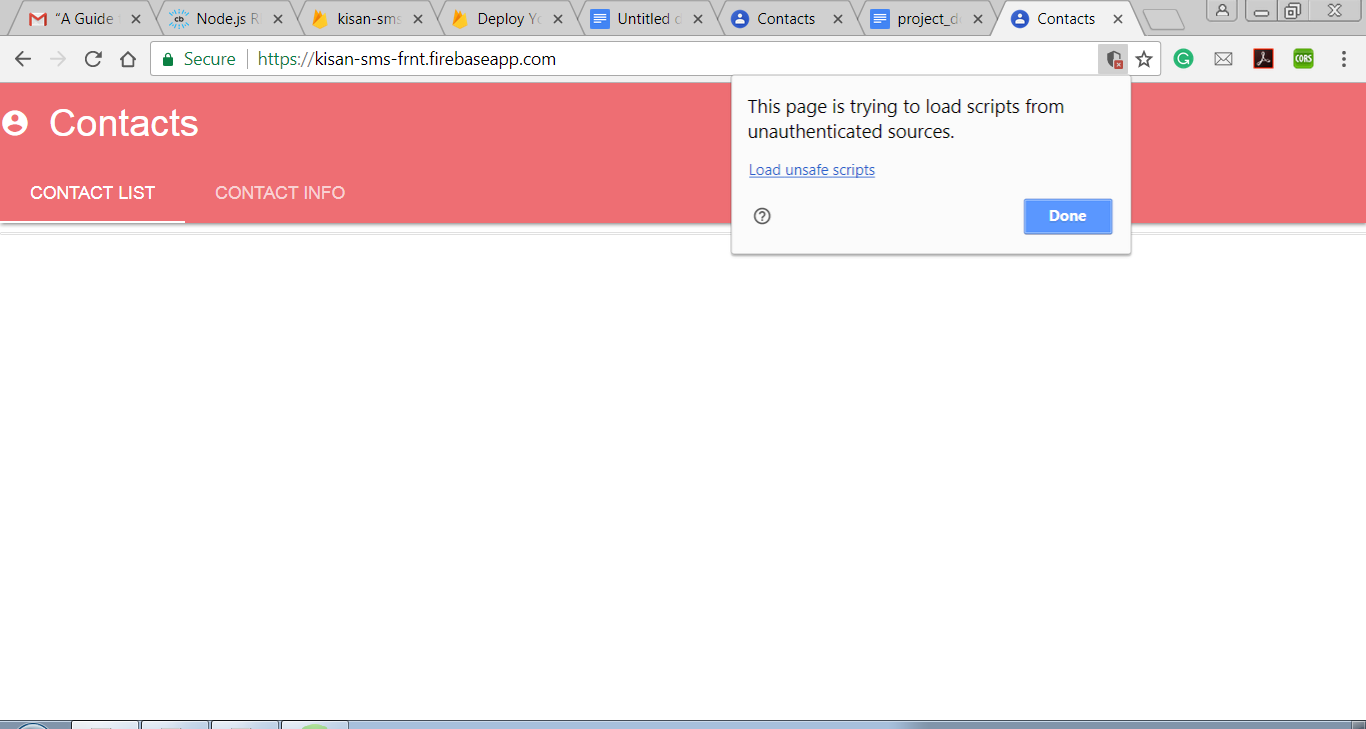
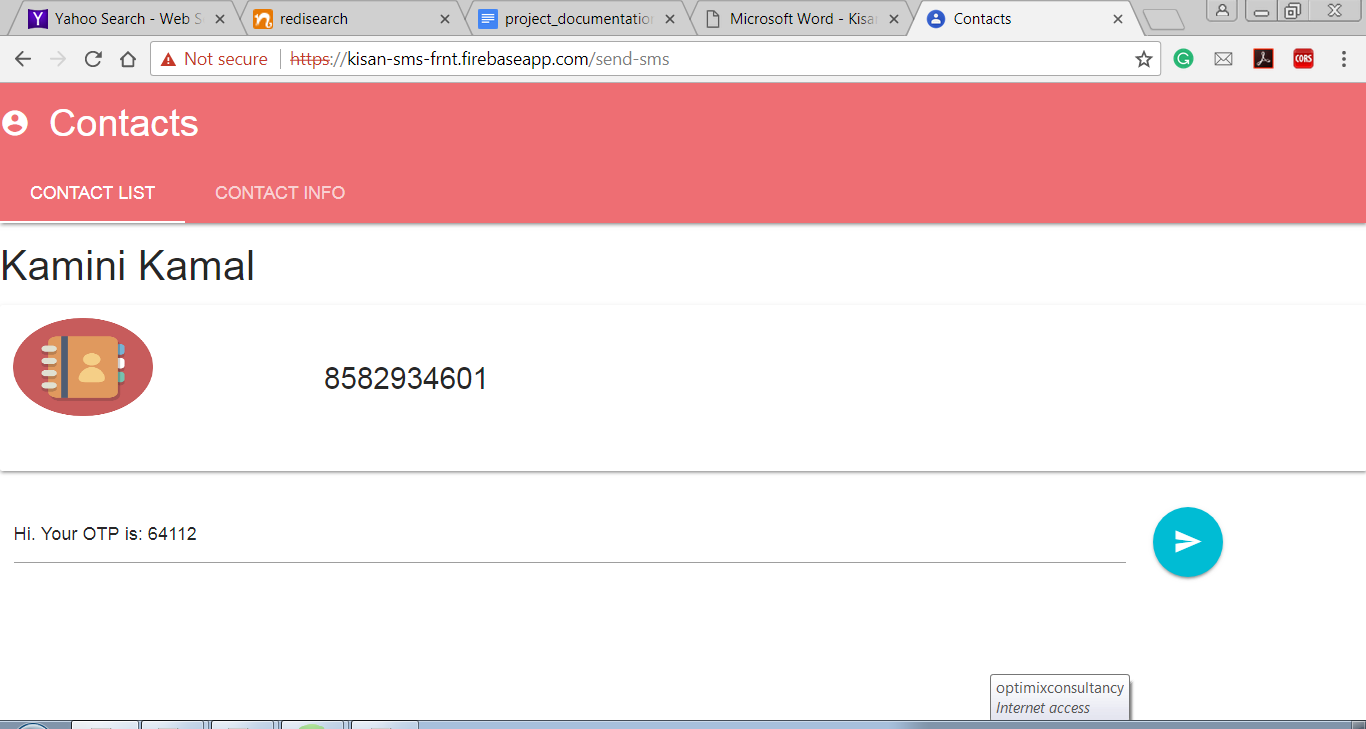
1. Visit the url to see the live demo: <https://kisan-sms-frnt.firebaseapp.com>

