BRAUDE
College of Engineering, karmiel

Choosing Best Fit Framework For Web Application "EVolve"

Department of Software Engineering

Capstone Project Phase A, 23-2-D-1

Supervisor:

Mr. Alexander Keselman,

alex@vir-tec.net

Authors:

Daniel Moshe Ben David,
Daniel.Moshe.Ben.Dav@e.braude.ac.il

Idan Daar,

Idan.Daar@e.braude.ac.il

Table Of Contents

Abstract

This paper presents a study on the selection of the best fit framework for web application development. The motivation behind this work is to provide web developers with a comprehensive solution to make an informed decision on which framework to choose for their web application development. Our study aims to address this gap in the literature by implementing a web application using a few framework combinations and evaluating which one performs best. The result of this project will be a web application designed for electric vehicle owners, allowing them to locate nearby charging stations around their current location and receive information for a specific position. By doing so, we hope to provide developers with a clear understanding of the benefits and disadvantages of different web development frameworks and ultimately help them make an informed decision on which one to use.

1. Introduction

Web development has evolved over the years, and new frameworks are being introduced every day, making it a challenge for developers to choose the best one for their web applications. With the increasing number of frameworks available, web developers often face difficulties in deciding which one to use, as each framework comes with its own unique features and capabilities. The problem that arises is how to choose the best framework for developing web applications that can satisfy user needs while providing a seamless user experience.

The motivation behind this work is to help web developers make an informed decision on which framework to choose for their web application development. While there are existing studies that have addressed this issue, none of them provide a comprehensive solution. Moreover, with the ever-increasing number of frameworks being developed, it has become even more challenging to make a choice. Our study aims to address this gap in the literature by implementing a web application using a few framework combinations and evaluating which one performs best.

In this work, we will provide a background and review of related work on web development frameworks. We will also discuss our expected achievements, research and engineering process, as well as the product and our verification plan. By doing so, we hope to provide developers with a clear understanding of the benefits and Disadvantages of different web development frameworks and ultimately help them make an informed decision on which one to use.

The structure of this work will be as follows: we will start with a background and review of related work on web development frameworks.

We will then discuss our expected achievements and objectives. Next, we will outline our research and engineering process, including the methods and tools used. We will then describe the process of implementing the web application using different framework combinations, followed by the evaluation and verification plan. Finally, we will conclude with a summary of our findings and provide recommendations for web developers.

1.1. Paper Overview

The remainder of this paper is structured as follows: In Section 2, we provide a review of the background and related work that has been conducted to address the problem we presented. Our goals and expected achievements are outlined in Section 3. Section 4 is dedicated to the research process we carried out during the semester to achieve our objectives, along with an explanation of the motivation behind our approach and the potential constraints that may impact our development process. We describe our software and testing methodology in Section 5, which is followed by an overview of the evaluation metrics utilized in this work in Section 6.

2. Background and Related Work

Modern web applications, due to the functionalities they provide in their user interfaces, have a complex program structure, and manually writing a program code, due to the complexity of the entire application, can result in uneven quality and content of individual application parts.

One solution to solve this problem is choosing the appropriate framework for a specific application.

There are various frameworks that can be used in the development of web applications, for different parts of the application.Today developers use a large amount of frameworks, and in most cases they are chosen according to popularity. On the other hand, in recent years several new frameworks have been developed, allowing to improve important application parameters.

Our goal is to find the best frameworks for our web application in terms of code size, complexity, etc.

2.1. Client-Side and Server-Side

Client-side and server-side are terminologies used in web development that describes where the application code is executed, Although sometimes they are referred to as frontend and backend, they do not have exactly the same meaning.

2.1.1. Understanding Client-Side term

In web development, the term "client-side" pertains to all elements of a web application that are displayed or occur on the client or end-user device. The client-side is often referred to as the frontend, although these terms do not necessarily imply the same thing. While the client-side denotes the location where processes are executed, the frontend pertains to the types of processes that operate on the client-side.

The client-side encompasses the user interface, such as text, images, and other visual elements, as well as any actions performed by the application within the user's browser like markup languages such as HTML and CSS that are processed by the client's browser.

Nowadays, developers often integrate client-side processes into their application architecture, shifting away from server-side processing.For instance, the Netflix.com webpage's HTML, CSS, and JavaScript determine how the main page is displayed to the user and is interpreted by the browser on the client-side. The page is also capable of responding to events, such as enlarging an image when the user's mouse hovers over it, with adjacent thumbnails moving to the side to make space for the enlarged image. This is an instance of a client-side process, where the code within the webpage responds to the user's action and initiates the reaction without communicating with the server.

Another example to the client side is a dynamic webpage that changes based on user input and does not display the same content for all users. The Facebook homepage is an example of a dynamic page, while the Facebook login page is typically static.

2.1.2. Understanding Server-Side term

Similar to the term "client side," "server side" refers to all the operations that take place on the server rather than on the client side. Similar to "client-side" and "frontend," "backend" also refers to processes that occur on the server, but "backend" specifically relates to the types of processes, while "server-side" refers to the location where the processes are executed.

In the past, the majority of business logic was carried out on the server side, which included tasks such as generating dynamic web pages, database interaction, identity authentication, and push notifications.

However, hosting all of these processes on the server side leads to high latency since each request involving them must travel from the client side to the server every time. As a result, modern applications run more code on the client side to reduce this latency. For example, running scripts within the browser that make real-time changes to the content that a user sees is a common use case.

