PATNA WOMEN'S COLLEGE AUTONOMOUS 1940

Explore – Journal of Research
Peer Reviewed Journal
ISSN 2278–0297 (Print)
ISSN 2278–6414 (Online)
Vol. XIV No. 2, 2022

https://patnawomenscollege.in/explore-journal-of-research/

Standard comparison of the front JavaScript frameworks Single page application React & Angular predicated on its, Performance, and Efficiency

• **Rishabh Manglam** • **Dheeraj Kumar**

Received : April 2022
Accepted : May 2022
Corresponding Author : Rishabh Manglam

Abstract: Netscape Communications Corporation introduced JavaScript on December 4, 1995. We can use JavaScript to create and control dynamic website content that moves, refreshes, or otherwise changes on our screen without having to manually reload a web page. In recent years, the field of web development has evolved significantly. This thesis delves at the most popular JavaScript frameworks nowadays, React Js and Angular. The study begins with a history and evolution of JavaScript to provide background for the study of various frameworks. It is also characterised in terms of its present state and versions. The key aspects of front-end JavaScript frameworks are then examined, as well as the development setup procedure for a generic framework. For each framework, a typical JavaScript application is specified and generated. Several specific benchmarks are run with this reference application, in which frameworks are graded based on how well they handle certain DOM data changes. Each of them will eventually be investigated and rated based on pre-determined criteria. Finally, a suggestion will be made regarding which frameworks are most suited for usage, as well as how the future landscape of front-end JavaScript frameworks is likely to evolve.

Keywords: *JavaScript, React Js & Angular, Comparison.*

Rishabh Manglam
MCA Student,
Affiliation: Amity University, Patna (India)
Email-id: rishabhmanglam25@gmail.com

Dheeraj Kumar
Assistant Professor,
AIIT, Amity University, Patna
Email-id: dkumar3@ptn.amity.edu

Introduction:

JavaScript is a client-side scripting language used to add dynamic properties and functionality to HTML websites. JavaScript was created to help browsers with asynchronous communication, to control browsers, and to allow users to interact with web page components.

An HTML page is nothing more than a static page. JavaScript may make a website more interactive and user-friendly by allowing for quick navigation and providing interactivity. Many frameworks have been built based on JavaScript, and many new functionalities for server-side scripting have been added. JavaScript has made significant inroads into the online business, and there are no web technologies that do not use JavaScript.

JavaScript is a dynamic scripting language that instructs after a page has been loaded, the browser can make modifications to its page components. JavaScript is simple to learn: all you need is a browser and a text editor to start creating and running code.

For many years, the JavaScript library of React Js and the Angular framework have been the two most popular front-end frameworks in web development. Both are excellent web technologies in terms of both technicality and commercial strategy. Vue.js is only one example of a JavaScript library and framework that has its own set of strengths. However, there are considerably more React and Angular developers on the market than any other JS option. That, as well as the fact that both are backed by significant companies and have active communities.

I. JavaScript Frameworks

i. React History

Facebook created React Js, as it is often known. In 2013, Facebook made react available as an open-source JS library. A huge, active OS community also lends its assistance. React is a JavaScript library that is open source and used for frontend development. It is used to create user interface or UI components. Its component-based and declarative characteristics make it simple for developers to construct interactive and sophisticated user interfaces. Because of the "learn once, write anywhere" philosophy, developers can create rapid and scalable programmes for various platforms. Facebook and a network of individual developers and communities handle React.

The most recent React updates, from v17 to v17.0.2, have prioritised backwards compatibility, making it safer and easier to embed a tree handled by one version of React within another.

ii. Angular History

Angular, developed by Google and released in 2010, is the oldest and perhaps most mature of the JavaScript frameworks and libraries that have impacted front-end development in the decade and counting thereafter. Angular is a typescript-based programming framework. It's a component-based framework for creating scalable web applications. It includes a slew of well-integrated libraries and capabilities including client-server communication, routing, and more. It includes a set of developer tools for creating and scaling projects ranging from single-developer to enterprise-grade systems. Furthermore, it is regularly updated technology, with the newest innovations spearheaded by Google's Angular team. Typescript-based Angular has undergone multiple variations, with one significant change occurring between Angular 2 and Angular 3 (the most recent of which was released in early 2022).

