

Web Based Emergency Directory

T. Deepak Shiva* and R. Revathi†

Department of Computer Science, Karpagam Academy of Higher Education, India

Keywords: React JS, Firebase, Firestore, Node JS, Authentication, Webapp Development, Essential.

Abstract: The Idea is to deliver an online resource that will be helping the individual in an emergency and need something for them at the time. The Web Based Emergency Directory is a web app that will be available for solving the emergency situation by asking the requirements and request from the help needed person. There are few more options available that will be useful for each individual. It's an essential one for quick and easy access to the information for the emergency situation through internet. Every life is a valuable one, each one needs something to live. The information provided in the directory includes the address, phone number, and other relevant details of each emergency service. This paper proposes a web based emergency directory it will be using Node JS, React JS, Firebase. The modern web developing tools like React JS for user friendly UI and Firebase will be used as technologies for making the web app development. These are the well developing and more efficient in the modern web app development. Further Research will be related to development of the web app with additional features adding related to the emergency solving techniques and options for better usability of the web app and to make it work on the mobile devices.

1 INTRODUCTION

React JS is a JavaScript library designed to enhance user interfaces for web applications. It was developed by Facebook and a group of smaller developers who crafted this impressive tool. React JS empowers developers to efficiently build web applications with just a few lines of code, making the process remarkably user-friendly. Custom functions can easily be included through scripts for added functionality. The growing prevalence of the internet has spurred the advancement of the digital world, opening up avenues for web applications to significantly impact individuals worldwide. The increasing popularity of web applications has posed a challenge: the creation of both simple and complex single-page web applications using React JS for their user interfaces. Firebase, a Google-powered software, serves as the backend support for these web applications, ensuring enhanced functionality.

Node JS operates as a server-side scripting language, providing bundled resources and functions to support the development process. The focus in crafting a functional web application lies in

continuous improvement and the introduction of new features, alongside meticulous maintenance to cater to the users' needs.

1.1 Overview

The Web-Based Emergency Directory is an online application designed with unique features specifically tailored for individuals facing emergency situations. It enables users to access help line numbers promptly based on their current circumstances. This emergency directory stands out due to its array of services, which are pivotal in its development. Services such as helpline numbers, health advisories, recovery features, and blood donation options render this web application essential in society. Many individuals in need often struggle to find assistance, and our web application aims to bridge that gap.

Our application requires no installation; it operates seamlessly with active internet access, even on slower connections. It encompasses registration and login features, allowing users to access emergency services without logging in. Preloaded numbers are available, and for registered users,

* Student

† Assistant Professor

additional functionalities like requesting aid and reporting incidents are provided by the Emergency Directory. This interactive approach aims to make the application not just a one-way service but a dynamic and responsive platform.

The web application is built in the working as all the browsers will be able to access the components and view without any error in the functions. Functioning of the web application: The emergency web directory using React JS has more to do in such emergency The functioning of the web application works as shown below:

Figure 1 serves as an illustrative diagram that succinctly explains the operational mechanism of user login functionality and user options, including report, request, and contact details within the web-based emergency directory. In Figure 2, the Admin side module delineates the functions associated with viewing the emergency list from the administrator's perspective. This application comprises various modules such as login.js, signup.js, contact.js, navbar.js, and several others, all integral to the seamless operation of the entire web application. Upon launching the app with an active internet connection, it opens in a web browser. Initially, users are directed to the login page. New users are guided to a signup page to register and gain access to further features. The web-based emergency directory ensures accessibility for non-registered users by providing essential help line numbers. Authentication is achieved through email and password inputs, with Google Firebase serving as the provider for Authentication Functionality. Additionally, Google Firebase Database stores the application's data securely.

The diagram presented in Figure 2 outlines the software, languages, and frameworks employed in developing the web application. Together, these components synergize to ensure the smooth operation and reliability of the web-based emergency directory. They serve as the technical backbone supporting its functionality.

VS Code served as the primary development environment for building the web application. NPM facilitated the installation of essential libraries and node modules required for the React app. JSX, similar to HTML, aids in creating the structural outline, while CSS contributes to the vibrant aesthetics of the web application. The scripts orchestrate the entire functionality of the application.

At the core of the framework lies React JS, a dynamic and constantly evolving framework, enabling greater responsiveness in web applications. Material UI provides a collection of pre-defined,

freely available components that can seamlessly integrate within the code structure.

Google Firebase plays a pivotal role, offering essential authentication services, while Firestore database efficiently stores data from users to administrators. React JS stands out for its continuous development and integration of new features, enhancing the responsiveness and functionality of the web application.



Figure 1.