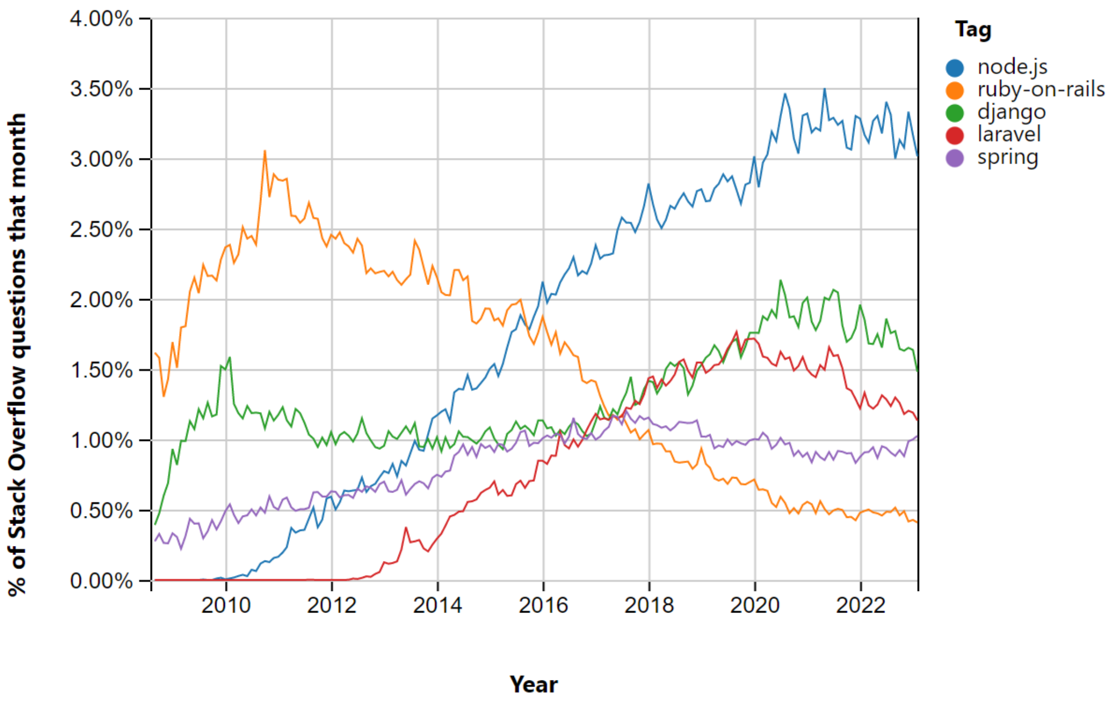
2.2. Web Development Frameworks

A software framework known as a web framework (WF) or web application framework (WAF) is specifically created to assist in the development of web applications, such as web services, web resources, and web APIs. The frameworks offer a consistent method for building and launching web applications on the Internet, while also refining the typical tasks involved in web development. Commonly, web frameworks come equipped with useful features, such as database access libraries, templating frameworks, and session management tools, which encourage using an existing code.

2.2.1. Server-Side Frameworks

Server-side frameworks are software frameworks designed to support the development and deployment of web applications. They provide a standard way to handle tasks such as database access, session management, and routing, which can streamline the development process and make it more efficient. The five most popular server-side frameworks at the moment are Node.js, Ruby on Rails, Django, Laravel, and Spring Framework. These frameworks are widely used due to their scalability, ease of use, and powerful features.

Figure 1. The popularity scale of the 5 most well-known server side frameworks

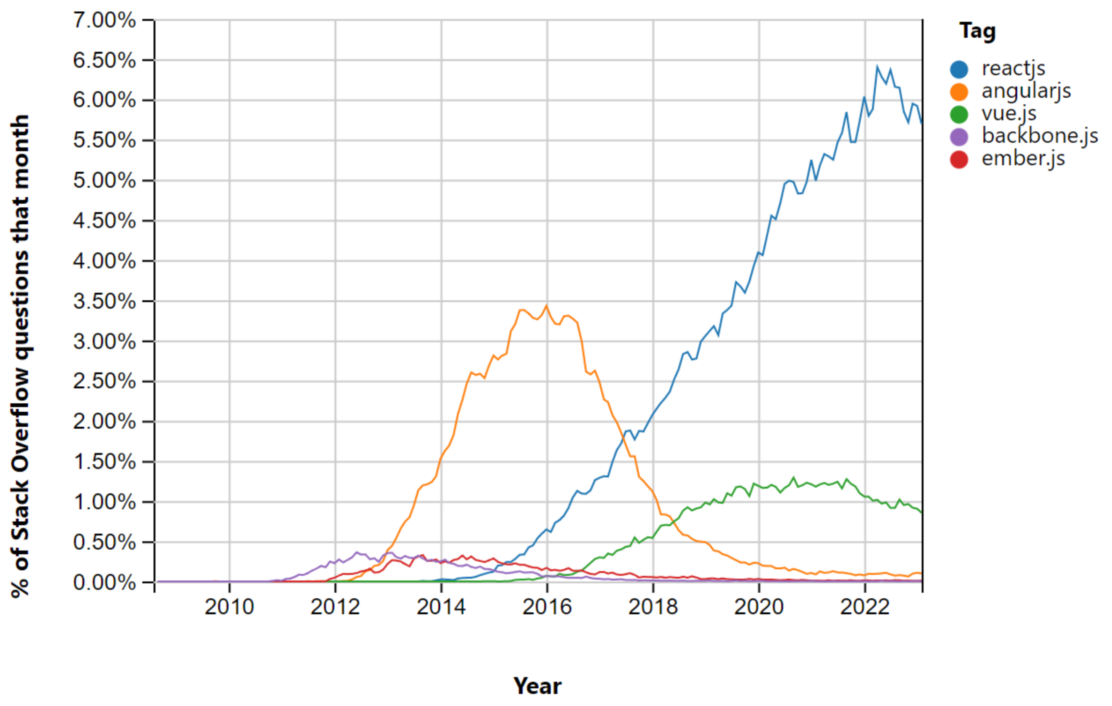


2.2.2. Client-Side Frameworks

Client-side frameworks for web development are a crucial tool for modern web developers. These frameworks provide a set of pre-built components and features that enable developers to create dynamic, interactive web applications with ease. They allow developers to build complex applications that run in the browser, without the need for server-side rendering, which can greatly improve the performance and user experience of web applications.

