JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

Empowering Scalable and Real-Time Ticketing Platforms with React.js

Author: Ashish Sharma,

Dr. Vishal Shrivastava, Dr. Akhil Pandey, Er. Aarti Sharma

Department: Computer Science Engineering

Abstract: In this research I am going to explore the utilization of the NodeJs in creating and developing a responsive website for the purpose of booking tickets of different events in real time.
The aim of this research paper is to bring to light how Nodejs is helpful in solving various challanges in the creation of an online ticket booking platform. How it helps in creating a smooth user interface for the ticket booking platform.

1. Introduction:

React.js is a javaScript library for the development of frontend in web application development. It was developed and is maintained by Meta (Facebook). It can work on the basis of components. It makes intersecting different pages of a website very easy. Because of its component based architecture it makes work of developers very easy to maintain different states in a website/

2. Foundation of React.js:

States are the foundation of the React.js. States are nothing but the representation of data that changes with time and can easily affect the behavior of different components of the web application. React.js is mostly used because it makes it very easy

3. Challenges in development of ticketing platform :

3.1. **Component Hierarchies** :

A ticketing platform is a very complex web application. For developing an effective and user friendly ticketing platform it will take more than one component in the web application like home page, event page, payment dashboard, admin dashboard etc.
All these components are interdependent on each other and will create a complex hierarchy. So this

complex hierarchy of components makes the work of development hard and rough.

3.2. **State Management** :

States are nothing but the representation of data in websites that can change with the time. It may be easy to manage states if there are fewer components but with the high number of components it becomes very complex to manage states in ticket booking platforms. Mismanagement of state of any one component can affect and compromise the whole website.

3.3. **Integration with Backend Technologies**:

Integration with backend technologies with the frontend can also be a challenge in development of ticketing platforms. The process of fetching data should be synchronous and there should not be much delay in the fetching process.

3.4. Responsiveness of ticketing platform:

In the present time the number of users that use their small screen devices like mobile phones to open a website is increasing by a high number. And developers should make sure that all user experiences are the same while exploring the ticketing platform.
To make sure the same experience of the user website should be responsive and the developer should use a frontend technology that helps in achieving the responsiveness of the website.

3.4.Security:

A ticketing platform will include a payment gateway. So the security of the user's money automatically becomes a very big challenge in the development process. And the front-end library should not compromise security in any condition.

4. Solution for a Ticketing Platform with React.js:

4.1. Component and State management:

Rect.js is based on component based architecture. All the components in react.js have their own state object. This state object helps all components to create, manage and manipulate its own data that helps in the management of the components of web applications. And all components are responsible to manage their state objects.

4.2. Real-Time Interaction:

The virtual DOM with the efficient rendering helps react.js to achieve the real time interaction of the front end with the backend technology and makes the fetching process of data synchronous in web application.

4.3. Security :

For the security in the ticket booking platform react.js can be used as input validation and sensitization on the client side to make sure that the entered data is not malicious. Other than this, react.js can also be used to add different security tokens like JWT in the platform for the authorization of the user.

4.4. Responsiveness of the ticket booking platform:

React.js enables responsiveness of websites though its modular component architecture. React.js supports css-in-javascript and also uses media queries. These features help developers to create a responsive website with use of react.js that provides the same experience to users with the different screen size of the device.

5. Methodology for Developing a Ticketing Platform with React.js:

5.1. Data Collection:

1. Market research : In the process of market research we did a survey of users to find the frontend that needed to be added using react.js in our website. We also studied the existing ticketing platforms and observed their strengths and weaknesses to improve our website.

2. Technology assessment: Technology is the crucial part of our ticketing platform and the perfect technology should be chosen to create a perfect ticketing

platform. That is why other than react.js for front end, node.js and mongodb are used for backend and database respectively. So MERN stack is used for the development of the ticketing platform.

5.2. Technology Integration:

1. React.js selection :First React.js was selected as the javascript front end library. It was selected because it is one the most popular libraries of javascript and many developers use it so there are a very high number of online documentation for react.js.

2. Integration of WebSocket for real-time updates: Websocket was integrated in the Nodejs to achieve the much needed real time updates.

3. Implementation of security protocols.: The security protocols were properly implemented to make sure that the data remains safe .

5.3. Testing and Optimization:

1. Functional testing of ticketing features : All the relevant features as well as functionalities were property tested so that the users get the maximum satisfaction while using the web application.

2. Cross-browser and device testing : The functions were checked using various devices .

3. User testing for feedback and iterative improvements: The website was tested by other real life testers to get the feedback on the performance and usability of the different methods and functions of the website.

6. Case Study: Real-Time Communication in CampFest Ticketing

Platform

6.1 Background:

There was a need for enhancing the real time communication capabilities of my event booking site. The main aim of it was to streamline the communication between the event organizers and the attendees.

6.2 Objectives:

1. Improved user experience through efficient communication.

2. Streamlined event management with real-time updates.

6.3 Implementation:

1. Selection of a communication solution compatible with React.js : We selected the WebSocket based methods to achieve the communication needs . Socket .io and other such libraries were considered to make the communication process more streamlined.

2. Seamless integration within the CampFest platform : The communication methods were properly integrated within the booking app that is CampFest.

7. Limitations of Developing a Ticketing Platform with React.js:

1. **Evolving Technology Stack**: React js is a dynamic library and all its associated libraries are all continuously evolving . At such time we need to continuously make changes so as to provide the app with the most optimal piece of code.

2. **Web Application Environment:** The efficacy and relevance of the chosen stack may vary over time with the advancement in the technology . To adapt there might be certain changes needed to be done which can affect the performance and integrity of the website.

3. **Security and Privacy Considerations:** Since there was a need for a payments gateway the security need was thus much higher as well. There was a high need to keep the data protected from various security attacks and breaches. Authentication and Authorization would play a key role in achieving security.

8. Conclusion for Developing using Node.js:

Through the creation of this app I came to know about the different aspects involved in developing a website using React js. I came across various problems and learned how to solve those as well. There were certain concepts which were very easy to implement using React js and also those which were kind of difficult to implement .

There were multiple steps and situations to getting to the final stage of creating the web-app that is the event booking platform such as the empowering of the platform using React js, addressing the challenges involved in making , optimizing the platform for the different users and working on it to make it secure for everyone.

References for Developing a Ticketing Platform with Node.js:

i. “https://reactjs.org/docs/latest/api/”

ii. “https://www.w3schools.com/reactjs/”

iii. Robin Weiruch(2017). "The road to learn reactjs".

iv. Mark Thomas(2018). "Reactjs in action”.