Tampere University

Asbah Amjad Usmani

GUIDELINES FOR SELECTION OF WEB DESIGNING TOOL & FRAMEWORK FOR WEB FRONT-END APPLICATION

Faculty of Information Technology and Communication Sciences (ITC)
M.Sc. Thesis
Examiner: Dr. Timo Nummenmaa
Examiner: Dr. Niko Siltala
November 2023

ABSTRACT

Asbah Amjad Usmani: Guidelines for Selection of Web Designing Tool & Framework for Web Front-end Application
M.Sc. Thesis
Tampere University
Master's Programme in Information Technology
November 2023

As the digital landscape evolves, the development of web front-end applications demands a judicious selection of web designing tools and web development frameworks. The continuous transformation of technologies, user expectations, and industry standards underscores the critical importance of selecting appropriate tools to navigate the complexities of web development. This research endeavours to close the divide between theoretical concepts and real-world application, providing a structured approach for developers and decision-makers manoeuvring through the complexities of web front-end development.

The purpose of this thesis work is to propose a strategy that can help in the selection process of web designing tools and development frameworks for web front-end application. The decision-making process is recognized as a pivotal element that holds the potential to substantially influence development efficiency, end-user experience, and overall project success. By meticulously aligning project goals with the unique strengths of each tool and framework, developers can craft and deliver innovative, user-centric web applications.

To accomplish this objective, this thesis answers the questions on which methodology should be adopted in choosing a web designing tool for web front-end applications and what considerations are crucial when selecting a development framework for web front-end applications. To show the usefulness of proposed approach, it was used in the selection of web design tools and development frameworks for a front-end application named Matchmaking Graphical User Interface (MM GUI) application. For MM GUI application, wireframes and prototypes are created as initial design and then final web application was developed.

The result of this thesis work includes guidelines for selection of web designing tools, JavaScript frameworks and implementation of MM GUI application using proposed guidelines.

Keywords and terms: Web design tools, JavaScript frameworks, GUI.

PREFACE

This thesis was written to meet the graduation criteria for the Master's Program in Information Technology at Tampere University. This work has received funding from the EU Hz-2020 AI-Regio project (GA. 952003) and I commenced working on the thesis while working in the project. I express my gratitude to my supervisor, Niko Siltala, for providing required guidance and support throughout the entire process of composing this thesis. Additionally, I extend my thanks to examiner Timo Nummenmaa for his assistance and support before embarking on the writing of this thesis.

Espoo, 24th November 2023

Asbah Amjad Usmani

Contents

List of Symbols and Abbreviations

GUI	Graphical User Interface
HTML	Hyper Text Markup Language
CSS	Cascading Style Sheets
JS	JavaScript
OS	Operating System
UI	User Interface
iOS	iPhone Operating System
API	Application Programming Interface
SPA	Single Page Application
MPA	Multi-Page Application
SSR	Server-Side Rendered Application
PWA	Progressive Web Application
CMS	Content Management System
PHP	HyperText Preprocessor
UX	User Experience
PDF	Portable Document Format
DOM	Document Object Model
AJAX	Asynchronous JavaScript and XML
XML	Extensible Markup Language
JSX	JavaScript XML
RxJS	Reactive Extensions for JavaScript
CLI	Command Line Interface
MM	Matchmaking
JSF	JavaScript Framework
REST	Representational State Transfer
MVC	Model View Controller
MVVM	Model View View Model
SEO	Search Engine Optimization
SERPs	Search Engine Result Pages

1. Introduction

Web front-end applications are an integral part of modern-day web development. These applications are also known as a client-side application because it is a type of software program that runs on the user's device such as a mobile app, web browser, or desktop application. Web front-end applications interact with the user directly, often using a graphical user interface (GUI). These applications are designed to perform specific tasks.

The process of developing a web front-end application involves different stages which includes careful planning, designing, development, testing, and ongoing maintenance to ensure that the application delivers the desired user experience and meets the user requirements. It also requires selection of tools and technologies to complete these stages successfully. It is important to invest proper time on selection of tools and technologies required for web front-end application.

The designing of an application is a creative process, and it demands experimenting with different designs and layouts until you find out the best design for your application. The designing stage of a web front-end application is essential for creating a positive user experience, ensuring accessibility, and building a scalable and adaptable website. There are variety of designing tools, selecting a design tool for a web application can be overwhelming, as there are many options available, but it is important to make a right choice to avoid any complexities.

After getting the designs of a web front-end application, the next stage is development of web front-end application. The front-end code of the application is developed by utilizing the chosen programming language and framework. There are several programming languages that can be used for web front-end development. JavaScript is the most popular language for web front-end development, but there are several other programming languages that can be used as well.

