

Reactive Website using React

Mr.Naresh Sammeta¹, Jothi Baskar M², Karthikeyan S³,
Krishna K⁴, Manikandan V⁵

¹Department of CSE/R.M.K. College of Engineering and Technology/Associate Professor
Chennai/Tamil Nadu

²Department of CSE/R.M.K. College of Engineering and Technology/UG Student
Chennai/Tamil Nadu

³Department of CSE/R.M.K. College of Engineering and Technology/UG Student
Chennai/Tamil Nadu

⁴Department of CSE/R.M.K. College of Engineering and Technology/UG Student
Chennai/Tamil Nadu

⁵Department of CSE/R.M.K. College of Engineering and Technology/UG Student
Chennai/Tamil Nadu

Abstract-To create a reactive website using the developing technology such as ReactJs, Node.Js, Express framework and MongoDB . The website is mainly developed for the college purpose .Since every board members, parents, students are coming to know about the college through the website it should be more user-friendly, optimized. Due to the dynamic update of website which is caused by the events that are happening in colleges daily. So we come with a concept allocating a module for each departments and having a coordinator from the respective departments and updating it via the respective portals. The website works through react Library which is built by the Facebook and reduces the loading time and shutoff the creation of bundle.js file again and again.

Keywords: Reactive, Web application, Component, Module, API

I. INTRODUCTION

In the past years I have observed there are two delicate techniques being used by some developers that take a web app from feeling slow and janky to highly reactive and polished. The technique is Reactive web design. It can be used to build sites that always feel fast and responsive to user input regardless of the network speed or latency. As web developers and framework authors, I believe finding ways to make these patterns default in everything we build is a top priority for improving UX and perceived performance on the web.

Website is made as responsive for the purpose of viewing in different screen. Text which are readable should be zoomed and most likely horizontal

scrolling is avoidable since users feel annoying at the time of usage. While clickable buttons are placed in websites user must have a space target to access it. Coming to the part of Back-end whereas all website takes large amount time to load when the values are hardcoded but due to the emergence of Node which has a concept of returning promises asynchronously, thread need not be allocated for each process to be completed. As far as a process is completed the thread responds the server. Express framework used for the purpose of avoiding the hardcoding techniques of End-points. It brings out several of fetching the API data by using fast and light-weight.

RESTful services are used for the purpose of implementing the http protocol operations i.e. CRUD. MongoDB is used for the purpose of storing the data as a documents. Since it is a NoSQL database program we don't have any requirement of tables. The website that can be built with these technology is very lightweight. Moreover we won't get direct access for the DOM. We access the Virtual DOM and it made the Web applications to be created very cheaply. Users that too in present days are not in a thought of waiting for a process to be completed. So by concerning this, the usage of these developing technologies are used and websites made light-weight in which Load is balanced easily by the browsers. Single thread is used in the process of answering all query where are if we go to synchronous call we must use multiple threads and usage of multiple threads increases the server space and this increases the cost of the project.

By concerning this JavaScript give us a better concept of awaiting the response and each response must be decorated by a async function. Server's

response is mounted in the ComponentMount phase of the JSX application so the request of the user are responded when the process is served and the ComponentMount function is decorated with async function. User feels the website very Lightweight and the access of this web application increases the amount of users.