These frameworks are designed to be user-friendly, making them accessible to developers of all skill levels. They offer a wide range of functionalities such as routing, state management, and data binding, which help developers to build web applications efficiently and effectively. In addition, many client-side frameworks are open-source, meaning that they are constantly being updated and improved by the community

Figure 2. The popularity scale of the 5 most well-known client side frameworks



2.2.3. CSS Frameworks

CSS frameworks are pre-built collections of CSS rules and styles that are designed to simplify and speed up the process of designing and developing websites. They are essentially a set of predefined styles that can be applied to a web page to achieve a consistent and professional look and feel.

CSS frameworks are used by web developers to create responsive and visually appealing websites quickly and efficiently. CSS frameworks also typically include a grid system that allows developers to create responsive layouts that adapt to different screen sizes. They also provide a standardized set of UI elements, such as buttons, forms, and navigation menus, that can be easily customized to match the specific needs of a website.

2.3. PlugShare API

PlugShare API is an application programming interface (API) that provides developers with access to a comprehensive database of electric vehicle (EV) charging stations around the world. It is used by developers to build applications and services that enable EV drivers to find and navigate to charging stations, as well as to share information about charging stations with others in the EV community.

The PlugShare API provides developers with a wealth of data on charging stations, including their location, availability, charging speed, and compatibility with different types of EVs. This data can be integrated into a variety of applications, such as mobile apps, websites, and in-car navigation systems, to provide EV drivers with real-time information about nearby charging options.

3. Expected Achievements

This project is designed to answer the main problem of a website programmer when he approaches a new project - the question of which frameworks he should use. Another goal of the project is to provide a beginner programmer with the advantages and disadvantages of the web development technologies that exist today.

The result of this project will be a web application designed for electric vehicle owners, the application will allow vehicle owners to locate nearby charging stations around their current location and receive information for a specific position.

The project's main criteria for success are the implementation of our entire web application in the most efficient way using appropriate frameworks.

4. Research / Engineering Process

This section outlines the research and work processes we undertook during the semester to achieve our project goal. It is divided into two parts, one that describes the process and another that describes the product of our project.

4.1. Process

In this project, we face the problem of choosing the best frameworks for a web application.

We began our engineering process by researching existing academy level articles that explain the differences between the existing popular frameworks. These articles will help us to understand the pros and cons of the frameworks and to offer a solution that fits our application.

In order to accomplish this project goals, we went over the related background and learned the differences between the frameworks, to do so we have conducted research on various existing frameworks. We also had to research the API we wanted to use in our project.

4.2. Product

This section provides a high-level description of the software development process for the “EVolve” web application. Section 4.2.1 describes the “Evolve Design Process” and how the team defined the product and its needs. Section 4.2.2 outlines the “Choosing Framework Process” and how the team researched existing frameworks and decided to focus on JavaScript-based frameworks such as React, Angular, Node.JS, Bootstrap, and Tailwind. Sections 4.2.2.1 and 4.2.2.3 provide comparative analyses of React and Angular, as well as Bootstrap and Tailwind, respectively. Section 4.2.3 shows the Use Case Diagram, while section 4.2.4 presents the Activity Diagram. Section 4.2.5 displays the Graphical User Interface (GUI) of the “EVolve” web application and describes its main features.

4.2.1. Evolve Design Process

The process of choosing the best fit framework for our application began by defining the product we wish to develop and its needs. We first started by looking for examples for a similar web application, surprisingly we found that there is only one website that provides a service for finding charging stations and allows you to navigate to a specific station, we found that most of the websites just providing a map with the locations of the stations without the option to navigate, this option exist just in applications for smartphones.

The second phase of our design process was to define the process that the user will have to do to use our application. We define this process using Use Case and Activity diagrams [4.2.3 , 4.2.4]. Afterwards we designed the Graphical User Interface [4.2.5].

The last phase of our process was to research the existing frameworks and understand the advantages and disadvantages of each one, then we thought about what is the best framework for our needs. We decided to try two combinations and make checks to see which will perform better.

4.2.2. Choosing Framework Process

We started this process of choosing the framework by first researching existing frameworks. We discovered that an enormous number of 10,000-15,000 frameworks currently exist today, because of that we decided to focus on JavaScript based frameworks thus allowing dynamic modification of the website content and are easy to use in terms of implementation. That decision helped us to reduce the number of frameworks to about 750. Then we decided to look at what are the most popular JavaScript based frameworks that developers use today. After careful thinking we decided to focus our research on: React and Angular (using Bootstrap and Tailwind for CSS) as our client side frameworks, Node.JS (using MongoDB) as our server side framework.

4.2.2.1. Comparative analysis of React and Angular

Angular:

Angular has extensive possibilities when it comes to the implementation of the front-end area of the application.

The latest version uses the TypeScript language,which is a kind of overlay to the JavaScript language.It introduces typing support and other things characteristic of well-known programming languages.

Version 1 and lower were poorly evaluated by programmers in terms of code syntax. Here, there is a clear division into components, templates, directives and services.

The characteristic MVC (Model View Controller) terminology known from previous versions of the tool has been abandoned.

The advantage of Angular is still the possibility of bilateral data binding. It allows you to update the component's object field through the user interface and vice versa - the variable updated in the component will also change its value on the presentation layer. This process is automated by the framework and does not require any programmer intervention to invoke it.

A simple HTML-based template system makes it easy to mix directives with familiar hypertext tags, making it easy to adapt the application to adapt content to different devices and screen diagonals (responsive web design).