We first need to enable the browser to run unsafe scripts. The app makes request to the http server while it has been hosted on https. To enable the access the script :





8. Application features: -

CONTACT LIST : Shows the list of all available contacts. Clicking on the specific list brings the below screen: 

OTP is generated and click on the send button to send SMS to the selected user.

It will display message : error/success showing the status.

CONTACT INFO : shows the list of users receiving the messages

1. The backend and frontend code has been zipped here.
2. Technologies used:
3. ReactJs with redux (frontend)
4. NodeJs (backend)
5. Json Generator for generating JSON data (ex. Contacts list)
6. Twilio for sending sms
7. Firebase for deploying frontend application
8. Heroku for deploying backend application
9. The application is responsive (material design has been used) as well but can be improved to a greater extent.
10. It is a single page application (up to some extend).
11. Improvements:
12. Serviceworkers can be used to convert the application to PWA. In that case, the application can run offline, get push notifications, upgrade itself etc. It can also be made to integrate with the browser cache.
13. Webpack can be used for building better application.
14. Lighthouse extension can be used to analyse the performance of the application.
15. Firebase functions can also be used to reduce the code size and hence our application will occupy less space.
16. While working as a team, bit react can be used to make the components (like headers, container, cards etc. ) reusable.
17. The code needs to be optimized.
18. We can use third party libraries like axios for http request.
19. Limitations of the application
20. Only those numbers which are registered and verified by twilio will receive the sms.
21. The applications needs to enable unsafe script to access the API available on jsonGenerator to access data.
22. ERRORS HANDLED
23. Enabling https server to access resources available on http server.
24. CORS : It can also be handled using chrome extension or can be handled on the server side.
25. State handling becomes tougher when application size increases. So it has been implemented using Redux. It works as a state management library.
26. While deploying the application on any platform, need to modify the code. There may be certain difference between the code on the development and production.