II. LITERATURE REVIEW

This paper gives information regarding Web applications differ from web sites as they have wide range of interactive features/functionality and dynamic content. The content delivered via thin client or server driven architecture vary in size, structure and visual design. To accommodate such rich content care should be taken when designing user interface. Various methodologies exist to deliver the content to the users in an efficient manner. Due to the versatility and diversity of the information to be delivered via web applications, the focus shifts on to user satisfaction. Hence a user experience design aimed at user satisfaction becomes the main focus for such applications. This paper looks into few issues of designing web applications and recommends design criteria to create efficient web applications.[1]. This paper gives us the information regarding web services for deploying automated interactions between distributed and heterogeneous applications, and for connecting business processes. Service mashups indicate a way to create new Web applications by combining existing Web resources utilizing data and Web APIs. They facilitate the design and development of novel and modern Web applications based on easy-to-accomplish end-user service compositions.[2]. This paper gives the information about companies and home users use Web sites offering services ranging from Web sites up to complex Web applications. Often the ergonomics of these applications remains unconsidered and they turn out to be hard to use. In order to examine the usability from within the Web applications, information about the usage of the application is collected. The techniques that have been used in the past for Web 1.0 are no longer adequate. Ajax programs (Web 2.0) are more flexible and require other techniques. This paper shows techniques for collection, analysis, processing and visualization of data for Web 2.0 applications.[3] This paper discusses the advantages of using AngularJS to build the front-end framework, the advantages of using NodeJS to construct the back-end Web server, and the performance advantages of storing data based on MongoDB. This paper focuses on the storage

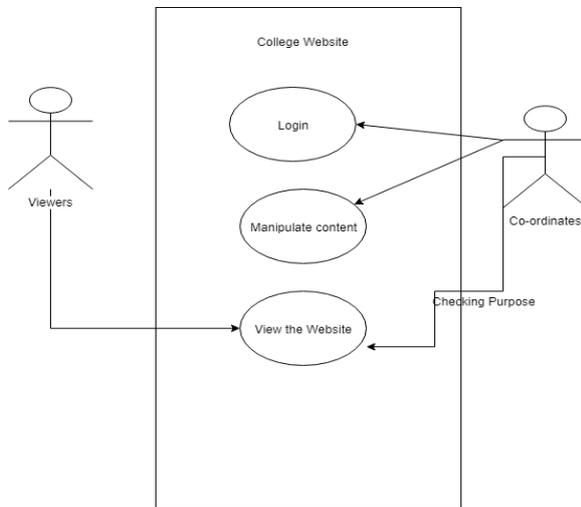
solutions of using MongoDB to store large data and the statistical analysis solutions based on MapReduce. This paper argues on how to build Web services that meet the requirements of large data visualization based on NodeJs.[4]. This paper deals about a large amount of geotagged data coined the term 'Geospatial Big Data', indicating the semi-structured and unstructured nature of such data. SQL relational databases have been known in the past to handle geospatial data very well. However, the abundance of geospatial big data pushed forward the need for NoSQL database which is expected to perform better in terms of handling and storing geospatial big data. This paper discusses the quantitative comparison of performance between the SQL (i.e., PostGIS) and NoSQL (i.e., MongoDB) databases in handling geospatial big data. A NodeJS-based angular-framework web app was developed to test the real-world performance of MongoDB and PostGIS in handling a large amount of simulated geospatial data. A different number of points were generated for testing the geospatial data storing and loading capability of both the databases. The test was conducted by comparing the result of XHR (XML HTTP Request) of both databases in each case. The result showed that NoSQL database, i.e. MongoDB, performs better in loading big geospatial data compared to traditional SQL database using PostGIS.[5].

III. PROPOSED SYSTEM

Each departments are made to maintain a login and so each can update with respective coordinators. Page will be loaded completely with 3sec by the usage of CDN's. Static website will be converted to Dynamic website .Page will be made responsive which can be viewed in mobile phone too.

A. Responsive:

The work has been made in an elegant manner where our page has been made responsive by undergoing various changes and through various factors. Considering the concept of size of an image, text ,menu, panel, table always we used to represent either in pixels,em or by rem and it has been replaced with percentage(%) and viewport width(vw) . Pages are changed Grid system of CSS and bootstrap for table's navigation like that. Flex concept gave us a big opportunity to move the content left, right, top (or) bottom which replaced the traditional method of margin or padding and which is most annoying process in styling.