For the coding of web front-end application, there are wide range of JavaScript libraries and frameworks available which are used by developers because they offer several benefits that make it easier and faster to build complex web applications. These frameworks solved many problems of developers, at the same time choosing the appropriate technology stack for your front-end application can be a daunting task, but it's essential to ensure that your project is successful. There is no rule of thumb for the selection of these frameworks, and this makes it confusing for the developer.

D'Souza et al. [1] in their article “Enabling the Generation of Web Applications from Mockups.” states the importance of mockups in elicitation and validation of requirements for web application. This research work addresses the limitation by presenting a method that extracts the database schema and logic of web applications from the information embedded within mockups. Specifically, the method analyzes the structure and relationships of widgets in the mockup to gather insights on organizing data and controlling the flow of the web application. The proposed method serves as the foundation for a tool supporting the generation of web applications adhering to the model-view-controller architectural pattern. The effectiveness of the tool has been evaluated through the involvement of several end users in the development of web applications across diverse domains.

Maurya et al. [2] in their article introduces an innovative approach for rapidly generating and validating interactive product behaviors during the early stages of design. Traditional methods often involve consideration of spatial, behavioral, and tangible elements, with functional prototyping performed later in the design process. However, outcomes are influenced by factors such as designer familiarity with tools, prototype fidelity, and design iteration frequency, limiting thorough exploration and creativity in the early stages. This research addresses these limitations by employing Mixed Reality for concept generation, establishing tool requirements, and presenting a proof-of-concept use case. Despite challenges related to tool familiarity, the study demonstrated a substantial increase in the use of iterative concept design behavior.

There has been some work done for evaluation of JavaScript frameworks and libraries. Gizas et al. [3] performed comparative evaluation on different JavaScript frameworks. For evaluation, authors defined some common functions such as Asynchronous JavaScript and XML (AJAX), Document Object Model (DOM) manipulation, loading utilities, and functions for event handling. In this work, authors also performed quality and validation tests. Graziotin and Abrahamsson [4] performed a qualitative study with experienced web developers to make a model for the comparison of JavaScript frameworks. This model suggests three criteria: documentation, community, and the pragmatics of a JavaScript framework. In another study [5], different factors and responsible actors are discussed which can lead to the adoption of JavaScript frameworks.

Stefi [6] investigated what motivates developers to adopt software components. Stefi assembled 142 answers from software developers. The results show that performance expectancy, the not-invented-here bias, and social influence are key factors in the adoption of software components.

Zhang and Chung [7] stated in their article that web development can be very complicated without an appropriate architecture, framework, and application model. They proposed a mockup-driven fast prototyping methodology (MODFM) for the development of Web applications.

To conclude the examination of relevant literature, these research works provide a foundation for this thesis work and highlight the importance of web designing and web development frameworks. There was not much work done on the approach to be considered while selecting a web designing tool and development framework. There is a need to establish comprehensive guidelines to ensure that development teams make informed decisions aligned with project requirements, industry standards, and long-term sustainability. Research in this domain is necessary to bridge the existing gap between theoretical guidelines and practical outcomes. By conducting research on this topic, the development community can gain a deeper understanding of the decision-making dynamics. This research is crucial for empowering developers and decision-makers to navigate the ever-evolving landscape of web front-end development with confidence and strategic foresight.

In this thesis work, these are the research questions to be answered:

- What approach to follow for the selection of web designing tool for web front-end application?
- What factors to consider during selection of a development framework for web front-end application?

This thesis work centers around the design and development stages of web front-end application. The designing stage is very critical and helps in visualizing the results before starting the development work. It helps in restructuring or debugging in the early stage of project and saves time, money, and effort. Therefore, this thesis work will focus on software designing tools and methods used for web front-end application, and it will investigate different parameters and aspects to consider for the selection of software designing tools. This investigation will provide an approach for selection of suitable design tools for web front-end application. The other focus area is to find the best ways which will help developer in the selection of technology stack for the development of web front-end application. This thesis focuses on exploring the ways which will help the developer in the selection of standard and stable technologies for the development of web front-end application.

For this thesis, descriptive and analytic research was conducted to formulate the guidelines for selection of web designing tools and JavaScript frameworks in section 3 and 4 respectively. To recognize and analyse factors that affect the web designing and development, a study was conducted on some relevant work which addressed requirements, architecture, resources, types of web designs and types of web applications. Moreover, the analytical approach was used to explore several web designing tools and development frameworks and their strengths and weaknesses were highlighted. This helped in understanding solutions which these tools and technologies can provide to web applications. These solutions helped in formulating a strategy for selection of design tools and development frameworks.

