Technology Survey
Group 8

By: Bilal Ahmed, Jeth Balabagno, Hazzem Cordash, Elijah Kruse, Timothy Xiong, Julian Moreno-Johns

Introduction:

In today's technological landscape, there is an overabundance of frameworks, libraries, and tools available for software developers. This makes choosing the right technologies not only crucial, but also daunting. This Technology Survey Report aims to simplify that decision making process by presenting a review of candidate technologies that are suited for our specific project needs. We have identified the top three to four frameworks and libraries used for tasks such as deploying web-based applications and backend data services. The report not only lists these technologies but also provides justifications for choosing them over their alternatives.

Front-end software:

Software	Pros	Cons
Bootstrap	● Easy to learn and use ● Javascript plugins ● Agile responsive ● Large developer community	● Creativity and uniqueness limitations ● Adds a lot of extra code and files ● Slow loading speed
React	● High performance (virtual DOM) ● Popularity and versatility (lots of libraries and tools) ● Mobile app development	● Not a full featured framework ● Poor documentation ● Lack of conventions
Tailwind	● Easy to use ● Better CSS styling process ● Highly customizable	● Large HTMLfiles ● Small learning curve ● Installation is required ● Lack of constraints

BOOTSTRAP

Pros: Bootstrap helps our members code more efficiently with built in responsiveness so our application can work and adapt with multiple devices. With the addition of the many javascript plugins, our members will make great use of them. Moreover, bootstrap is simple to work with. Things like resizing images helps automate our process better and efficiently. Finally we can grab from the pile of user-made templates to make our process smoother as well.

Cons: Even though there are so many good things about bootstrap, there are still some drawbacks to it. First off a lot of websites use bootstrap. One might think this is a good thing but, if you want to make your page look unique and distinct, it won't unless you really work around it. Given Bootstrap's learning curve, that's easier said than done (easy to start, mastering more features takes time). Another thing is that it'll add a lot of extra files and code. This will make loading longer and less efficient for whoever is opening the page.

REACT:

Pros: React is one of the most popular javascript libraries, React can be used to create anything you see all around the web. React has a high performance, creating highly scaled single page applications that can work dynamically on different platforms. React uses the concept of virtual DOM to avoid having to update the DOM for every change caused by a user's interaction on the webpage. Moreover, REACT is also easier to pick up with a background of programming overall. With flexibility and not following a specific architectural pattern.

Cons: REACT is not a full-featured framework not like angular. It is robbed of different things like if you look into the MVC (model view controller) architecture, react only handles the view. This can result in poor structure of code and its patterns (but could help with code readability). Another con is the poor documentation because REACT is evolving so fast with new tools and patterns. It's hard for new developers like us who want to start with React and make it slower for us to learn.

TAILWIND:

Pros: Tailwind CSS is easy to use to help create a unique UI. Tailwind also has 500+ components for us to use in our designs. This will aid in the implementation of a consistent design system. Another thing it has a better CSS styling process with hundreds of built in classes that eliminate the need to start from scratch when creating designs overall.

Cons: With Tailwind the HTML files are large because it involves a large number of classes in HTML. This will therefore increase download size of the HTML file and slow loading speed is bound to happen. Another thing is we have an installation process to generate the CSS. This may require additional resources and could potentially incur additional costs. Tailwind's uniqueness also makes it a little harder to learn. Also, with all the flexibility it provides, there's risk of inconsistency due to its lack of design constraints.

Back-end software:

Software	**Pros**	**Cons**
Django	• Python based • Variety of built-in features • Extensive documentation	• Can be overkill for smaller projects • Many built-in features limit customizability
Flask	• Python based • Lightweight • Easy learning curve	• Less built-in libraries • Not as well documented as Django
Node.js with Express	• JavaScript on front and back-end • Well suited for real-time applications • Easy learning curve	• Single-threaded • Less stable than Python or Java

DJANGO:

Pros: The django framework provides a lot of built-in features like an admin panel, user authentication, and ORM (Object Relational Mapper). Given that a lot of us have been exposed to and have worked with Python it offers a smoother learning curve when building the app. It also has a lot of comprehensive and easy to follow documentation which can aid the team in faster development.