B. Reducing Load time:

This is major factor from which a website is validated .Loading time is calculated from the network tab of chrome developer tools .It depends upon how much bytes transferred for the purpose of loading a site . Usage of React router and map () method saved the cost of creating bundle.js file and creating a large amount of div’s for the purpose of tables. Object oriented Programming gave us a better concept having the values as objects and fetching it through the map method. Submitting a form plays a major role in which always it creates a bundle.js file and make the page to load even for wrong credentials. Usage of e.preventDefault() method eradicated the loading of bundle.js again and again .

- Load time => 1.00 sec
- Document Complete => 3.00 sec
- Fully Loaded => 5.00 sec

CModule for Each Department:

A college will be containing more than 3 departments for sure. Some of the colleges which is most popular will be conducting events daily, trainings may happen, career guidance will be given and so these things are happening a daily based routine. For the purpose of updating dynamically and routine-wise we came to a conclusion of creating a module for each department in which the respective co-ordinates from each department update the content by their own given credentials from the portal.

D. Component Based:

Ever module developed is component based by the usage of library known as React. Contrast to the PHP for adding a header (or) footer we embed the header.php file which might create many http request and increase load time of webpage which will make users annoying. So, by the concept of react we just mount the component name either functional or statefull component and it won’t create http request like how PHP does.

IV. ALGORITHM

A. Lodash Library:

Lodash is an JavaScript library which helps programmers to maintain the js code in easier way. It provides many functionalities like string functions, array functions, conversion methods and much more inbuilt methods. It also simplifies mathematical functions and works fine with different datatypes. It helps with easy methods for sorting, filtering and wrapping contents. It is mainly used for its iterating functionality which helps many dynamic websites to load much larger contents without larger lines of codes.

B. Prop-types in React:

Prop-types are used in react to validate the type of props we send and receive. The basic props are parameters or values which we send from one component and receive it in another component. These passing values are the actual data which we need in another component to access or modify or display it in interface. These data are in different formats like strings, arrays, Booleans, etc., So to identify and validate the type of props we receive, we uses Prop-types in react components. This is an inbuilt function which react provides for us to make the validation.

So, components will now know all the props and the respective types required by it. React would throw an error(which can be viewed on the console), when an incorrect datatype is passed to the component. This will lead us to know where the error and what’s the error. So, developers can make their work little easier and feel better while programming with react.js. Some example codes of prop-types are given below:

```
Array: PropTypes.array
Bool: PropTypes.bool
```

Function: PropTypes.func
 Number: PropTypes.number
 Object: PropTypes.object
 String: PropTypes.string
 Symbol: PropTypes.symbol

To identify the node type we use the following:

- Node: PropTypes.node,

To identify the element in each component we use this:

- Element: PropTypes.element
- Element_Type: PropTypes.elementType

You can also declare that a prop is an instance of a class. This uses JS's instanceof operator.

- Message: PropTypes.instanceOf(Message)

Like this Prop-type play a major role in validating the props in components of react codes.

C. React-Route:

React-Route is generally used when your front-end app has multiple screens or 'pages' that must also have a unique sharable URL.

Much like a back-end rendered app, when you click a link it navigates to a new page with a URL and if you were to give that URL to a friend, your friend could load that URL to view the same content. React Router enables this same functionality but completely on the front-end.

D.Reconciliation

React provides a declarative API so that you don't have to worry about exactly what changes on every update. This makes writing applications a lot easier, but it might not be obvious how this is implemented within React. This article explains the choices we made in React's "diffing" algorithm so that component updates are predictable while being fast enough for high-performance apps.