In chapter 5, a descriptive method has been performed to show the usefulness of proposed approach. This proposed approach is applied for the implementation of a real-world project named Matchmaking Graphical User Interface (MM GUI) application. This provides a qualitative understanding of how proposed approach can be practically implemented. This section explains the project, the selection criteria used, the tools and frameworks chosen, and the outcomes. The completion of project without any complexities and results obtained after using this approach show the usefulness of the proposed approach on a web front-end application.

2. Background

This thesis work is formulated around the selection of web designing tools and development framework used for the web front-end applications. In this chapter, important concepts web front-end application will be considered in detail such as web designing, JavaScript, and JavaScript UI frameworks.

2.1. Web Front-end Application

Web front-end application also known as client-side applications are designed to provide an interactive user experience and enable users to perform various actions, such as submitting forms, inputting data, and navigating through different pages of a website. Some examples of client-side applications include social media platforms, web-based email clients, online banking systems, and e-commerce websites. These applications are essential for delivering a smooth and intriguing user experience on the web and have become a fundamental part of modern-day web development [8].

The web applications are in trend, and it is understandable because of various advantages [9]. Web front-end applications can be accessed from anywhere in the world. User needs only an internet connection and a web browser. Generally, there is no need for installation. This makes them ideal for businesses that want to reach a global audience or offer remote services. Web front-end applications are popular with software developers and businesses because they can be maintained, updated, and developed remotely without any need to install (or reinstall) them. The development and deployment of these applications is not dependent on the operating system of users' computers. There is no need to create separate versions of applications for Microsoft Windows, GNU/Linux, OS X, and other operating systems. These applications can be developed and deployed quickly and efficiently, reducing the time and cost associated with traditional software development. These applications enable developers to leverage the latest technologies and design trends to create innovative and engaging user experiences. These applications are scalable, and this helps in meeting the changing needs of the business or user base. Nowadays, people are familiar with the use of browser and browser terminologies such as forward, back, home, bookmarks, submit buttons, and so forth. With this knowledge, many barriers are reduced for the web application user. This results in widespread adoption of web applications.

In this section, architecture and different types of web front-end application will be discussed.

2.1.1. Architecture

Web application architecture refers to the structure, design, and organization of components that make up a web application. It encompasses various aspects, including the interactions between different layers, the technologies used, and the overall system's performance, scalability, and maintainability.

Identification of different types of architecture for web front-end applications helps in making informed decisions that impact the long-term success, performance, maintainability, and user experience of application. It's a foundational step that sets the direction for development process and ensures that front-end aligns with overall project goals. There are several types of architectures for web front-end applications, each with its own advantages and use cases. Some of them are discussed in detail below:

- **Model View Controller (MVC) Architecture:**

In MVC, application is divided into three distinct components: the Model, the View, and the Controller. This separation helps in maintaining a clear and organized codebase, making it easier to manage and update the application over time [10]. MVC is widely used in web development, including front-end and backend frameworks.

The Model embodies both the data and business logic of the application, taking on responsibilities such as state management, data retrieval and storage, validation, and enforcement of business rules. The Model component actively responds to requests from the Controller to update or retrieve data.

The View assumes the role of rendering the user interface and showcasing data to the user. It receives data from the Model and formats it in a manner that is both comprehensible and visually pleasing. View is primarily focused on the presentation of data rather than its acquisition or modification. Views can include HTML templates, CSS styles, and sometimes a bit of conditional logic for rendering dynamic content. Each View corresponds to a specific part of the user interface, such as a page, a widget, or a component.

Serving as a mediator between the Model and the View, the Controller receives user input from the View, processes it, collaborates with the Model to update or retrieve data, and subsequently ensures the corresponding adjustments in the View. The Controller bears the responsibility of managing user interactions, making decisions based on input, and overseeing the seamless flow of data between the Model and the View.

- **Flux Architecture:**

Flux is an architectural pattern used for managing the flow of data in web applications, particularly in front-end development. It was introduced by Facebook to address the challenges of managing complex state and data flow in applications built with the React framework [11]. Flux enforces a unidirectional data flow. The key concepts of Flux are actions, dispatcher, stores, and views.

Actions represent events or user interactions that occur in the application. They are simple objects that describe what happened. An action typically includes a type and additional data (payload) relevant to the event. For example, an action might represent a user clicking a button or submitting a form.

The Dispatcher is the central hub that receives actions and dispatches them to registered stores. It ensures that actions are processed in a consistent order and avoids race conditions in state updates. Actions flow through the Dispatcher to reach the appropriate stores.

Stores hold the application's state and business logic. They respond to dispatched actions and determine how the state should be updated. Stores are responsible for maintaining their own data and notifying views when the state changes. Each store typically manages a specific domain of data.