Methods implemented in the framework can be seamlessly tested using Javascript automated testing tools such as Karma.

Additionally, for AngularJS, it is possible to create a UML profile to design model-driven applications. The project implemented in this way requires only filling in the form generating the ready application. 87% of the code coincides with the established concepts, which reduces development time.

React:

React is also classified as a JS library. However, it has very specific features of the framework. Its order and hierarchy mean that the programmer has a predetermined way of implementing his application in this tool. As in Angular,components play an important role.

Components are objects of a class that inherits from React.Component. Each module must contain a render() function that draws the component in the document tree. It usually includes templates and other subcomponents. The latter are arranged in compositions. Starting from the parent component, ending with subcomponents of small application controls. The form of a component may contain its state elements and props.

Parameters are immutable for a given module, they are its properties, which can only be changed by the parent component, if it exists. The parameter may be, for example, the font size of the inscription, the number of possible ratings on the rating list. State is a set of current, dynamic values stored in the component body. As an example, you can give the currently viewed page of an e-book or even the state of a chess game.

Writing templates and expressions in JSX syntax can replace writing code directly in JavaScript. This allows you to avoid complicated structures composed of nested React.createElement methods. This makes the code readable, easier to modify and review.

The syntax of JSX resembles JavaScript combined with HTML. Views are listed as standard tags (plus component tags), but such structures can be assigned to variables, arrays, and object fields.

Factor	Angular	React
Powered By	Google	Facebook
Release Year	2010	2013
Definition	Framework	Library
Templating	HTML + TypeScript	JavaScript + JSX

Binding	Two-way	Uni-directional
DOM	Regular DOM	Virtual DOM
Popularity	Popular	Very Popular
Documentation	Good	Excellent
App Size	Large	Small
Performance	Good	Excellent
UI Rendering	Client-side	Both client and server-side
Price	Open Source	Open Source

4.2.2.2. Comparative analysis of React and Preact

React and Preact are two popular JavaScript libraries used for building web applications. Both libraries provide developers a flexible framework for building user interfaces. However, there are some differences between React and Preact in terms of size, and performance. We will compare and analyze these differences to help developers make informed decisions when choosing between the two libraries.

Syntax

React and Preact have similar syntax, as Preact is a lightweight alternative to React. Preact has a smaller API than React, but it is designed to be compatible with React's API. Therefore, developers can use the same syntax they would use in React when working with Preact.

Size

One of the key differences between React and Preact is their size. Preact is a lightweight library, with a file size of around 3KB, while React's file size is much larger, at around 100KB. This makes Preact an excellent choice for developers who need to keep the size of their application to a minimum.

Performance

Performance is another important factor to consider when choosing between React and Preact. Preact has been designed to be faster than React, with a smaller footprint and faster rendering times. This is because Preact uses a smaller API and has less overhead than React.

Conclusion

In conclusion, React and Preact are both excellent libraries for building web applications, but there are some differences between them that developers need to be aware of. Preact is a lightweight alternative to React, with a smaller file size and faster rendering times. However, React has a larger API and is more feature-rich than Preact. Ultimately, the choice between React and Preact will depend on the specific needs of the developer and the requirements of the application being built.

Factor	Preact	React
Size	3kb	100kb
Build time	Faster	Slower
Syntax	Same as React	Same as Preact
DOM	Virtual	Virtual
Ecosystem	Small	Large
Performance	Good	Excellent
Server-side rendering	Supported	Supported
Community	Small	Large

4.2.2.3. Node.JS pros and cons

Node.js is a server-side JavaScript runtime environment that has gained popularity among developers due to its unique features and capabilities. As with any technology, there are pros and cons associated with Node.js, particularly from a programmer's perspective.

One of the significant advantages of Node.js is its speed, which is achieved through its reliance on Chrome's V8 JavaScript engine. This engine is renowned for its efficiency and speed, allowing Node.js to handle a high volume of requests quickly and efficiently. As such, it is a well-suited technology for building scalable web applications that require fast response times.

Another significant advantage of Node.js is that it allows developers to leverage their existing JavaScript skills for server-side development. This reduces the learning curve and time required to develop server-side applications and may be particularly advantageous for developers with expertise in client-side web development.

Node.js also boasts a large and active community of developers who contribute to the development of open-source modules and packages. This community-driven approach makes it easier for programmers to find and use third-party tools and can significantly reduce development time and effort.

However, Node.js is not without its disadvantages. One of the most significant is its single-threaded nature, which can limit its ability to handle computationally-intensive tasks effectively. As such, Node.js may not be the best choice for CPU-intensive applications, such as those involving image processing or scientific computing.

In addition, Node.js uses an asynchronous programming model, which may be challenging for some programmers to understand and work with, particularly those accustomed to synchronous programming models. Furthermore, as a relatively new technology, Node.js may have fewer resources available for troubleshooting and debugging issues than more established programming languages and frameworks.

In conclusion, Node.js offers significant benefits for programmers, including its speed, ease of use, and a large and active community of developers contributing to open-source packages. Nonetheless, it is important to be aware of the potential limitations, such as its single-threaded nature and asynchronous programming model, when deciding whether to use Node.js for a particular project.

4.2.2.4. Comparative analysis of Bootstrap and Tailwind

Bootstrap

Bootstrap, which was launched in 2011, has become the most widely used CSS framework within a short period. It gained recognition for its intelligent CSS framework design. Bootstrap is a component-based framework and follows an object-oriented CSS approach. This framework includes a collection of pre-built components that can be utilized to rapidly design standard websites. Additionally, Bootstrap maintains consistency across multiple browsers and devices, making it simple to learn and customize for website development, particularly for those familiar with CSS.

