

Implementation of Cloud Based Professional Employee Query System: using Firestore Cloud Functions

¹P. Dastagiraiah, ²SP. Chokkalingam, ³N. Deepa

¹UG Scholar, ²Professor, ³Assistant Professor
Computer Science and Engineering

Saveetha School of Engineering, Saveetha Institute of Medical and Technical Sciences
Chennai, India

¹pdastagiraiah@gmail.com, ²chokkalingam@saveetha.com, ³ndeepa.sse@saveetha.com

Article Info

Volume 83

Page Number: 11590 - 11594

Publication Issue:

March - April 2020

Abstract

Implantaion of cloud based query gives methodologies used to develop a system which provides the good environment, platform [1] for professionals to communicate and discuss their technical problems within their industry people with good security. It highlights questions and replies on neighborhood organization extend in PC programming. The primary objective of this undertaking is to give an increasingly open option in contrast to prior inquiry and answer in stage, for example, specialized issues and programming issues. The fills in as a stage for representatives to ask and answer inquiries, and, through enrollment and dynamic cooperation, to cast a ballot questions and replies up or down and alter questions and replies in a manner like a wiki [4] or Reddit. Clients of this venture can win notoriety focuses and "compensates". Implmentation needs a Fdb (database) setup for the back end side with njs, expr, Cf and other things for back end. And next Mui, Rjs, Re are required for the front end development.

This will resolve the many problems involved in the small scale IT industries.

Keywords: Firebase as Fdb, React.js as Rjs, node.js as njs, Material-UI as Mui, Express as expr, Cloud Functions as cf, Redux as Re

Article History

Article Received: 24 July 2019

Revised: 12 September 2019

Accepted: 15 February 2020

Publication: 16 April 2020

1. Introduction

This implenatation is going to give absolutely basic and overview of the implantation and the entire thing the system do has given below like how it will be and how it works.

Here client or representative will go to stage for making a few inquiries. It features questions [5] furthermore, answers on a wide extent of subjects in programming. It point is to be an inexorably open choice rather than earlier inquiry and answer goals, for instance, specialized issues and programming issues. The site fills in as a phase for agents to ask and answer requests, and, through enlistment and dynamic speculation, to cast a voting form questions and answers up or down and modify questions and answers in a way like a wiki or Reddit.

Clients of this endeavor can win reputation centers. And here one of the most important thing nedded is React. React is essentially js library that was chiefly intended to address the presentation issues in the web application. React utilizes virtual DOM that chooses whether the compo-

nent must be reloaded or not founded on the present condition of the component and the progressions that have happened. And helps to control the data flow in application and it simplifies the execution and make the app stable.

2. System

A. Front End

One of the most important in any applications is front end here platform is created usinf Rjs, Mui, Re which are essential in it. To do so it needs npm environment so install the environment called npm which manges the java script packes in js programming. And one is Rjs (React.js). Creating the app with is easy using create react app command in a directory.

After creating the application run it by using the command called npm start. Then there will be a basic react app with Rjs wecome.

B. Back End

This gives the functionality to the platform as front is the

skeleton for the platform or application. It needs the Fdb, njs, express and cf which are the cloud functions. Fdb is the database which helps to quickly develop applications like ios, web and android. There are two databases here one is realtime and second one is cloud store. Cloud is

used. Functions can be created which will manipulate the database. The function can be accessed by a link provided in cloud like below.

<https://asia-east2-react-firebaecurd.cloudfunctions.net/api>

3. System Architecture

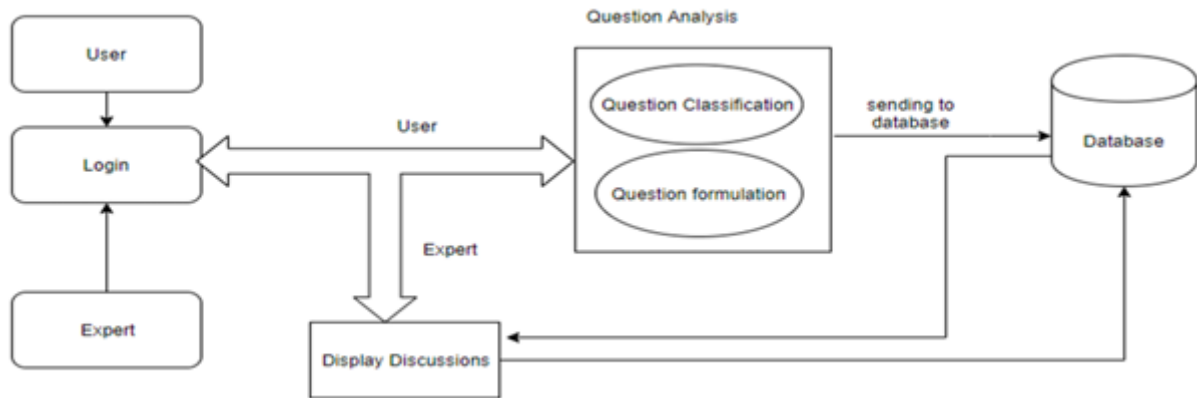


Figure 1: System Architecture

The proposed architecture is having the different components and connections between them will give the overall idea about proposed system.