Cons: Since django is a mature framework with plenty of tutorials and third-party packages, the abundance of built-in features can be overwhelming. Also, because of this fact it can limit customizability and could be overkill for a smaller project.

FLASK:

Pros: This framework unlike django is lightweight and gives us the basics to get an app up and running. This in turn gives us more control on how to structure our app. It's also fairly simple and straightforward giving it a smaller learning curve and it's python based which fits with our teams' experience.

Cons: One of the biggest drawbacks with using Flask is that it lacks a lot of built-in features and libraries compared to other frameworks. Meaning that we would have to either implement some features manually or find third-party libraries. Flask's ecosystem is growing and in comparison to django it is not as well documented and lacks an extensive package library unlike django.

NODE.JS WITH EXPRESS:

Pros: This framework is designed to be scalable and perform quickly. It uses JavaScript on the front and back-end which allows for uniformity and streamlined development. It also has a large selection of libraries available through NPM. One of the best reasons why this framework could be beneficial to our project is that it was built with real-time, data intensive operations in mind and can easily integrate with WebSocket libraries.

Cons: This framework is single threaded meaning it is less suitable for CPU heavy operations. Also, some of the libraries can be less stable compared to more mature frameworks such as django. Lastly, asynchronous programming can get complex but can be manageable, this just adds an extra layer to take care of.

Interactive Mapping Services:

Software	Pros	Cons
Google Maps JavaScript API	● Widely used & recognized ● Plethora of features & tools ● Strong dev community	● Not open source; less customization ● Higher use equals higher cost
Mapbox	● Large amount of customization ● Provides geographical depth in features	● Free tier has limitations ● Complex system
Leaflet	● Provides plugins for customization ● Open source	● Integration not as smooth as others ● Less functionalities

Google Maps JS API:

Pros: Google Maps stands out for its accurate data collection and vast user trustworthiness. The ease of integration with other services also makes it a standout applicant compared to alternatives.

Cons: Despite its vast data and user base, Google Maps is very restrictive in terms of customization, and can become costly when working at bigger scales. This makes it less ideal for an application that wants to have a *distinct* look.

Mapbox:

Pros: Mapbox offers an unparalleled amount of customization and map styles, which may be able to give us a more distinct and unique look to our application. Due to its vector rendering, it also ensures smooth and efficient map visuals.

Cons: While intricate customization is offered, this requires a steeper learning curve compared to more straightforward platforms. This may be a hurdle for members of the team with lesser experience in the development schedule.

Leaflet:

Pros: The open source nature Leaflet provides allows for a broad plugin ecosystem that allows for a lot more flexibility in functionality. This can help us implement more features and make our application stand further apart from other similar applications.

Cons: The lack of built in features as competitors and range of customization may not be a fair trade off for the ease of use of Leaflet. With a weaker foundation in building a unique mapping system, it may not be worth the lightweight integration.

Citations

Pillai, Sujeet. "Pros and Cons of Using Tailwind CSS." *Incentius*, 30 Dec. 2022, www.incentius.com/blog-posts/pros-and-cons-of-using-tailwind-css/.

"What Are the Pros and Cons of React." *What Are the Pros and Cons of React*, www.knowledgehut.com/blog/web-development/pros-and-cons-of-react#cons%C2%A0%C2%A0. Accessed 26 Oct. 2023.

Inouye, Jenna. "What Is Bootstrap? Pros and Cons of This Framework." *Hackr.Io*, hackr.io/blog/what-is-bootstrap-framework. Accessed 26 Oct. 2023.

"Node JS Documentation." *Node.Js*, nodejs.org/en/docs. Accessed 26 Oct. 2023.

"Django Documentation." *Django Project*, docs.djangoproject.com/en/4.2/. Accessed 26 Oct. 2023.

"Welcome to Flask¶." *Welcome to Flask - Flask Documentation (3.0.x)*, flask.palletsprojects.com/en/3.0.x/. Accessed 26 Oct. 2023.

"Differences between Django vs Flask." *GeeksforGeeks*, GeeksforGeeks, 6 Sept. 2023, www.geeksforgeeks.org/differences-between-django-vs-flask/#.