Decision Analysis Report - Front End

Project Name: Tresearch
Application Type: Web Application

Trial By Fire
Ian Ho-Sing-Loy
Jessie Lazo
Matthew Chen (Team Lead)
Pammy Poor
Ryan Harrison
Viet Nguyen

Instructor: Vatanak Vong

California State University, Long Beach
College of Engineering
CECS491B Section 06 Spring 2022

Purpose

The purpose of this document is to recommend a framework or library that will provide a front end framework/library for our application.

Technology Comparisons

- Angular version 13.1.1
    - Open Source framework used to build mobile and desktop web applications. Angular uses html and typescript.
- React version 17.0.2
    - Open source front end Javascript framework used to build user interfaces for large web applications.
- Vue version 3.2.20
    - Open source front end Javascript framework used to build interfaces and single-page applications.

Metrics

Show/Hide Speed

- Analyzes the speed at which each framework hides and shows an element/component.

Custom Invokable Code

- Custom invokable code refers to code that is triggered when an event happens. In our case we are interested in invoking code when a user's mouse is dragged. This should be necessary for dragging through the tree portal.

Data share across Views

- We are interested in the ability to share data across all components. We want the ability to share data even if there is no direct connection.

Frequency of Updates

- We are interested in a framework that attempts to stay up to date as we want to futureproof our project as much as possible

Benchmark Data

I. Show/Hide Speed
    - For this metric we measured the time each framework took to show and hide 15,000 elements displaying Hello World!. We then measured the speed using a stopwatch. We chose a higher number of elements as there was almost no difference between the frameworks with lower numbers

Angular Framework

Trial	Show	Hide
Trial 1	1.44 s	1.42 s
Trial 2	1.46 s	1.39 s
Trial 3	1.42 s	1.41 s
Trial 4	1.42 s	1.42 s
Trial 5	1.44 s	1.44 s
Trial 6	1.43 s	1.42 s
Trial 7	1.42 s	1.43 s
Trial 8	1.44 s	1.42 s
Trial 9	1.43 s	1.40 s
Trial 10	1.46 s	1.42 s
Average Time	1.44 s	1.42 s

React Library

Trial	Show	Hide
Trial 1	1.42 s	1.21 s
Trial 2	1.41 s	1.23 s
Trial 3	1.39 s	1.29 s
Trial 4	1.42 s	1.24 s
Trial 5	1.45 s	1.26 s
Trial 6	1.39 s	1.24 s
Trial 7	1.42 s	1.23 s
Trial 8	1.42 s	1.23 s
Trial 9	1.42 s	1.23 s
Trial 10	1.43 s	1.24 s
Average Time	1.42 s	1.24 s

Vue Framework

Trial	Show	Hide
Trial 1	1.43 s	1.42 s
Trial 2	1.46 s	1.42 s
Trial 3	1.42 s	1.41 s
Trial 4	1.41 s	1.42 s
Trial 5	1.42 s	1.44 s
Trial 6	1.43 s	1.42 s
Trial 7	1.42 s	1.44 s
Trial 8	1.44 s	1.42 s
Trial 9	1.42 s	1.4 s
Trial 10	1.46 s	1.42 s
Average Time	1.43 s	1.42

Analysis Between Angular, React and Vue

	Angular Framework	React Library	Vue Framework
Show/Hide Elements (0.5)	Showed 10,000 elements on average 1.44 s. Hid 10,000 elements on average 1.42s (0.75)	Showed 10,000 elements on average 1.42 s. Hid 10,000 elements on average 1.24s (1)	Showed 10,000 elements on average 1.43 s. Hid 10,000 elements on average 1.42s (0.85)
Invoke code when mouse is dragged (1)	Yes but through the use of an external component (0.25)	Yes, using ondrag, onDragStart, onDragEnd. Only needs one function call (1)	Yes, using mousemove and mousedown. (all at the same time) (0.5)
Shares data across views (0.75)	Yes, parent to child via input, child to parent using view child or output, without direct connection can use BehaviorSubject	Yes, can pass data from parent to child using prop, child to parent using callback and states, between siblings using the above features.	Yes, can pass data from parent to child, child to parent using custom events, and application shared state

	component (1)	(0.75)	using Vuex. (0.50)
Frequency of updates (0.25)	6 updates within 3 years (0.5)	17 updates within 3 years (1)	2 updates within 3 years (0.25)
Total:	2	2.25	2.125

*Through the use of an external component

Recommendation

Based on the analysis between Angular, React and Vue.js, React is the clear choice based on its performance in our metrics. Since we care most about invoking custom code when dragging, we emphasized the score more than the others. When it came to invoking code when mouse is dragged, we scored React the highest as they had a single method call (onDragStart) while Vue had to use a mixture of both mousemove and mousedown. While Vue performs similarly to React, the frequency of updates means that React should be future proof.