Views are responsible for rendering the user interface based on the data provided by the stores. When a store's data changes, views are notified and re-rendered accordingly. Views do not directly modify the state; they send actions to the Dispatcher to request changes.

- **Redux Architecture:**

Redux serves as a state management library tailored for JavaScript applications. While it is frequently employed in conjunction with frameworks such as React, it is also adaptable to other view libraries. Drawing inspiration from the Flux architecture, Redux is crafted to handle an application's state in a centralized and predictable fashion. The key concepts of Redux are store, actions, reducers, dispatch, and selectors [12].

At the core of Redux lies the store, serving as the repository for the entire state tree of your application. In Redux, the state mirrors the current data and UI state of your app. The store, a straightforward JavaScript object, encompasses the complete state of the application and is strictly read-only. Modifications can only occur through dispatched actions.

Actions, in the context of Redux, are simple JavaScript objects delineating events or user interactions that prompt changes in the application's state. Each action necessitates a type property, specifying the nature of the action. Additional information, referred to as the "payload," can be included to provide context for the action.

Reducers, as pure functions, have the role of specifying how the state undergoes changes in response to actions. When provided with the current state and an action as input, a reducer produces a new state. It's crucial to note that the existing state should remain unaltered, and instead, a new state object should be created. Redux employs these reducers to aggregate changes and update the state of the store.

The act of dispatching an action involves sending that action to the store. Utilizing the store's dispatch method, an action is passed as an argument, subsequently reaching the reducers. The reducers then ascertain how the state should be updated based on the type and payload of the action.

Selectors are functions that extract specific pieces of data from the state. They help in isolating the component from the actual structure of the state tree. By using selectors, components can access the required data without needing to know the entire state structure.

2.1.2. Types of Web Front-end Application

There are several types of web front-end applications based on their purpose and functionality. Different types of front-end applications have varying requirements and characteristics. Identifying the type of application helps in selecting the most appropriate tools, technologies, and frameworks that best suit those requirements. This ensures efficient development and optimal performance. Some common types of web applications are discussed here. For example, Single-Page Application (SPA), Multi-Page Application (MPA) etc.

SPAs are web applications that load and manipulate all the necessary content on a single web page, without requiring full page reloads during user interactions [13]. SPAs often utilize JavaScript frameworks such as Angular, React, or Vue.js to handle the rendering and data manipulation on the client-side. SPAs provide a seamless user experience as they don't require full-page reloads during navigation.

Unlike SPAs, Multi-Page Applications consist of multiple separate web pages. Each page represents a different section or functionality of the application. Navigation between pages typically involves full page reloads. MPAs are suitable for simpler applications or situations where search engine optimization (SEO) is a high priority. SEO is the practice of optimizing online content and websites to improve their visibility and ranking on search engine results pages (SERPs).

Server-Side Rendered Application (SSR) renders the web pages on the server and sends pre-rendered HTML to the client. This approach combines the benefits of server-side processing and dynamic client-side interactivity. SSR is useful for applications that require search engine visibility and initial page load speed, as the server provides the pre-rendered HTML.

Static websites are simple web applications that consist of static HTML, CSS, and JavaScript files. They are typically used for informational websites, blogs, and portfolios where the content doesn't change frequently. Static websites help user to find helpful information efficiently. Thus, the website has a general system structure. It usually contains Contact, About us, Home, and other information pages [14]. Static websites are easy to develop and deploy as they don't require server-side processing or complex backend infrastructure.

Dynamic websites generate HTML content dynamically based on user requests and interactions. They use server-side programming languages such as PHP, Ruby, or Python, combined with client-side technologies like JavaScript and CSS. Dynamic websites often retrieve data from databases or APIs and can provide personalized content to users.

Progressive Web Applications (PWAs) provide users an app-like experience. PWAs are the intersection between a mobile app user experience and a web interaction [15]. These applications are accessible through a web browser but can also be installed on the user's device like a native mobile app. PWAs offer offline capabilities, push notifications, and can be added to the home screen, providing an immersive and responsive user experience.

A summary of advantages and disadvantages of different types of web applications are listed below in Table 2.1:

Table 2.1 *Advantages and Disadvantages of different types of web applications*

Types of Web Application	**Advantages**	**Disadvantages**
SPA	Enhanced user experience Faster navigation Rich interactivity	Slower initial loading SEO challenges
MPA	SEO friendly Better compatibility with older browsers	Slower navigation Code Duplication Higher resource consumption
SSR	SEO friendly Faster initial loading	Development complexities Limited interactivity Caching challenges
Static Website	Fast loading Cost effective	Limited interactivity Maintenance challenges Scaling
Dynamic Website	Rich interactivity Flexibility	Development complexities Can be slower
PWA	Offline access App-like experience	Limited browser support Learning curve

2.2. Web Designing

Web designing is important stage for web front-end applications because it directly influences user experience, navigation, usability, and performance of application. A well-designed front-end can contribute to the achievement of project goals.

