SINGLE PAGE APPLICATIONS — A GROWING TREND IN MODERN WEB DEVELOPMENT

Single-page applications (**SPAs**) are now becoming very common on the internet. Majority of the most popular websites (e.g. Facebook, Google, etc.) are using SPAs in one way or another. In this article, we will look compare SPAs with traditional websites and discuss its advantages and disadvantages.

First, what is an *SPA*? A *single-page application* is basically a web application that runs on a single webpage aiming to provide a similar experience to that of using a desktop application.

In order to compare SPAs with traditional web applications, let us recall how traditional web applications work.

Traditional web applications usually require full-page reloads when browsing from page to page. User input utilizes forms that will send the data to the server and then redirect the user to another page, causing again another full-page reload. This process of sending data back-and-forth from client to server takes time and it uses more bandwidth due to the full page being reloaded.

The traditional model worked well before when the world wide web was just a world of mostly static pages and links. Throughout the years however, the web has evolved from being static to being dynamic and most of the content on the internet are now user-generated. This shift eventually led to the need to provide a better experience to users.

In an *SPA*, user actions do not usually result into a page reload. These actions are instead processed by *Javascript* and appropriate changes to the page are made dynamically. If there is a need to retrieve or send new data to the server, it will be done in the background through *AJAX*. Data being sent and received this way uses less bandwidth since it usually contains just the raw data needed to update a partial area of the page. Since the page does not reload itself, it provides a more seamless and interactive user experience better than traditional web applications.

With the SPA model, two obvious potential problems come to mind:

1. *Search Engine Optimization* - The content is usually generated by Javascript and search engine crawlers do not use Javascript. As a result, the search engine crawler cannot see and index your content.

2. *Page linking* - Since the content of the page can be altered by Javascript at any time, the webpage address may or may not reflect the current state of the page. A user bookmarking the page may or may not be brought to the same content when that bookmark is clicked in the future.

You do not have to worry though since there are solutions to those problems. The way these problems are solved would vary depending on the Javascript framework being used and is beyond the scope of this article. Suffice it to know that these obstacles should not hinder you from using the SPA model in your website.

So which model should you use for your website?

If your website contains mostly static information and does not require much user input and interaction, you may do well with the *Traditional Web Application* model. The benefits that you get from the SPA model are just not worth the added complexity.

If your website requires a lot of user input and interaction and client-side processing like calculations, it may be worth it to develop your website with the *SPA* model. Your users will really appreciate the much better dynamic experience.