II. Comparison on the basis of Flexibility and Performance

i. Framework vs. Library

There are several differences between Angular and React. One distinction is that Angular is a full-fledged MVC framework, whereas React is only a JavaScript library (just

the view). Allow me to elaborate. Angular is classified as a framework since it has strong views on how your application should be constructed. It also has a lot more "out-of-the-box" capabilities. You don't have to think about the routing libraries to use or other similar details — you can just start coding. The disadvantage is that you have less freedom - you must utilise what Angular gives.

Angular comes with the following features "out of the box": Templates, based on an extended version of HTML

- XSS prevention
- Dependency injection
- Ajax requests through @angular/HTTP
- Routing via @angular/router
- Component CSS encapsulation
- Unit-testing utilities
- @angular/forms for constructing forms

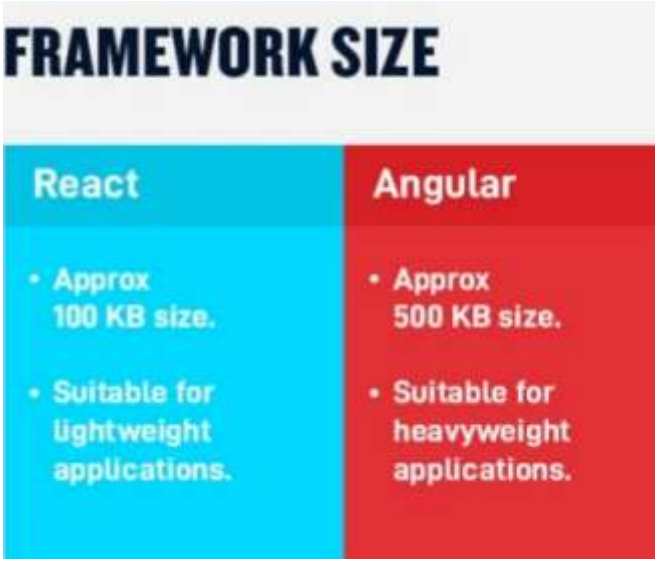
React, on the other hand, allows you far more flexibility. In MVC, it simply supplies the "view" — you must solve the M and C on your own. As a result, you can select any of your own libraries as you see fit. You'll wind up employing a lot of different, fast-moving libraries. As a result, you will have to handle the necessary upgrades and migrations on your own. Furthermore, each React project is unique and necessitates a decision about its folder organisation and design. Because of this, things may go wrong far more easily.

React includes the following features "out of the box": Instead of classic templates, it has JSX, an XML-like language built on top of JavaScript [2]: -

- XSS prevention
- No dependency injection
- Fetch for Ajax queries
- Unit-testing utilities

These are some common libraries for adding functionality.

- React-router for routing
- Redux or MobX for state management
- Enzyme for extra testing



ii. Regular DOM vs. Virtual Dom

One of the aspects that makes React so quick is its usage of a virtual DOM. You've most likely heard of it. When it was originally launched, it was dubbed the "killer feature" of React. Let me illustrate with an example:

Assume you wish to change a user's age within a block of HTML elements. A virtual DOM just considers the differences between prior and current HTML and modifies the section that has to be changed. Git uses a similar mechanism to discern between changes in a file.

Angular, on the other hand, chose to use a standard DOM. This will update the entire HTML tag tree structure until it meets the user's age.

So, why is this important? The above example is insignificant. This is unlikely to make a difference in a real-world app. However, if we have hundreds of data queries on the same page (and the HTML block is modified for each page request), it has a significant impact on performance as well as the user experience.

iii. Templates – JSX or HTML

React chose to integrate UI templates with inline JavaScript code, something no other

business had done previously. The end result is known as "JSX." Although it may have sounded like a horrible idea at the time, Facebook's risk paid out handsomely. React employs the concept of a component, which contains both the HTML and the functionality in the same file. It also employs an XML-like language, which enables you to create markup right in your JavaScript code. Because everything is in one location, and code completion and compile-time checks perform better, JSX is a major advantage for development.