Tailwind

Tailwind CSS was introduced in 2017 and quickly gained recognition for significantly expediting development. It is classified as a Utility-First CSS Framework and provides unparalleled customization capabilities, allowing each website component to be styled as per Tailwind UI requirements. Unlike other frameworks, Tailwind CSS does not offer pre-made components for design purposes. Instead, users can create their own custom components using utility classes. As opposed to writing extensive CSS code, this framework emphasizes creating several classes for HTML elements. Tailwind CSS is exceptionally versatile and can easily transform the appearance and behavior of website elements.

Comparison and conclusion

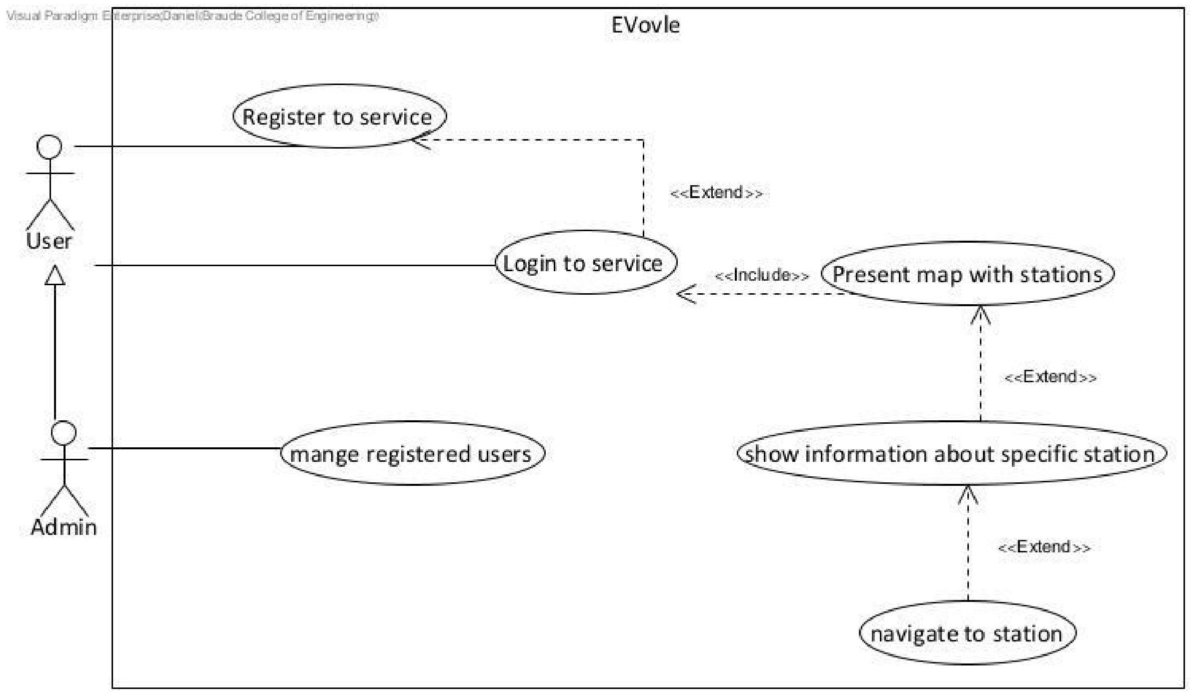
In terms of performance, there are notable differences between Bootstrap and Tailwind CSS. Bootstrap requires four primary files to be integrated into the project, occupying approximately 300kb of space. On the other hand, Tailwind CSS only requires a style sheet, which consumes around 30kb of space.

The build time for Bootstrap is generally lengthier, around 160 seconds, due to the amount of overhead code it contains. In contrast, Tailwind CSS's built-in utility libraries help shorten the build time, reducing it to approximately 8 to 10 seconds.

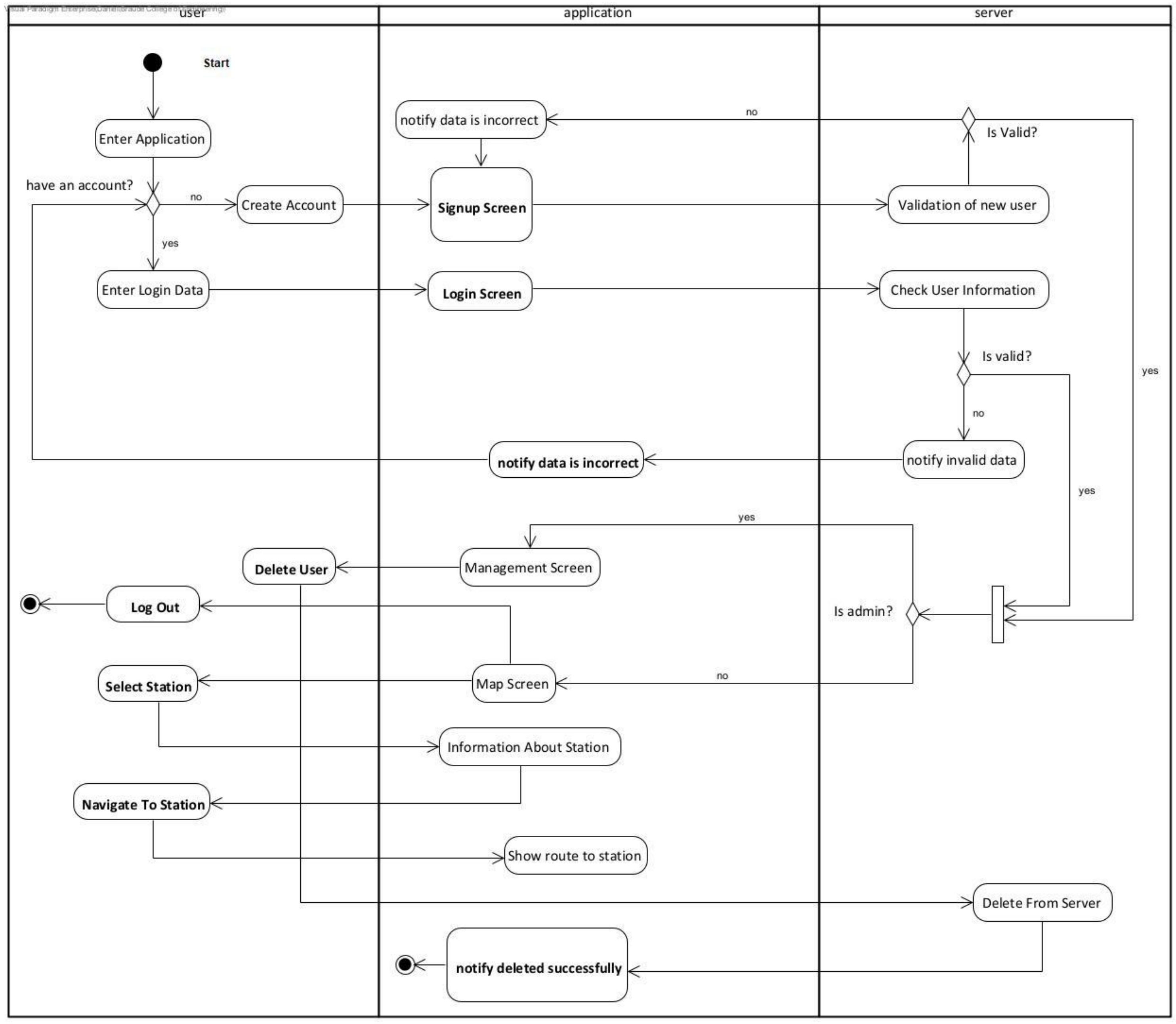
Bootstrap has been around for approximately nine years and has an extensive community of millions of developers with a plethora of tools and forums. Tailwind, on the other hand, is a relatively new framework that is gradually gaining popularity, which means it still has plenty of room for growth in terms of community and developer tools.

