Optimizing Single-Page Applications with Angular JS

Hareesh Kumar Rapolu

hareeshkumar.rapolu@gmail.com

Abstract

The following research project has examined the optimization of single-page applications with AngularJS. It has explained the key benefits of single-page applications such as enhanced user experience, development perceived performance and useful for offline functionalities. Moreover, nurturing with the best practices like ensuring smooth performance and enhancing user experience has served to be detrimental to resolving challenges like SEO concerns with accessibility issues and low initial loading time. Therefore, this research project has exemplified a detailed understanding of utilising AngularJS in single-page applications to boost user engagement and streamline performance.

Keywords*:* Single-Page Applications, AngularJS, Optimization, Lazy Loading, Caching, User Experience, MVC architecture, Best Practices

I. INTRODUCTION

The research project will provide a nuanced understanding of the optimization of single-page applications with AngularJS. It will contain processes like the application of lazy loading for modules with minimising data bonding complexity and optimization of data collection. At the same time, it will also contain caching mechanisms followed by minifying the code and optimum management of the DOM applications. AngularJS will stand to be responsible for using bidirectional UI data binding that is used by organisations to stay adapted regarding single-page applications. This research project will also shed light on the key benefits of single-page applications and building them with the help of AngularJS. Furthermore, the challenges in single-page applications will be reflected and will be mitigated by the incorporation of best practices.

II. DEFINING SINGLE-PAGE APPLICATIONS IN ANGULARJS

The following section describes single-page applications also abbreviated as “SPA”. It is defined as a web application functioning within a single HTML page and the server. It is not like the traditional applications that were used to load a new page for every user interaction. SPA is considered to be a much more advantageous titan JavaScript as it dynamically updates the overall content of the present page[1]. A few examples of single-page applications refer to Gmail, Google Maps, Facebook and others. Therefore, single-page applications are termed to be much more effective and deliver impeccable app-like experiences.

Figure 1: Portraying AngularJS

III. DESCRIBING THE KEY BENEFITS OF SINGLE-PAGE APPLICATIONS IN ANGULARJS

This section explains the key benefits of single-page applications. These benefits are observed below.

Developed Perceived Performance: In terms of perceived performance, single-page applications only load the relevant data for each view or action. It is observed that the initial page can be longer due to the incorporation of more code[2]. This results in fuellingup the interactions to load it swiftly.

Enhancing User Experience: Single-page application is advantageous for enhancing user performance. It is attained by providing a streamlined and more responsive UX. This results in maximised user engagement and satisfaction.

Useful for Offline Functionality: It is evident that a single-page application is beneficial for offline usage. This is done by caching the data locally in the browser from the user[3]. As a result, this makes the SPA work even without an internet connection or in times of poor network connection.

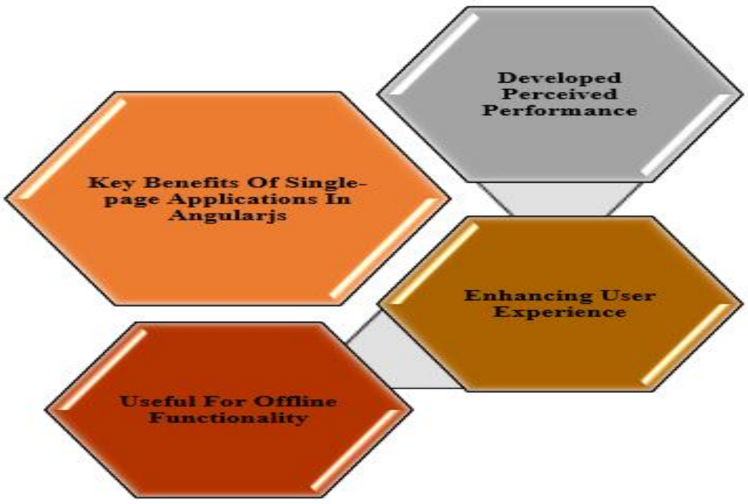
Developed Perceived Performance
Key Benefits Of Single-page Applications In Angularjs
Enhancing User Experience
Useful For Offline Functionality