Ex. We declare a variable name in this example and utilise it within JSX by enclosing it in curly braces:

```
const name ='Josh Perez';
const element =<h1>Hello,{name}</h1>;
```

Angular employs templates that are augmented HTML with Angular directives ("ng-if" or "ng-for"). React merely requires knowledge of JavaScript, but Angular requires knowledge of its unique syntax.

React Fiber

React Fiber will take React from "quick" to "blazingly fast." Fiber is a complete rework of the React core that is backwards compatible. It was added to react v16, and the update went so smoothly that you probably didn't notice [3]. React may pause and restart processing as needed using Fiber to get what matters onto the screen as rapidly as possible.

iv. Components

Both React and Angular are component-based frameworks. A component accepts an input and, after some internal logic, returns as output a displayed UI template (for example, a sign-in form or a table). Components should be simple to reuse within other components or even across projects. For example, a sign-in component may consist of two text inputs (user and password) and a "Login" button. This component may have distinct characteristics and underlying logic, but it should be universal so that it may be reused with different data on another page or app.

Components are self-contained "chunks" of your programme that you may reuse in a variety of contexts [1]. They are intended to include logic. The web is gradually transitioning to a component-based architecture. As a result, I propose that you begin acclimating to them immediately away.

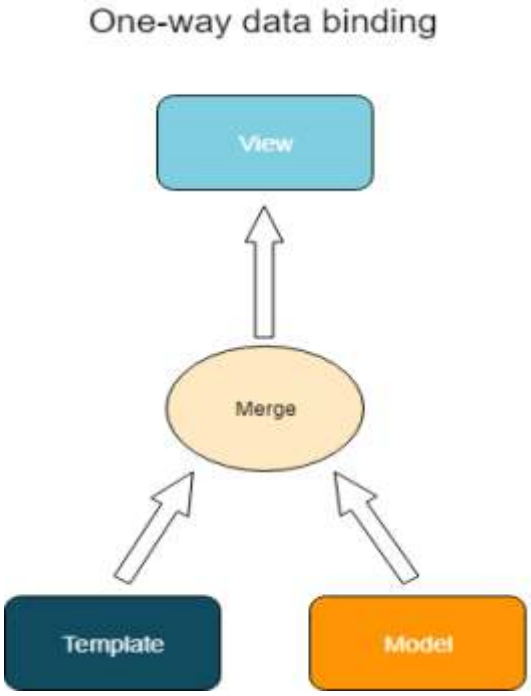
v. Data Binding

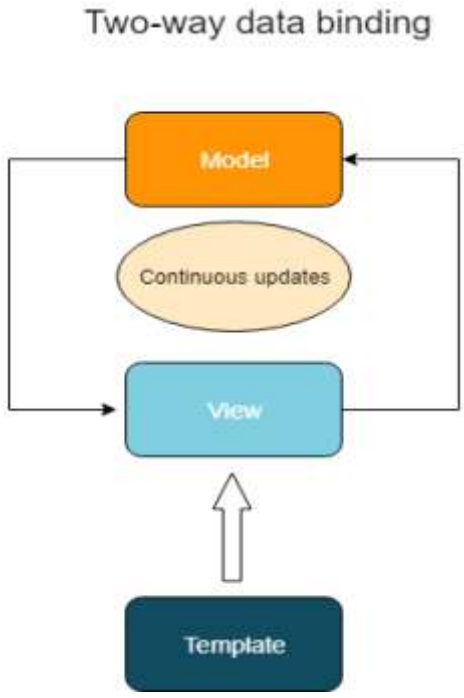
One major distinction between React and Angular is one-way vs. two-way binding. Angular makes advantage of two-way binding. For example, with Angular, whenever you modify the UI element (a user input), the accompanying model state changes as well. Furthermore, changing the model state causes the UI element to change, resulting in two-way data binding.

React, on the other hand, only supports one-way binding. The model state is first changed, and then the change is displayed in the UI element. However, changing the UI element has no effect on the model state. You must determine this for yourself. Callbacks and state management libraries are two typical methods (see State Management in the previous section).

