Team Details

Team name : THE_VETERAN
Project : Event Registration Web App
Team member : 1
Team details :
Name : Parth Chauhan
Collage : DAIICT
Email : 00chauhanparth@gmail.com

Project Overview

Main Technologies :

- **Frontend side :**
  - React.js (frontend library for build use experience) + Hooks
  - Context API (State management)
  - React hooks form (handling forms)

- **Backend side :**
  - Node.js/Express.js (API)
  - MongoDB (Database for storing the use information)
  - Crypto-js (end to end encryption)
  - Node-mailer (sending emails)
  - Joi (object validator)
  - Passport.js (handling admin functionality)
  - Helmet.js (sanitize the API and preventing from various attacks)

Hosting servicer :

- Frontend : https://www.netlify.com/
- Backend : https://www.heroku.com/
- Database : https://www.mongodb.com/cloud/atlas
- Image Storage Bucket : https://firebase.google.com/

Workflow

User Workflow:

- First user should add an email address for getting the form link. On the given email address the user gets the mail regarding the registration link. This link is only valid for 2 hours.if User miss that so the user is again able to fill the email address and able to generate a new link.User is able to try three times to fill up the form after that user's email has been blocked by the system.
- Getting the email TeckStack IT Department are able to send new email/newsletter on that email without able to filling up the form[+1]
- On that given email address if the user clicks the form link within the valid time so first it sends a request to the backend system to check the validity of the token and send the response back to the user.
- After being able to validate on token bases, the user fills the basic information and uploads the IDCard photo. After filling the details the user is able to see/confirm the information without being uploading the photo.
- For handling the user ID card photos I am using a firebase storage bucket. (other services like S3 bucket)
- After confirmation by the user, user information is sent to the backend server,where the server is again validated with the user information with the help of Hapi/Joi validator.
- In this information sending process , all the user data has to be encrypted and sent to the server. Because of data leak and various attacks by hackers. For encryption I am using crypto-js node modules.[+1]

Admin workflow:

- If the IT department starts a server with a fresh database, first you should provide ADMIN_EMAIL on the environment variable. On that admin email system sent mail regarding admin registration(commented code)
- After filling the admin registration form admin is able to login to the system. And admin is able to see the overview where admin is able to see the doughnut chart for registration types and see the details where admin is able to see the detailed table of user information and on the preview admin is able to see all the details of that particular user.

Why am I selecting these technologies? :

I am selecting React because of its unique features like virtual DOM,functional components, react hooks, react context API for managing the application state, fast response and with the help of react I am able to submit my solution. One problem is there with the react is initial it takes time to load all the components bundle file. On this solution I might not need SPA solution because of the above problem. So I am using react-loadable for page level code splitting so it can help me to load faster my react web app at initial time.

Installation Instruction:

```
# Install dependencies for server
yarn

# Install dependencies for client
yarn client-install

# Run the client & server with concurrently
yarn  dev

# Run the Express server only
yarn server

# Run the React client only
yarn client

# Server runs on http://localhost:5000 and client on http://localhost:3000
```

Notes :

- **Current I am not able to send the GitHub links because of GitHub account issues**
- **I am not able to style the webpage because of shortage on time and personal issues**
- **I am not able to deploy this website**

I am really very sorry for not able to submit and deploy a full proof webapp. But it is really fun to build the solution and get the hoods of how to design small software with full proof.