Ultimately, the choice between Bootstrap and Tailwind CSS depends on the project's requirements and personal preferences. If working as a new backend developer or a "Full-Stack" developer, Bootstrap may be the optimal choice. However, for front-end developers seeking greater customization in designing custom web applications, Tailwind CSS may be the better option.

4.2.3. Use Case Diagram



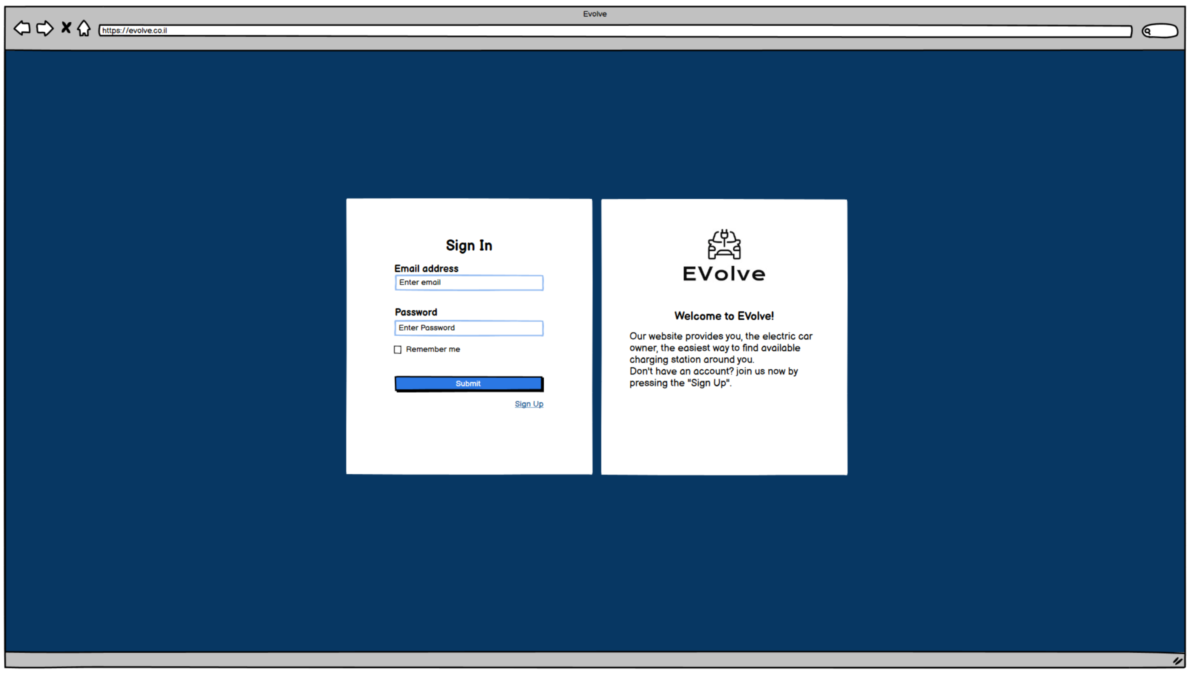
4.2.4. Activity Diagram



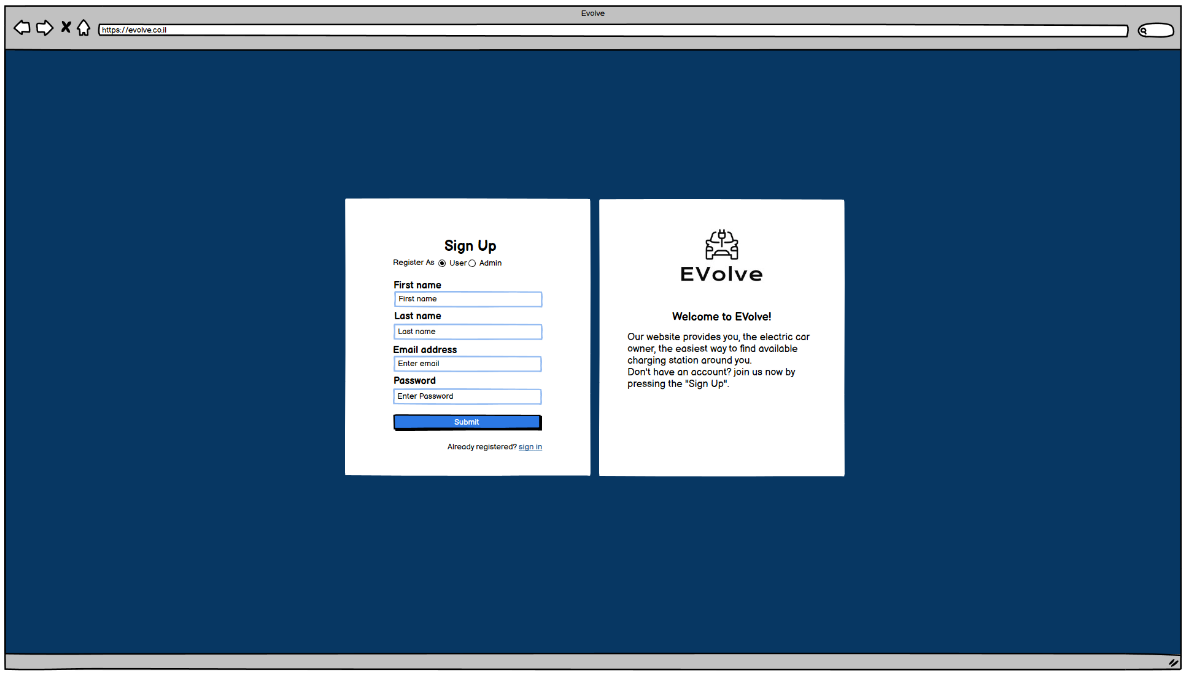
4.2.5. Graphical User Interface (GUI)

This section demonstrates a mock of the GUI and describes its main features.