Figure 2: Depicting Key Benefits of Single-Page Applications in AngularJS

IV. BUILDING SINGLE-PAGE APPLICATIONS WITH ANGULARJS

The following section builds a single-page application with AngularJS. It is noticed that AngularJS utilises a Model-View-Controller (MVC) infrastructure which contains application data and the view indicates the user interface. Massaging the data followed by the comptroller making the model and the view has a two-way binding. This renders complete dependency with injection and routing. To build a single-page application it is recommended to begin with a module and a controller and then use routing to portray several dimensions. Then implementation of AngularJS directives such as UI elements helps to handle the event and thus create a highly interactive application responding to the user input. A single-page application constructed with AngularJS is mentioned below.

```
// Defining the AngularJS module
angular.module (‘myApp’, [‘ngRoute’])
//Configuring the Routing Channels
.config(function($routeProvider)
{
$routeProvider
.when(‘/home’, {templateUrl: ‘home.html’, controller: ‘HomeCtrl’})
.when(‘/about’, {templateUrl: ‘about.html’, controller: ‘AboutCtrl’})
.otherwise({redirectTo: ‘/home’});
}
)
// Creating the Home Controller
.controller(‘HomeCtrl’, function($scope)
{
$scope.message = “Welcome to the Home Page!”;
}
)
// Creating the About Controller
.controller(‘AboutCtrl’, function($scope)
{
$scope.info = “This is the About Page.”;
}
)
```

V. ILLUSTRATING THE CHALLENGES OF SINGLE-PAGE APPLICATIONS

This section highlights that while implementing AngularJS, there are certain challenges faced by single-page applications which cannot be ignored. These challenges need to be identified and mitigated to determine positive outcomes. These changes are described below.

Accessibility Concerns: It is found that single-page applications stand to be widely dependent on JavaScript that might encounter challenges related to accessibility for the users with disabilities. It poses to be critical as it ensures proper semantic structure and signifies to apply robust Accessible Rich

Internet Applications “ARIA”[4]. This signifies that the users get complete access to single-page applications.

Low Initial Loading Time: At times it is observed that loading a basic webpage requires less time as contrasted with other fundamental code like CSS, HTML and JavaScript with regards to a single page. This is identified to be a significant challenge due to a slow internet connection[5]. As a result, this seeks to block the users from accessing features such as lazy loading and code splitting.

Search Engine Optimization (SEO) Challenges: Search Engine Optimization challenges occur from single-page applications when they are dependent on content generation with the help of JavaScript. It poses a gradual challenge for search engines to develop curated understanding[6]. This type of challenge needs to be limited by the utilisation of server-side rendering also known as SSR to solve this kind of challenge.

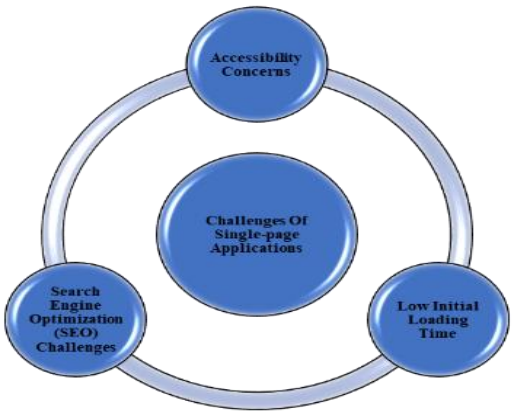
Accessibility Concerns
Challenges Of Single-page Applications
Search Engine Optimization (SEO) Challenges
Low Initial Loading Time

Figure 3: Identifying Challenges of Single-Page Applications

VI. ELUCIDATING BEST PRACTICES FOR SINGLE-PAGE APPLICATIONS DEVELOPMENT

The following section nurtures best practices to contribute to the overall development of the single-page application in a progressive manner. These practices are explained below.

Enhancement of User Experience (UX): User experience (UX) can be enhanced by the optimization of assets that aidin minimising the size of images with scripts and other resources. It considers server-side rendering for the initial page load to elevate SEO and perceived performance. Additionally, the involvement of code splitting provides complete access to the users to use the core functionalities at a swift pace.

Ensuring a Smooth Performance: A smoother performance is achieved by collecting the important data for each view leading to the development of single-page applications. It consists of caching mechanisms to store accessed data constantly[7]. Furthermore, the utilisation of pagination for large data sets the stage to portray information to handle the pieces.