The process of web design involves the creation and design of the visual layout, user interface, and overall aesthetic of websites. It is a comprehensive field that encompasses both artistic and technical aspects in crafting a website. Web designers must consider factors such as accessibility, branding, user interface (UI), responsive design principles, and user experience (UX) to achieve success in their web design endeavors.

In web designing, there are some key components such as layout, color schemes, and so on. Color has a significant impact on how users perceive and interact with a website. A well-chosen color scheme can create a visually appealing and harmonious design that engages users and makes a positive first impression. It is important to choose an appropriate color palette that creates visual harmony and enhances the overall user experience. Colors evoke specific emotions and associations. By strategically selecting colors, web designers can create an emotional connection with users and elicit desired responses. The choice of colors affects the readability of text and content on a website. The right color contrast between text and background is crucial for ensuring legibility and accessibility for all users, including those with visual impairments [16].

Colors can be used to establish a visual hierarchy, guiding users' attention to key elements or important information on a web page. By using color contrast and intensity, designers can create emphasis and help users navigate through the content more easily. Colors can have cultural associations and symbolic meanings that vary across different regions and contexts. Understanding the cultural and psychological impact of colors is important for designing websites that resonate with the target audience. It's also crucial to ensure sufficient color contrast for accessibility and legibility, and to test the color scheme across various devices and screens to ensure a consistent experience for all users.

Layout is a fundamental aspect of web designing and holds significant importance and it includes determining the arrangement and placement of various elements on the web page, including headers, content sections, sidebars, navigation menus, and footers [17]. The layout provides the structure and organization of a web page, determining how different elements are arranged and positioned. It helps users understand the hierarchy of information. A well-designed layout enhances the user experience by guiding users through the content in a logical and intuitive manner. It ensures that the most important information is prominently displayed and easily accessible, reducing user frustration and increasing engagement. The layout contributes to the visual appeal and overall aesthetics of a website. A visually pleasing and balanced layout can create a positive first impression.

Responsive design has become crucial because of increasing use of various devices and screen sizes. A well-designed layout ensures that the website adapts and maintains its usability and visual appeal across different devices, providing a consistent experience for users [18]. The layout helps to prioritize and highlight key content. By carefully considering the placement and size of elements, designers can draw attention to important information, call-to-action buttons, or multimedia content, guiding users towards desired actions or conversions. The layout influences the readability of content. Clear visual hierarchy and grouping of related content help users quickly scan and find relevant information. Layout can be tested across different devices and conducting usability testing can help identify areas for improvement and ensure a user-friendly experience.

The primary purpose of typography is to ensure that the text on a website is readable and legible. A well-chosen and properly formatted typography enhances the user experience by making the content easy to read and comprehend. It includes choosing suitable font styles, sizes, and line spacing to enhance readability on various devices and screen sizes. It is essential to choose fonts that are legible for users with visual impairments or reading difficulties [16].

User Experience (UX) design specifies the process of enhancing user satisfaction and overall usability by improving the interaction between users and a product or service. It involves designing and optimizing the entire user journey, from the initial discovery and use of a product to the post-interaction experience [19]. UX design focuses on users' needs, goals, behaviors, and emotions to create products that meet their expectations and provide a positive experience. The objective of UX design is to ensure that the product or service is effective, efficient, enjoyable, and valuable to the users. For UX designs, it is required to conduct research to gain insights into the target users, their behaviors, needs, and pain points. This can involve methods such as user interviews, surveys, usability testing, and analytics. Usability Testing helps in improving UX design, it evaluates the usability of a product or service by testing it with representative users [20]. It helps identify issues and gather feedback to make iterative improvements. It ensures that the product or service is accessible to users with disabilities, allowing them to interact and engage with it effectively. UX design is an iterative process, where designs are continuously tested, refined, and improved based on user feedback and data analysis.

By focusing on user needs and providing a seamless and enjoyable experience, UX design aims to increase user satisfaction, engagement, and loyalty. It is applicable to various digital products and services, including websites, mobile apps, software applications, and interactive systems.

UI design, short for User Interface design, refers to the process of designing the visual elements, layout, and interactive components of a digital product or application. It involves designing the interface elements, including buttons, forms, input fields, dropdowns, and other interactive components, to ensure easy and efficient user interactions.