<u>Main Screen:</u> In this screen the user can choose between registration and

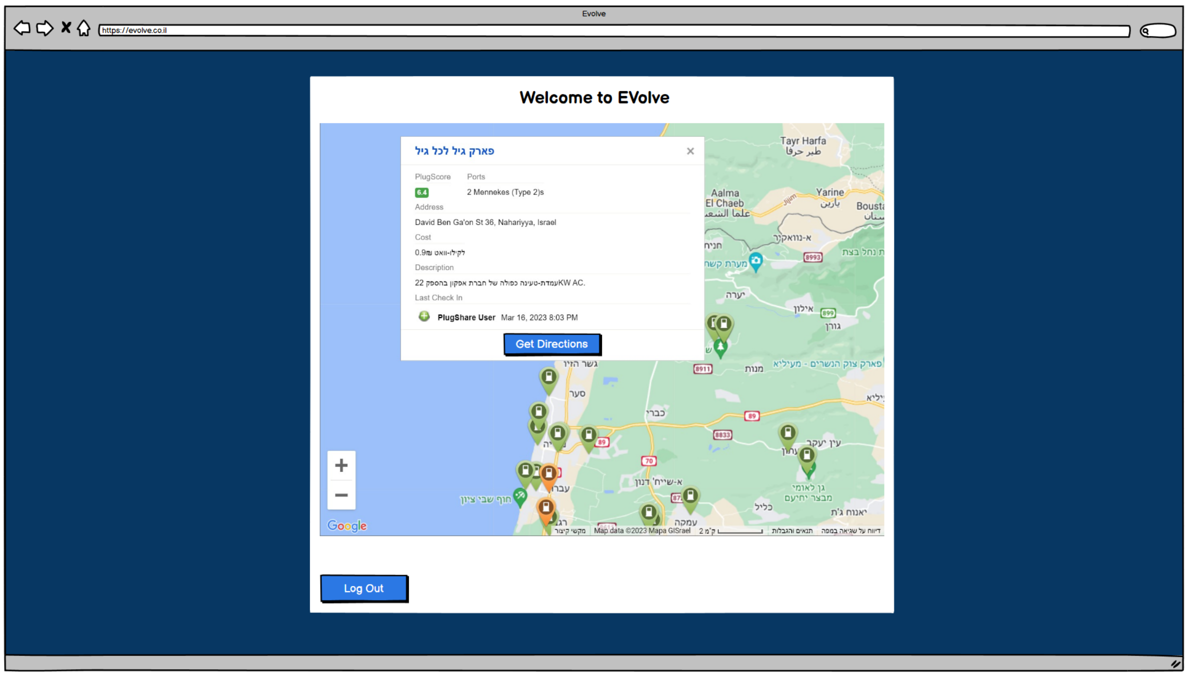


Our website provides you, the electric car owner, the easiest way to find available charging station around you.
Don't have an account? join us now by pressing the "Sign Up".



Our website provides you, the electric car owner, the easiest way to find available charging station around you.
Don't have an account? join us now by pressing the "Sign Up".

Map Screen: In this screen the user can see the stations around his current position, choose a specific station and get information about it or start a navigation to it.



