



---

## **Research and Analysis of Frontend Development Framework for Web Developing E-commerce Platform for People in Remote Areas**

<sup>1</sup> *Bhay Yadav*

**B. Tech Scholar, Computer Science (Engineering) Department,**

Email:- [yabhay232ay117585@gmail.com](mailto:yabhay232ay117585@gmail.com)

---

### **ABSTRACT**

react.js plays an important role in front-end development and gives developers new options to create new applications. This article explains how react.js helps create user interfaces and its benefits in front-end development. According to research by Web Technology Surveys, the majority of all websites today use React.js. It would not be an exaggeration to say that React.JS is used everywhere. New visitors have different interests when it comes to the web. This article discusses the key elements of the framework, including its advantages over competitive models, how it works, and its architecture.

---

### **INTRODUCTION**

Nowadays, web developers often call the products they create web applications rather than web pages. Although there is no difference between the two, web usage is generally more interactive and dynamic. It allows users to perform actions and receive responses to those actions. Business front-end development has never been this interesting and challenging.

New applications, libraries, frameworks and plugins are released every day.

There is so much information to absorb. Grab's online team is up to date on best practice recommendations and therefore our web applications now use a modern JavaScript environment (Mandapuram, 2016). In the traditional model, the browser is responsible for rendering HTML after receiving it from the server. When a user accesses a new URL, needs to complete a new page and the server sends a new HTML page to the user.

On the other hand, SPA no longer uses server-side processing, but client-side processing. The first page of the application is loaded into the browser from the server, along with scripts (base, libraries or application code) and documentation required for all applications. When the user visits a different page, no action is taken to update the current page. The HTML5 History API makes the necessary changes to the URL of the page. The browser sends an AJAX query to the server to fetch the new data required for the new page. Usually this data is stored in JSON format. SPA then uses JavaScript to refresh the page with the information previously retrieved when the page was first loaded. The architecture works similar to traditional mobile applications. The advantage is that the application is more responsive and prevents the user from flickering on page navigation caused by the new full page. The server receives fewer HTTP requests as there is no need to download the same resource multiple times each time the page loads (Desamsetti and Mandapuram, 2017). Clear separation of responsibilities between the client and the server; it is simple to construct new clients for various platforms (such as mobile, chatbots, and smartwatches) without having to rewrite the code that runs on the server. Altering the technological stack individually on the client and server is also possible, provided the API contract is not violated (Mandapuram, 2017a).

Initial page load time is longer since the framework, app code, and assets needed for many pages must be loaded simultaneously. Configuring our server to direct all requests to a single entry point and then letting client-side routing take over from there is an additional step that has to be completed on our server. This can be done by following the instructions in the following section. The rendering of content in SPAs depends on JavaScript, but not all search engines can run JavaScript when they scan websites; therefore, search engines may interpret the material on our page as empty (Gutlapalli, 2017a). This may affect your app's SEO (Search Engine Optimization). Clear client-server separation scales well with large engineering teams because client and server code can be developed and deployed independently. This is because classic server-side renderers are still a viable option.

This is especially true for Grab because we often have multiple user applications accessing the same API server at the same time (Mandapuram, 2017b). As web developers choose to build applications rather than pages, the style of client-side JavaScript becomes important. It is common practice to add jQuery code to every web page displayed server-side to enable user interaction. However, when creating large applications jQuery alone is not enough. Finally, jQuery is not a framework but a library primarily for manipulating the Document Model (DOM); but it is a library for managing the Document Model (DOM). Therefore it does not create a special structure and organization for our application.

Functions written in JavaScript are designed to provide a higher level understanding of the DOM (Gutlapalli, 2017b). These abstractions make it possible to store state.

---

## USER INTERFACE – REACT

React is a JavaScript project that has dominated the front-end ecosystem lately. Facebook Smart people create and open source React. Developers write and author React Web interface elements. React introduces radical concepts and requires developers to rethink best practices (Capala and Skubblewska-Paszowska, 2018). For years, web developers have been taught how to create HTML, JavaScript, and CSS separately. React recommends using JavaScript to write HTML and CSS. After testing this seems less critical. Because the development front is moving towards material-based development. Features of React:

**Report:** We want to see something, not how to do it. With jQuery, developers need to modify the DOM to change application state. In React, we change the state of a component and the view will update itself according to the state. The render() tag makes it easy to predict how things will look. A pure support case function, the idea is functional. In general, React components are defined by props (external parameters) and state (internal data). The same stimuli and situations produce the same emotions. It is easy to try to work well and work pure. React's well-defined interface makes testing easy. We can test this by passing members to different objects and states and comparing the output.

**Sustainability:** Contextual thinking encourages reuse. When defining prop types, React code is self-written so readers know what to use. Finally, our thoughts and emotions are contained within the product and should not affect other functions. If the product has the same hardware, large-scale refactoring is not a big deal.

**Excellent performance:** React uses virtual DOM (not shadow DOM) and redraws the same content when the state changes. Why do you need virtual DOM? Modern JavaScript engines are fast, but reading and writing the DOM is slow. React keeps lightweight DOM virtualization in memory. It is wrong to recreate everything. React re-renders the DOM in memory, not the actual DOM. When data is modified, a new virtual representation is created and compared to the previous model.

Then adjust the browser's DOM with a small change.

React API surface is smaller; There are fewer APIs to understand and they rarely change.

The React community is one of the largest. It has great tools, open source UI products, and great online resources to learn React. Many React development tools improve the developer experience. The React development tool allows us to search, view and change the connection and state in the browser (Ciliberti, 2017). Using webpack hot reload allows us to see code changes in the browser without support. Front-end development requires editing code, saving and restoring browsers. Hot reload helps remove the last step. Facebook provides coding scripts to migrate code to the new API after library changes. This makes installation easier. The Facebook team deserves praise for their work on React development.

Over time, tracking frameworks have been created that outperform React. React may be a complex library, but its ecosystem, usability, and benefits are phenomenal. Facebook is also rewriting the integration process to make React faster (Madaj, 2018). React teaches us

to create better code, manage web projects, and become better engineers. We love it. To learn about React, see the tic-tac-toe tutorial on the React homepage. Highly rated free React Fundamentals course for.

Deep learning from React experts, developer of React Router. Complex concepts not provided in the React documentation are also covered (Naiki et al., 2018). Facebook's Create React App creates React projects with minimal configuration, making it perfect for new React projects.

---

## E-commerce

E-commerce, often referred to as digital business, is the process of buying and selling goods on the internet, as income and information are required to carry out the entire business.

E-commerce was not very well known in the beginning, but with the increase in mobile phone usage, many people began to show interest in online shopping, and it became more and more popular. E-commerce is coming in very good form. E-commerce business models are generally divided into 4 categories. 1.

**Marketing to customers.** This is an Internet business model that allows us to sell products to end customers. 2. **B2B (Business to Business)** is a type of business that takes place between large companies, organizations and businesses; Nowadays, most e-commerce falls into this category. 3. **B2C (Consumer to Business)** Individuals can use this e-commerce to sell their products to businesses.

People will ensure that tasks are completed within a certain period of time through the website or other e-commerce media. In this type of e-commerce, customers can create their own brands for their businesses. Freelancing is a good example of this type of work. 4. **C2C (Consumer to Consumer)**

This e-commerce business connects with consumers, exchanges goods and makes money through business that supports buyers and consumers.