UI designers often work with design systems and create UI guidelines to maintain consistency throughout the product or application. Design systems establish a set of reusable components, styles, and guidelines that ensure a consistent and cohesive user interface. UI designers, UX designers, and developers collaborate closely to ensure that the visual and interactive elements align with user needs, usability principles, and technical feasibility [21].

2.3. JavaScript

JavaScript is a powerful and versatile language that offers a wide range of capabilities for building interactive, dynamic, and engaging front-end applications. Its popularity, extensive ecosystem, and continuous development make it a compelling choice for modern web development.

JavaScript is primarily a client-side programming language, meaning it runs on the user's web browser rather than on the web server. This allows it to interact with the HTML and CSS of a webpage, making it possible to create interactive elements and modify the page content dynamically [22]. JavaScript has a C-like syntax and supports a wide range of programming features, including variables, data types (such as numbers, strings, arrays, and objects), functions, conditionals (if/else statements), loops (for and while loops), and more.

One of the most common uses of JavaScript is manipulating the Document Object Model (DOM) of a webpage. The DOM represents the structure of an HTML document, and JavaScript allows you to access and modify its elements, attributes, and styles. This enables you to dynamically update the content and appearance of a webpage based on user actions or other events.

JavaScript provides the ability to handle events triggered by user interactions, such as mouse clicks, keyboard input, form submissions, and more. You can attach event listeners to specific elements on a webpage and define functions that will be executed when the events occur. JavaScript enables asynchronous communication with servers through techniques like Asynchronous JavaScript and XML (AJAX). It allows you to retrieve data from a server and update parts of a webpage without having to reload the entire page. Additionally, JavaScript can interact with various web APIs, such as the Document Object Model API, Web Storage API, Geolocation API, and others, to access browser functionality and external services.

JavaScript offers a variety of frameworks and libraries that broaden its capabilities and simplify web development. Some popular frameworks include Vue.js, React.js, and

Angular.js, which provide tools for building complex user interfaces and managing application state. Libraries like jQuery, lodash, and Axios offer utilities and abstractions to streamline common tasks.

While JavaScript is mainly associated with client-side development, it has also gained popularity for server-side development because of Node.js. It is a runtime environment that lets JavaScript to run on the server, enabling full-stack JavaScript development. With Node.js, developers can use JavaScript to build web applications on both the client and server sides.

2.4. JavaScript UI Frameworks

JavaScript UI frameworks are libraries or sets of tools that provide developers with the necessary components and functionality to build user interfaces for web applications. These frameworks streamline the process of creating interactive and responsive web interfaces by offering pre-built components, data binding, event handling, and other features.

JavaScript UI frameworks offer pre-built components, libraries, and tools that help in streamlining the development process. They offer a set of reusable and customizable UI elements, such as buttons, forms, icons, and grids, saving developers time and effort in creating these components from scratch [23].

JavaScript UI frameworks often handle browser compatibility issues by abstracting away the differences between browsers and providing consistent behavior across various platforms and devices. This allows developers to work on developing the application logic without concerning about browser-specific quirks. Modern web applications require responsive and interactive user interfaces. UI frameworks offer built-in features for handling user interactions, such as handling events, managing state, and updating the UI efficiently. They provide mechanisms for data binding, which ensures that changes in the underlying data automatically reflect in the UI, making it easier to create dynamic and reactive interfaces.

Component-based architecture is common in many JavaScript UI frameworks, where the user interface is divided into modular and reusable components. Components enclose their own structure, behavior, and styling, making it easier to manage and maintain complex user interfaces. This approach promotes code reusability, modularity, and separation of concerns. Popular UI frameworks like React.js, Angular.js, and Vue.js have vibrant communities and extensive ecosystems. This means there are numerous third-party libraries, plugins, and tools available that extend the functionality of the frameworks. Developers can leverage these resources to enhance their applications, access additional features, and address specific use cases.

JavaScript UI frameworks often optimize rendering and performance by utilizing techniques like lazy loading, code splitting, virtual DOM diffing, and caching. These optimizations help reduce unnecessary re-renders and improve the overall performance and responsiveness of web applications.

In short, JavaScript UI frameworks simplify the development of web interfaces, enhance productivity, improve user experience, and provide a solid foundation for building modern, responsive, and interactive web applications.

3. Selection of tools for Web designing

Web development projects often require the use of web designing tools to create interactive and functional prototypes that can serve as a blueprint for the final product. However, with the abundance of web designing tools available, selecting the most suitable one can be a challenging task for user.