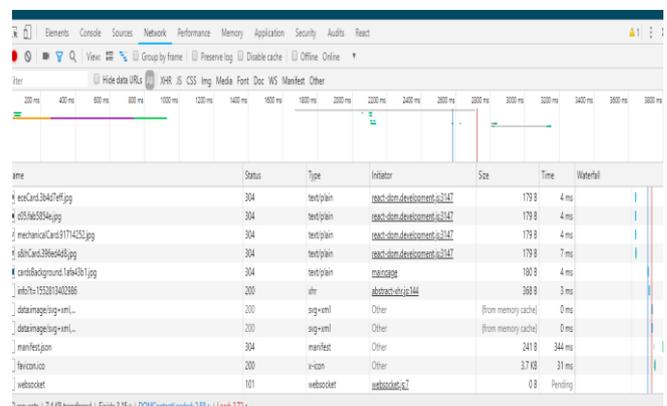
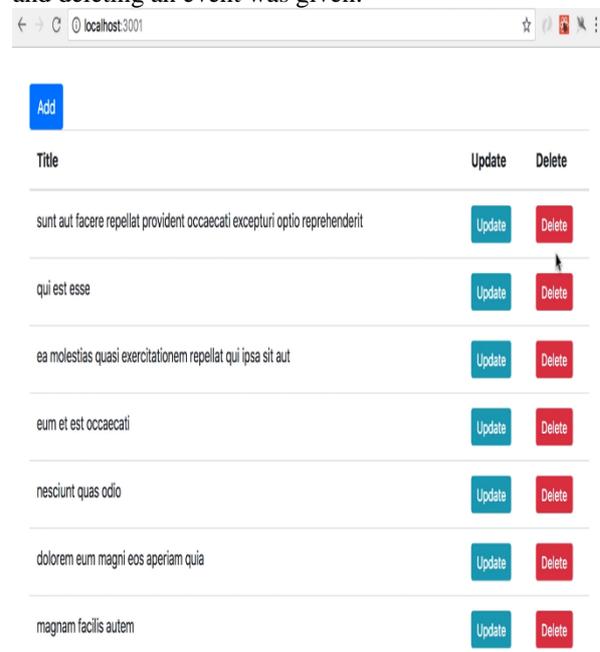
E. Diffing Algorithm:

Whenever the root elements have different types, React will tear down the old tree and build the new tree from scratch. Going from <a> to , or from <Article> to <Comment>, or

from <Button> to <div> - any of those will lead to a full rebuild. When tearing down a tree, old DOM nodes are destroyed. Component instance receive component Will Unmount(). When building up a new tree, new DOM nodes are inserted into the DOM. Component instance receive component Will Mount() and then component Did Mount(). Any state associated with old tree is lost.

V. RESULT AND DISCUSSION

From the networks tab we can know the the bytes and request transferred and made are very low compared to the normal application and the options of updating and deleting an event was given:



VI. CONCLUSION

In this paper, we have presented a reactive webpage system for the students and college which helps them in accessing the website at faster time than before. Also, we used react js to make the site more user responsive and the unwanted loading will won't happen anymore. This site will be more user friendly and UI design will be more responsive and attractive.

VII. FUTURE ENHANCEMENT

The college details are more important and should not be disclosed to anyone. No one should have any access to the details of the student. So, the database which stores the college details and student details should be encrypted. Even if someone accesses the database without the knowledge of the college authorities, only the encrypted data should be shown to him.

To reduce the difficulty of applying leaves, paying fees, seeking some permissions from principal, hod's and staffs we make a interface in which students can do every request to their respective authorities.

REFERENCES

- [1] Jay kiruthuka, SouheilKhadakDarre; Greenhill, JarekFrancik© 2016 IEEE . " User Experience Design in Web Applications".
- [2] Diamal Benslimane, SchahramDustdar © 2008 IEEE. " Services Mashups: The New Generation of Web Applications".
- [3]Ludger Martin © 2008 IEEE "Usability analysis and visualization of Web 2.0 applications"
- [4] Li Liang,Lign ZhuWenqianShanq, DongyuFen,ZidaXioa© 2017 IEEE "Express supervision system based on NodeJS and MongoDB".
- [5] Dany Laksono,© 2018 IEEE"Testing Spatial Data Deliverance in SQL and NoSQL Database Using NodeJS Fullstack Web App"