2 LITERATURE REVIEW

There are very few emergency web directories that have just the contact details of the helping persons. They may be traditional method but in this modern developing society internet plays major role in everybody's daily life.so we can make use of the internet to be used in some emergency situation to help others. Simplicity A very important reason why React is so popular among front end developers is due to its simplicity (Komperla et al., 2022). React focuses in better UI for better interaction. React has more emerging features that's why its been learnt and developed by the developers. Some of these features are its simplicity, its component based nature and its support if JSX (Komperla et al., 2022). Routing is done with React router is the standard routing library used in React (Komperla et al., 2022).

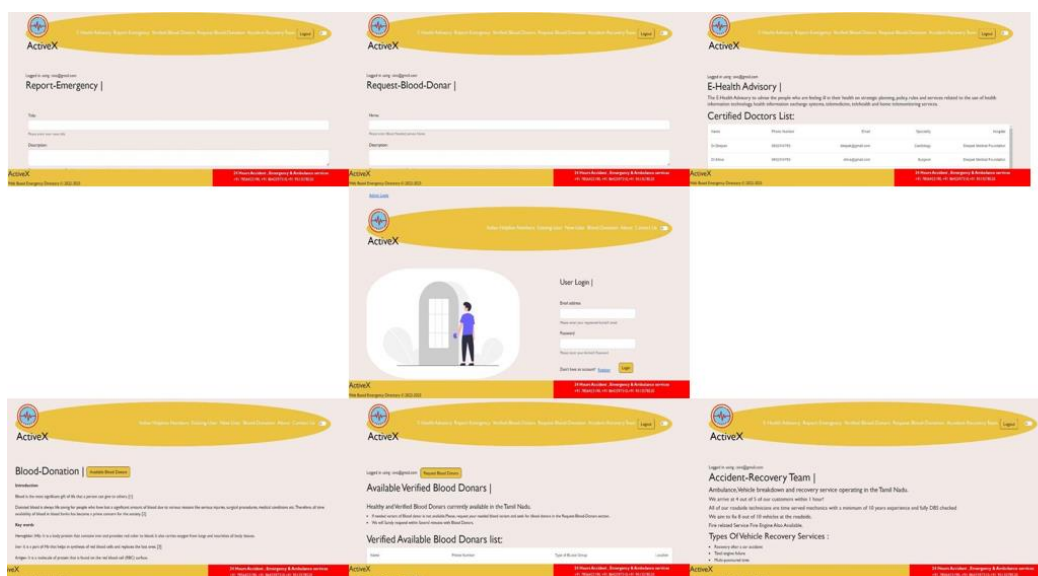


Figure 2: Proposed System with its various components.

The existing systems for web-based directories are not that much effective and not developed with updated technologies. They just contain the information about the hospitals contact information but now in our updated web application we can report and request in the event of emergency in the newly proposed system.

2.1 Concepts of React JS and Firebase

Features of React JS:

Components: The components are the materials that are making up the React application to work. These are the most used and reusable parts like piece of codes of the react web application.

Virtual Dom: Virtual Dom for the JS to compare and update the changes in the UI.

JSX: JSX is extension that allows to code like HTML in the React JS. Read and Write made easier in the use of the JSX.

State: State stores the information or data, The declared data is managed here with certain conditions to be executed at the time of execution of the components render placed in the react app.

Props: This props is the Read only values that from the parent to child component to share data among components.

React lifecycle methods:

render(): It is called during the mount and update phases of the component. For class components, it is a mandatory method. The state cannot be modified within render() (Komperla et al., 2022).

componentDidMount(): It is called after the component has mounted and is ready. The state can be changed within this function. Any API calls are usually initiated within this function (Komperla et al., 2022).

componentDidUpdate(): It is called after the component has updated, usually in response to changes in the “props” or “state” objects. The state can be changed within this function (Komperla et al., 2022).

componentWillUnmount(): It is invoked before the component is unmounted. Clean-up operations like aborting network calls, destroying objects that were created, etc. are usually carried out in this function. Since this is called just before a component is unmounted, the state cannot be modified (Komperla et al., 2022).

Firebase features:

There are plenty of features that are helpful for the web app development and bring rich features in our development.

Firebase Authentication:

There are some security rules and that centrally manage to secure the app that we are building to launch. Such as email/password authentication, Phone number OTP authentication, Google

authentication like methods which can be updated features of authentication in the web app.

Emulator Suite:

We can develop the firebase apps in a separate offline sandbox to develop and test the features we are including to reduce to risk of bricking.

Cloud Firestore:

The cloud services are more useful now a days because of the reach of the services of the internet from anywhere so we need some server for our webapp to keep our data among the users. This can be done and managed well by the cloud firestore.

Cloud Functions:

Serverless framework that is the concept of cloud functions. The events that are initiated and made to work in the environment is done by the firebase by Google servers.

Realtime Database:

The Firebase Real Time Database is a remotely hosted database. JSON data is the type of data synced in real time with cross platforms.