The components are in architecture (Fig.1)

1. Login
2. User(Employees)
3. Experts(professionals)
4. Question Analysis
5. Database
6. Discussions

Generally every system has its own methodology but here there is no a formal methodology to build it. But

there are some steps involved while development of project

- UI Development
- Functionality
- Database Connectivity
- Input data

A. LOGIN & SIGNUP

As every login & signup has two fields for user_name and password and like confirm password, user who is handling. That can be seen below. (Fig.2 & Fig.3)

B. LOGIN

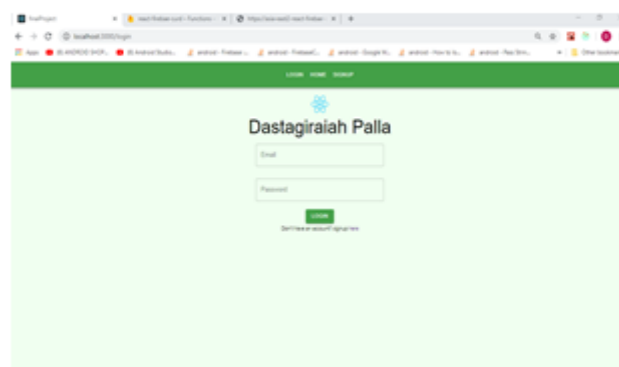


Figure 2: Login Page

C. SIGNUP

Here user can direct from the both login to signup(Fig.3) and signup to login(Fig.2).

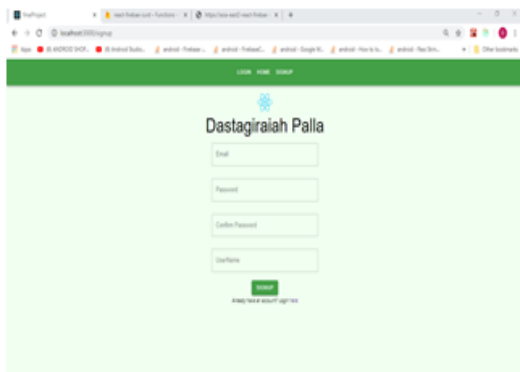


Figure 3: Signup Page

```
<div className={classes0790.form}>
  <img src = '/logo192.png' width={50} height={50} />
  <Typography variant= 'h3' >Dastagirah Palla</Typography>
  <form noValidate onSubmit={this.handleSubmit0790}>
    <TextField id="email" name="email" label="Email" type="Email" variant="outlined"
      className={classes0790.textField}
      value={this.state.email} helperText={errors.email}
      error={errors.email?true:false}
      onChange={this.handleChange0790} fullWidth/>
    <br/>
    <TextField id="password" name="password" label="Password" type="Password" variant="outlined"
      className={classes0790.textField}
      value={this.state.password} helperText={errors.password}
      error={errors.password?true:false}
      onChange={this.handleChange0790} fullWidth />
    <br/>
    {errors.error && (
      <Typography variant="body1" className={classes.customError}>
        {errors.error}</Typography>
      )}
    <Button type='submit' variant="contained" color="primary"
      className={classes0790.button1}>
      Log-
    </Button>
    {loading0790 && <CircularProgress size={30} className={classes0790.progress00}
      color="secondary" />}</Button>
```

```
<br/>
<small>Don't have an account? signup <Link to="/signup">here</Link></small>
</form>
</div>
```

Here is the simple form for submission.

4. Proposed Things

1) Login and Authentication: Clients or representatives can get to the site through sign-in page. Client needs to sign in with g-mail id which is enlisted under the organization he/she is as of now working. If not client can not get to the conversations.

2) Client and master: After login into it either master or client, conversations will be appeared. Here the clients mean the representative who is going to put inquiry or question and master is additionally a worker of the organization however the individual who answer the inquiries is considered as master.

3) Question Analysis: This is for the most part for the examination and grouping of a worker inquiry and to store the information in a decent structure for bringing it from database effectively at whatever point it requires. There will be two sections one is question grouping and question detailing [2].

4) Database: Here the database going to utilize is cloud database and if firebase. The Firebase Realtime Database is a cloud-encouraged database. Data is taken care of as JSON and synchronized in realtime to each related client[9]. Right when you amass cross-organize applications with our iOS, Android, and JavaScript SDKs, the total of your clients share one Realtime Database case and normally get invigorates with the most exceptional data.