I must acknowledge that Angular's way is first simpler to grasp. However, when the project

grows in size, React's approach provides a superior data overview (making debugging much easier).





vi. TypeScript vs JavaScript/Flow

React employs JavaScript, a dynamically typed language (which means you don't have to declare the variable's type). Because many developers already know and enjoy JavaScript, this might be viewed as a benefit.

If you wish to utilise Angular, you must first become acquainted with TypeScript. TypeScript is a statically typed language, which implies you must declare the variable's type (string, number, array, etc.). It is just a transpiler that converts TypeScript code to JavaScript code, allowing you to write ES5 code in a TypeScript file.

The goal of Typescript is to enable a smooth transition for programmers with an Object-Oriented Programming (OOP) background. TypeScript was also launched during the ES5 era, although ES5 was not a class-based OOP language at the time.

Since then, JavaScript has matured and undergone several significant modifications. Modules, classes, spread operators, arrow functions, template literals, and other features are available in ES6. It enables developers to create declarative code while retaining the features of a genuine OOP language (that is, including class-based structure).

However, one advantage of TypeScript is that it provides greater uniformity in online examples (React examples can be found in either ES5 or ES6).

We are undoubtedly also aware that we can utilise Flow to enable type checking in our React app. Facebook created a static type-checker for JavaScript. We can utilise Type Script in our React project if we want (but it isn't natively included).

Ex1. JavaScript and TypeScript property comparison

```
// JavaScript (ES6)
const name;
```

```
// TypeScript
const name: string;// <-- static typed!
```

Ex2. JavaScript vs. TypeScript Argument Comparison

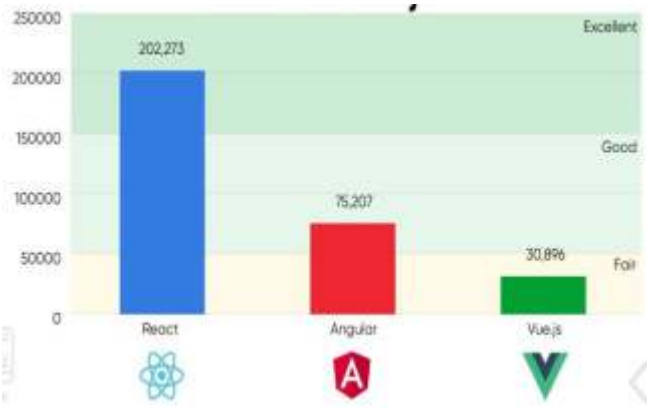
```
// JavaScript (ES6)
functiongetName (name,age) {
return name + age;
}

// TypeScript
functiongetName (name: string, age:number) {//
                    <-- static typed!
return name + age;
}
```

Ex3. Here's a quick class-object comparison between JavaScript and TypeScript.

```
// JavaScript (ES6)
classGreeter{
constructor (message){
this.greeting= message;
}

greet(){
return"Hello, "+this.greeting;
}
}

let greeter =newGreeter("JavaScript!");
greeter.greet ()
//  Hello, JavaScript!

// TypeScript
classGreeter{// <-- static typed!

greeting: string;

constructor (message:string){
this.greeting= message;
}

greet(){
return"Hello, "+this.greeting;
}
}

let greeter =newGreeter("TypeScript!");
greeter.greet()

//  Hello, TypeScript!
```