These features are much useful in the easier web app development in the Google's Firebase.

3 PROPOSED SYSTEM

We briefly describe the proposed system as follows:

User Login Module: This section ensures secure access for registered users, employing Google Firebase Authentication for authentication purposes.

User Registration Module: Individuals looking to use the web application can create their unique user IDs through this section.

Indian Help-Line Module: Designed for non-registered users, it provides essential helpline numbers relevant to India.

Emergency Module: Catering to registered users, this section facilitates reporting critical emergencies or requesting urgent assistance.

Request Blood Donation Module: Enables users to report their blood type requirements for emergency situations, connecting those in need with potential donors.

Registered Specialist Doctors Module: Offers access to details of specialist doctors available from various hospitals, providing their contact information.

Blood Donation Information Module: Provides information on blood donation and educates users on available methods of donation.

Contact Details of Verified Blood Donors Module: Displays contact details of verified healthy blood donors for potential recipients.

Accident Recovery Team Module: Allows users to contact various registered recovery vehicles such as ambulances, fire trucks, and tow vehicles for emergency assistance.

Admin Registration Module: Enables the addition of new monitoring admins to manage the web-based emergency directory.

Admin Login Module: Provides login access for admins, ensuring access to administrative functions and data.

Emergency List Module: A monitoring section for admins to view reported emergencies, blood type requirements, and manage critical situations. Access is authenticated through Google's Firebase.

4 DISCUSSION

React (Komperla et al., 2022) was developed with the aim of building large scale applications whose data repeatedly changes with time, and it has addressed this well. React allows for easy creation of interactive UIs, supports JSX, has a component-based structure, and is much faster due to the virtual DOM. The proposed system is enough to be working in the environment to solve most of the problems. The modules will be more easy to be understand by the individual and can be used without any hesitation and immediate response will be available once the user reported or requested anything in the web based emergency directory. There is an idea of making the web application more responsive in smaller screen devices also.

5 CONCLUSION

The Emergency portal is freely available and no registration cost will be applied. This allows us to make the internet more productively and wisely in the emergency situation. The authentication and cloud storage of the data make the usage of the web app more powerful. It has some features that other web directories don't have already available. The only need is the possible internet connection and a browser to access this web based emergency directory. Each individual can be more safely and quickly manage their situation.

Overall, we can make sure that our web based emergency directory by using React JS and Google's Firebase features is worth enough and useful in the time of emergency situation.

6 FUTURE ENHANCEMENT

The React JS is a powerful framework and a developing one. We can make this web application more to develop like IoT, integration with emergency response systems, personalized emergency plans and response sensors. Getting location based alerts about the surroundings situation or warnings. We can integrate the web based emergency directory with smart devices. By also making it more useful through social medias we can share the maximum among the friends and family circle to know about updates. The most interesting this is the Augmented Reality we can also integrate with this augmented reality features, that will allow users to get information more clearly and interactively.

REFERENCES

- Document Vault Using React JS, Dr.D.B.Kadam,Parshv Patil,Mohammadzaid Farque Bhati Dept. of E&TC Engineering, Padmabhooshan Vasantaoada Patil Institute of Technology, Budhgaon, Maharashtra, India, International Journal of Scientific Research in Engineering and Management (IJSREM) Volume: 05 Issue: 10 | Oct - 2021 ISSN: 2582-3930.
- International Journal of Engineering and Management Research e-ISSN: 2250-0758 | p-ISSN: 2394-6962 Volume-12, Issue-5 (October 2022) www.ijemr.net <https://doi.org/10.31033/ijemr.12.5.10> 82 This work is licensed under Creative Commons Attribution 4.0 International License. A Web Based Employee Management System Sanuji Nanayakkara¹, Uditha Ekanayake², Gayesha Subasinghe³, Chamuditha Jayasena⁴, D. I. De Silva⁵ and Dulanji Cooray⁶ 1

An Architectural Style for Single Page Scalable Modern Web Application Suresh Kumar Mukhiya¹ Hoang Khac Hung² 1 (skmu@hvl.no) Ph.D. candidate at Western Norway University of Applied Sciences, Bergen, Norway <https://www.hvl.no> 2(hunghk.it@gmail.com), Ho Chi Minh City University

Indonesian Journal of Electrical Engineering and Computer Science Vol. 26, No. 3, June 2022, pp. 1710~1717 ISSN: 2502-4752, DOI: 10.11591/ijeecs.v26.i3.pp1710-1717 ρ 1710 Journal homepage: <http://ijeecs.iaescore.com>

A detailed survey Varun Komperla, Pratiba Deenadhayalan, Poonam Ghuli, Ramakanthkumar Pattar Department of Computer Science and Engineering, R V College of Engineering, Mysore Rd, RV Vidyaniketan Post, Bengaluru, India