Firebase applications remain responsive in any occasion, when detached on the grounds(offline) that the Firebase Realtime Database SDK endures your data to circle. At the point when accessibility is reestablished, the client contraption gets any movements it missed, synchronizing it with the present server state

5) Conversations: Here all conversations will be shown for the clients. There will be numerous highlights like survey all the representatives and offering labels to different workers. It features questions and answers on a wide extent of topics in programming. It point is to be an inexorably open choice rather than earlier inquiry and answer goals. So everybody can see answers, give responds to and poses inquiries. And finally it needs internet [6].

5. Model Output: with Question and answering System

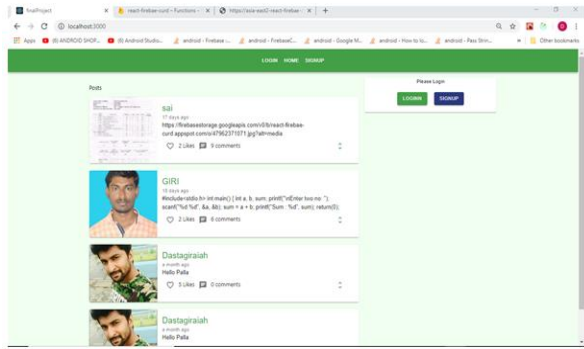


Figure 4: posts page

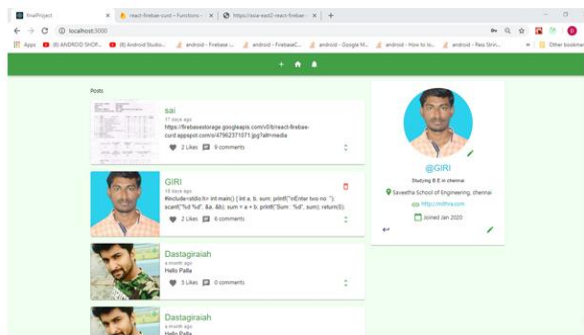


Figure 5: posts after login

- Every can see the questions or queries posted (Fig.4 & Fig.5) on the platform for first but can not do any operations with out logging in there [3].
- After login only user can comment or view entire query as mentioned below.
- So there can see the two buttons like signup and login. If user is already signup, then user can go for login or else continue with the login.

And one more last model demo is commenting or answering as given below.



Figure 6: Comments view

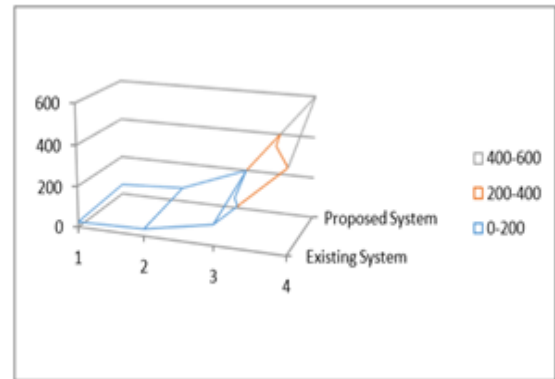


Figure 7: Comparison

6. Conclusion

The main problem in present IT industry [8] can be resolved and this project can be a solution for that. This can conclude that the automation will be the great change in the future industry. If this is included or supported with machine learning then using this, employee area of difficulty can be identified and [7] accuracy (Fig.7) will be increased when there are developing of software product such as performance and quality.

References

- [1] Stack Overflow: A Code Laundering Platform, Ons Mlouki[1], Foutse Khomh[2], and Giuliano Antoniol[3]
- [2] Discovering Value from Community Activity on Focused Question Answering Sites: A Case Study of Stack Overflow, Ashton Anderson[1], Daniel Huttenlocher[2], Jon Kleinberg[3], Jure Leskovec[4].
- [3] S. Fortunato, A. Flammini, F. Menczer, A. Vespignani. Topical interests and the mitigation of search engine bias. Proc. Natl. Acad. Sci. USA, 103(34):12684--12689, 2006.
- [4] J. Leskovec, D. Huttenlocher, J. Kleinberg. Governance in social media: A case study of the Wikipedia promotion process. ICWSM, 2010.
- [5] J. Preece, B. Nonnecke, D. Andrews. The top five reasons for lurking: Improving community experiences for everyone. Computers in Human Behavior, 20(2004).
- [6] J. Ratkiewicz, S. Fortunato, A. Flammini, F. Menczer, Vespignani. Characterizing and modeling the dynamics of online popularity. Phys. Rev. Lett., 105(2010).
- [7] F. Wu and B. A. Huberman. Novelty and collective attention. Proc. Natl. Acad. Sci., 104(45):17599--17601, Nov. 2007.
- [8] B. Vasilescu, V. Filkov, A. Serebrenik, "Stackoverflow and github: Associations between software development and crowdsourced knowledge", Proceedings of the International

- Conference on Social Computing (SocialCom), pp. 188-195, 2013.
- [9] C. K. Roy, J. R. Cordy, R. Koschke, "Comparison and evaluation of code clone detection techniques and tools: A qualitative approach", *Science of Computer Programming*, vol. 74, no. 7, pp. 470-495, 2009.