פארק גיל לכל גיל

PlugScore: 6.4

Ports: 2 Mennekes (Type 2)s

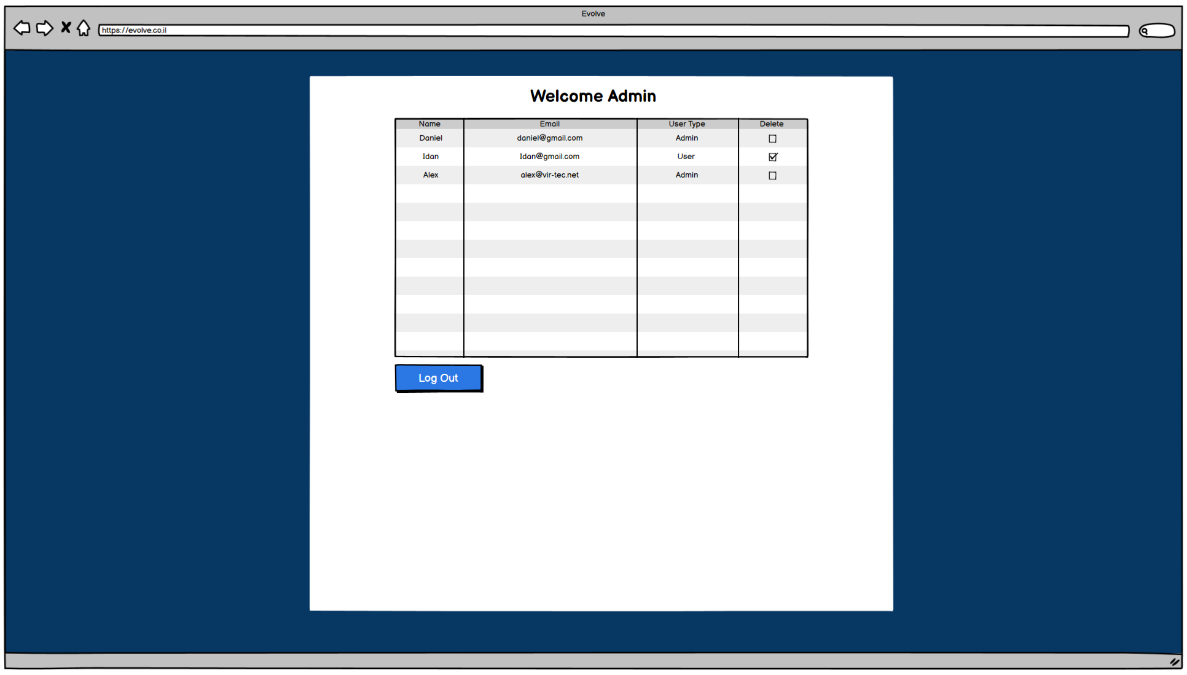
Address: David Ben Ga'on St 36, Nahariyya, Israel

Cost: 0.9₪ לקילו-וואט

Description: עמדת-טעינה כפולה של חברת אפקון בהספק 22KW AC.

Last Check In: PlugShare User Mar 16, 2023 8:03 PM

Manage Screen: In this screen the admin can manage the registered users, the admin can see the users information such as email address and first/last name and to delete users from the database.



Name	Email	User Type	Delete
Daniel	daniel@gmail.com	Admin	☐
Idan	Idan@gmail.com	User	☑
Alex	alex@vir-tec.net	Admin	☐

5. Evaluation / Verification Plan

The technique of System Testing (ST) is employed to evaluate the whole system's compliance with the specified requirements. ST involves testing the system functionalities from an end-to-end perspective. This project utilizes the ST approach to verify the desired operation and accuracy of the software presented. The use of ST ensures that the software performs as intended. Additionally, we conduct white box tests to assess the input's integrity by examining the methods and functions that require an input.

The table presented below provides a summary of the tests conducted in this project.

Test No.	Test Subject	Expected Result
1.	User enters an email in the wrong format and press submit.	Pop up with an error message.
2.	User tries to login with the wrong user name.	Alert message “wrong username or password” appears.
3.	User tries to login with the wrong password.	Alert message “wrong username or password” appears.
4.	User tries to register as Admin with the wrong secret key.	Alert message “Invalid Admin” appears
5.	User tries to register with the wrong email format.	Pop up with an error message.
6.	An already registered user tries to register using the same email address.	Alert message “Something went wrong” appears.
7.	New user fills all the sign up forms correctly and presses the “sign up” button.	Alert message “Registration successful” appears.
8.	User enters his email and password	Application moves to a “map screen” and

	correctly and press submit.	presents the closest stations to the user's location.
9.	User clicks on a station icon.	A floating window with information about the station appears.
10.	User clicks the “Get Direction” button in the station window.	Navigation to the station from the user's current location starts.
11.	Admin enters his username and password and presses “submit”.	Application moves to a “manage screen” and presents a table with all registered users information.
12.	Admin clicks on “delete” in the user's information row.	Alert message “are you sure you want to delete this user” appears.
13.	Admin clicks the “approve” button when he asks if he is sure he wants to delete a user.	Alert message “deleted” appears and table updates.
14.	Admin clicks the “Logout” button in the management screen.	Application moves back to the main sign in page.
15.	User clicks the “Logout” button in the map screen.	Application moves back to the main sign in page.

6. References

1. P. Jaiswal, S. Heliwal. Competitive Analysis of Web Development Frameworks. Available from: https://link.springer.com/chapter/10.1007/978-981-16-6605-6_53
2. Marin Kaluza, Bernard Vukelic. Comparison of front-end frameworks for web applications development. Available from: https://hrcak.srce.hr/clanak/294389%3f
3. Rishi Vyas, Comparative Analysis on Front-End Frameworks for Web Applications. Available from: https://www.ijraset.com/best-journal/comparative-analysis-on-frontend-frameworks-for-web-applications
4. Bartosz Miłosierny, Mariusz Dzieńkowski. The comparative analysis of web applications frameworks in the Node.js ecosystem. Available from:https://ph.pollub.pl/index.php/jcsi/article/view/2423
5. Tauheed Khan Mohd, Jordan Thompson, Amedeo Carmine, Grant Reuter. Comparative Analysis on Various CSS and JavaScript Frameworks. Available from: 10.17706/jsw.17.6.282-291
6. What do client side and server side mean?