Every web designing tool offers different features. Just because a tool was successfully used by others does not justify that it would be also suitable for your application. If a web designing tool is selected without proper consideration, it could introduce complications in the design process. The wrong tool may lack important features required for specific design tasks, it will restrict the designer's ability to create desired layouts or achieve certain effects. Designers may waste time trying to work around the limitations of the wrong tool, resulting in delayed project delivery and missed deadlines. If the tool doesn't meet project requirements, designers may need to invest in additional software or plugins to bridge the gaps, leading to unexpected costs. Using an outdated or unsupported tool can result in a lack of updates, security vulnerabilities, and difficulty finding help or resources when encountering problems. Therefore, tool selection is an essential part of web designing process.

By considering following factors, user can make informed decisions when choosing web designing tools to streamline their web design process.

- What are the requirements of web application?
- What are the resources available for web designing?
- What are the types of web design required for web application?
- What are the tools available for web design?
- How to use the selected tool efficiently?

The justification of these questions is given below.

3.1. Requirements

Requirements of an application are the detailed specifications and features that outline what a software application or system is expected to do. These requirements serve as the foundation and ensure that the end result satisfies the needs and expectations of its intended users [24]. Gathering and defining clear and comprehensive requirements is crucial for the success of any software project. It is also important for web designing process. Web designing is concerned with the functional, business, and user requirements of an application.

3.1.1. Functional Requirements

Functional requirements are concerned with the specific functionalities or features that the software must provide [24]. For web design of a front-end application, functional requirements focus on the features that the user interface (UI) should provide to users. These requirements describe how the front-end application should behave, what actions users can perform, and how the application responds to user inputs. Functional requirements serve as the foundation for the software design and development process.

Here are examples of some functional requirements for web designing of a front-end application.

1. Define the layout and placement of UI elements, such as buttons, forms, menus, navigation bars, and other interactive components.
2. Specify the visual design and style guidelines, including colors, fonts, and icons.
3. Describe how users can navigate through the application, including menu structures and page links.
4. Describe how data should be presented and visualized, such as tables, charts, graphs, and maps.

Functional requirements play a substantial role in the selection of a web designing tool as they provide valuable insights into the specific features and functionalities that the tool must possess to meet the project's needs.

Here are some key points discussed which explain how functional requirements can help in the selection of a web designing tool. For example, by comparing the functional requirements with the tool's feature set, you can determine if the tool has the capabilities needed to implement the required functionalities. It helps you ensure that the selected tool aligns with the project's objectives. Functional requirements often outline the desired user interactions and workflows. Evaluating a web designing tool against these requirements can help determine if the tool's workflow and user interface are conducive to achieving the intended design goals.

Functional requirements may call for the creation of interactive prototypes. Selecting a tool with strong prototyping capabilities helps designers demonstrate the user experience and functionality more effectively during the design phase. If functional requirements demand collaborative design efforts or version control capabilities, you can evaluate tools based on their collaboration features, such as real-time collaboration, commenting, and version history.

By assessing how well the tool aligns with the functional requirements, you can gauge its usability and determine if it requires a manageable learning curve for your design team. Functional requirements might call for integrations with external services or design asset libraries. Evaluating whether the tool supports these integrations can impact the efficiency and productivity of the design process. Functional requirements might specify the need for specific export formats or delivery options for the design deliverables. Evaluating the tool's export capabilities ensures that it can deliver design assets in the required formats.

Functional requirements help identify essential features versus nice-to-have options. This helps in selecting a tool that fits within the project's budget without compromising critical design needs. By closely aligning functional requirements with the features and capabilities of different web designing tools, you can make a well-informed decision when choosing the most appropriate tool for your project. Careful evaluation ensures that the chosen tool empowers designers to meet the project's goals and deliver high-quality web designs efficiently.

3.1.2. User Requirements or User Stories

User requirements or user stories represent the needs and expectations of the end-users in a clear and concise manner. User stories are typically written from the perspective of the user and describe what the user wants to achieve when using the application [25]. These requirements function as a valuable communication tool between stakeholders, designers, and developers, ensuring that the application's design and functionality align with user needs.

User stories are specific, actionable, and focused on the user's perspective, and are free from technical jargon. These user requirements can be used as a basis in the selection process of web designing tool. By basing the tool selection process on user requirements, decision-makers can prioritize user satisfaction and ensure that the chosen web designing tool aligns with the project's objectives and delivers a successful and user-friendly web product.

Here's an example of a user story:

"As a designer, I want to be able to create interactive prototypes quickly and easily so that I can gather user feedback and validate design decisions efficiently."

This user story emphasizes the need for a web designing tool that supports interactive prototyping capabilities. The selection of a web design tool should consider features such as, the tool should enable designers to create interactive prototypes with clickable elements, transitions, and animations. The tool should facilitate usability testing and gathering user feedback on the interactive prototypes. The tool should be user-friendly and intuitive, allowing designers to create prototypes quickly without a steep learning curve. Collaboration features, such as real-time sharing and commenting, are essential for designers to share prototypes with stakeholders and receive feedback efficiently. The tool should offer export options that allow designers to share prototypes in various formats, making it easy to present the designs to different stakeholders.