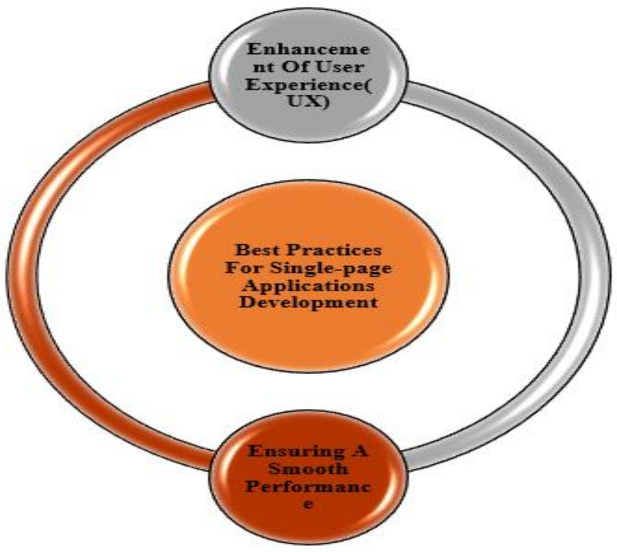
Enhanceme nt Of User Experience(UX)
Best Practices For Single-page Applications Development
Ensuring A Smooth Performanc e

Figure 4: Demonstrating Best Practices for Single-Page Applications

VII. CONCLUSION

This research project has optimized single-page applications with AngularJS that are used to represent optimum opportunities. It has been considered to be useful to improve user experience with performance and efficiency. At the same time, creating responsive applications has served to be beneficial for the developers to observe effective functioning in several scenarios. Moreover, highlighting challenges such as SEO concerns, accessibility issues and initial load times has been mitigated with the help of best practices. This has resulted in controlling the challenges and thus maximising the benefits of AngularJS.

Abbreviations and Acronyms

- SPA- Single-Page Applications
- MVC- Model-View-Controller
- SEO- Search Engine Optimization
- ARIA- Accessible Rich Internet Applications
- SSR- Server-Side Rendering
- UX- User Experience

Units

- Time is measured in milliseconds
- Data is calculated in kilobytes

Equations

- Time to Interactive (TTI) = [Page Load Time + JavaScript Execution Time]
- First Contentful Paint (FCP) = [Time to First Byte (TTFB) + Content Rendering Time]

REFERENCES

[1] B. Sutar and D. Pratibha Adkar, “Angular JS and Its Important Component,” May 2019. Available:https://www.irejournals.com/formatedpaper/1701238.pdf

[2] G. Zhang and J. Zhao, “Visualizing Interactions in AngularJS-based Single Page Web Applications,” *International Conferences on Software Engineering and Knowledge Engineering*, vol. 2018, pp. 403–450,Jul.2018,doi:https://doi.org/10.18293/seke2018-066.

[3] J. Gunawan and R. R. Kosala, “Genie Enterprise Resource Planning for Small Medium Enterprises Implementing Single Page Web Application,” *IOP Conference Series: Earth and Environmental Science*, vol. 426, p. 012170, Mar. 2020, doi: https://doi.org/10.1088/1755-1315/426/1/012170.

[4] M. Sultan, “Angular and the Trending Frameworks of Mobile and Web-based PlatformTechnologies:AComparativeAnalysis,”Nov.2017.Available:https://mohamed-sultan.com/paper/128_Paper_264Angular_and_the_Trending_Frameworks_of_Mobile.pdf

[5] N. S. Bondili and Singh, “Online Banking Application with Angular JS, RESTful Web ServicesandCassandraDatabase,”vol.26,Mar.2017.Available:https://repository.stcloudstate.edu/cgi/view content.cgi?article=1028&context=msia_etds

[6] N. Sanandan, “Designing Efficient Single Page Web Applications,” Dec. 2018.Available:https://etd.auburn.edu/bitstream/handle/10415/6525/DesigningEfficientSinglePageWeb Applications.pdf?sequence=2

[7] S. Huang, “Load time optimization of JavaScript web applications,” May 2019.Available:https://www.divaportal.org/smash/get/diva2:1318692/FULLTEXT01.pdf