

## App (build using React)

---

You can see the live Kisan-network app - <https://kisan-network-8f226.firebaseio.com/>

Currently having **subscription of only 5 messages** of <https://www.textlocal.in/>

Enable Cross-Origin if browser blocking -

<https://chrome.google.com/webstore/detail/allow-cors-access-control/lhobafahddgcelffkeicbaginigejlf?hl=en>

API Key can be edited to the NewMessagePage component in sendMessageViaAPI method.

## Available Scripts

---

In the project root directory, you can run:

### Extract the Dropbox project

Or you can also git clone - <https://github.com/YadavAnurag/kisan-network>

### Install Node.js

Install Node.js - <https://nodejs.org/en/>

### npm install

Open terminal, go to root directory of the project and run npm install using terminal. Install all dependencies for the project as specified in package.json file.

### npm start

Runs the app in the development mode.

Open <http://localhost:3000> to view it in the browser.

### npm test

Launches the test runner in the interactive watch mode.

### 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.

## App Pages as Routes

---

BASE-PATH (development) - <http://localhost:3000/>

BASE-PATH (production) - <https://kisan-network-b9951.firebaseio.com/>

Home Page - [\\${BASE-PATH}/](#)

Contact Info Page - [\\${BASE-PATH}/info/:id](#)

New Message Page (compose) - [\\${BASE-PATH}/compose/:id](#)

Sent Messages Page - [\\${BASE-PATH}/messages](#)

Not Found Page - [\\${BASE-PATH}/\\${anystring\\_that\\_not\\_match\\_existing\\_path}](#)

## Directory Structure

---

. firebase/	- contain firebase configuration (created by firebase deployment)
build/	- contain production build app
config/	- contain configuration of React project (webpack, test etc.)
node_modules/	- contain all installed for the React project
public/	- contain the project static files

scripts/	- contain list of all aliases that can be used to access React scripts commands in an efficient manner.
server/	- contain express setup script, can be used to host React project
src/	- contain the project code, all dynamic components
. firebase	- contain firebase configuration (created by firebase deployment)
.gitignore	- contain git configuration
firebase.json	- contain firebase configuration (created by firebase deployment)
package.json	- contain overall dependencies and scripts required for the project
README.md	- contain information and documentation of the project

## **src Directory Structure**

src/	- contain the project code, all dynamic components
src/action/	- contain actions methods of the project
src/component/	- contain actions methods of the project
src/fixtures/	- contain static json of contact list (as specified in the Dropbox test link document)
src/reducers/	- contain pure functions that take previous state and action, and returns the next state
src/routers/	- contain routers that keeps React UI sync with the URL
src/selectors/	- contain functions that accepts Redux state as an argument and returns data that is derived from that state
src/store/	- contain plain object that allows components to share state
src/styles/	- contain SASS styles to design nice UI
src/index.js	- contain main entry point of the project for the components

## Deployment

Project is deployed at firebase and link - <https://kisan-network-8f226.firebaseio.com/>

## Key Features

Project developed using React which support Virtual DOM, JSX, One Way Data Binding, Simplicity and give more performance and Smooth user interface.

Front-end Routing implemented using React-Router-DOM.

Front-end data validation.

Responsive user interface designed for various screen resolutions.

A powerful state management tool Redux, used to manage state.

Custom 404, Not found page for incorrect URL resource request used.

Organized project code structure and directory hierarchy.