vii. Developer's Perspective

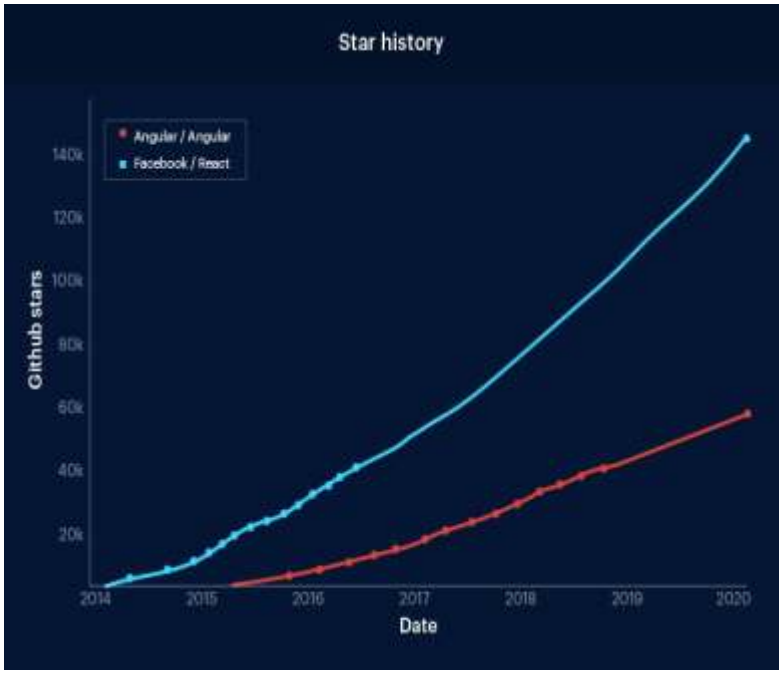
Angular is straightforward to learn but takes longer to produce projects since it has a higher learning curve and employs a lot of superfluous syntax for the most basic tasks, increasing coding effort and delaying project delivery [3]. React is more difficult to set up than Angular, but it allows you to develop projects and build apps faster. In addition, unlike Angular, you may add new functionality to react by using other libraries and React lacks model and controller components.

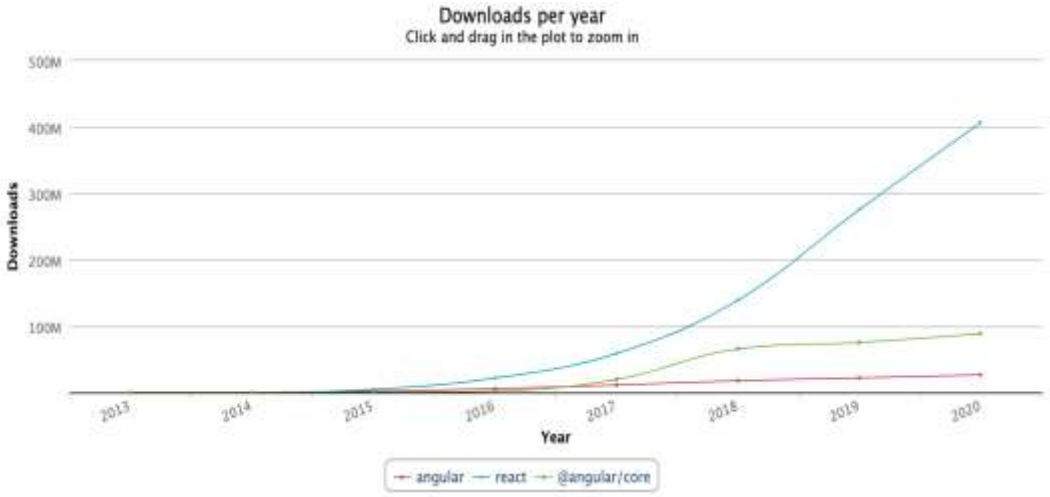


viii. Community Support

As previously said, React is more popular among developers on GitHub and NPM. React has over 135k stars on GitHub, indicating its popularity within the developer community 5]. However, 2018 Stack Overflow research indicated that Angular was more popular. According to another poll, over 75% of React users would use it again.

Because of the Virtual DOM implementation, React has a broad user base and the apps may be updated and rendered more quickly than Angular apps. However, Angular has been endorsed and heavily utilised by Google programmes such as Google AdWords.





NPM Statistic

Conclusion:

Finally, we must get to a stage when we can choose any one technology. One pizza team chose React because they wanted to design an application with a low learning curve. When developing an enterprise-grade application, teams may choose Angular since the steeper learning curve is no barrier.

Overall, React chooses an easy approach to get the work done because it has nothing to do with structuring HTML and instead provides the easiest ways to reuse the UI components. While Angular is capable of managing various tasks without the need for external assistance, it may appear difficult at first. However, the benefits outweigh the work put in the more extensive notion.

References :

1. https://www.cuelogic.com/blog/
2. https://scholar.google.com/
3. https://www.theseus.fi/bitstream/handle/10024/261970/Thesis-Elar-Saks.pdf?sequence=2
4. https://www.cuelogic.com/blog/
5. https://www.google.co.in/