By aligning the user story with the selection criteria, the web designing tool chosen can make the web design process effective, user-focused, and delivers a successful product.

3.1.3. Business Requirements

Business requirements are specific statements that outline the needs and objectives of a business or organization. They are essential for guiding projects, defining solutions, and aligning development efforts with the overall business strategy [26]. Business requirements provide a clear understanding of what a project or initiative aims to achieve and how it should be executed to meet the desired outcomes. Business requirements provide the context for the front-end design, guiding designers to create a web application that meets the business goals effectively. Business requirements can significantly influence the selection of a web design tool. The choice of the web design tool should align with the business objectives, project scope, and the specific needs identified in the business requirements.

Here's an example of a business requirement:

"Streamline Collaboration and Communication among Design Team Members"

According to this business requirement, the business seeks to improve the efficiency and effectiveness of the design process by enhancing collaboration and communication among members of the design team. The objective is to select a web design tool that facilitates real-time collaboration, feedback sharing, and version control to streamline the design workflow.

3.2. Resources

It is important to consider project resources before the selection of web design tool. Here are some examples of resources which can influence the selection of web design tool.

- Cost: The evaluation of project resources helps in selecting a web design tool that fits within the allocated budget. It prevents overspending on tools that may have unnecessary features or functionalities, allowing the project to utilize its resources more efficiently.
- Team expertise: Considering the skill level and expertise of the design team ensures that the selected tool is suitable for the team's capabilities. Choosing a tool that matches the team's proficiency leads to a smoother design process and faster adoption of the tool.
- Time Management: Selecting a web design tool that integrates well with existing project resources, such as version control systems or collaboration platforms, can save time and effort during the design and development phases.
- Scalability: Project resources help in assessing whether the selected web design tool can scale with the project's growth and handle future expansion and complexity. It ensures that the tool remains relevant as the project evolves.

3.3. Types of Web Design

It is important to understand what kind of web design will be required by the application because every web designing tool can create web design of different levels of visual fidelity. Having a clear concept of the type of web design required for web application will help designer to choose the appropriate web design tool. There are different types of web designs, some of them will be discussed in detail to show their influence on the selection process of web designing tool.

3.3.1. Wireframe

A wireframe is a blueprint or visual representation of an application. Wireframes provide an outline of the design's content, functionality, and user interface elements without getting into specific visual details like colors or typography [17]. They serve as a starting point for designers, developers, and stakeholders to understand the project's layout and functionality before moving on to more detailed stages.

Wireframes serve several key purposes in web design. They establish the basic layout and organization of the content on a web page or application, defining the placement of key elements such as headers, navigation, content areas, and footer. They outline the interactive elements and user interface components, showcasing how users will interact with the site or application. Wireframes facilitate communication among team members, clients, and stakeholders, as they provide a visual representation of the project's layout and features.

There are different types of wireframes, each offering varying levels of detail and complexity.

1. Low-Fidelity Wireframes are basic sketches that outline the layout and key elements without much detail. They are quick to create and ideal for early-stage brainstorming and concept validation.
2. Mid-Fidelity Wireframes add more detail and structure to the design, including basic text labels and sometimes grayscale representations of the content.
3. High-Fidelity Wireframes are more detailed and may include actual content, visual design elements, and interactions. They bridge the gap between wireframes and prototypes.

Multiple tools exist for generating wireframes, spanning from uncomplicated pen-and-paper sketches to digital software. Common wireframing tools include Adobe XD, Sketch, Figma, and Balsamiq. The decision to make wireframes can significantly impact the selection of a web design tool. The choice of tool will depend on various factors, such as the complexity of the project, collaboration needs, and the level of fidelity required for the wireframes.

If you need to create quick, low-fidelity wireframes to outline the basic layout and content structure, you might prefer using simple tools like pen and paper, whiteboard, or online wireframing tools that offer a straightforward interface for quickly sketching ideas. For wireframes that require a moderate level of detail and some visual representation, you might choose specialized wireframing software like Balsamiq. These tools offer pre-built UI elements and a library of components to speed up the wireframing process while maintaining a clean, unpolished look.

3.3.2. Mockup

A mockup is a detailed visual representation of a web page, application, or digital product. It is more refined and visually polished compared to a wireframe, incorporating actual design elements such as colors, typography, icons, images, and layout. Mockups are often static and lack interactive elements like those found in prototypes. They are crucial for presenting the visual aesthetics and overall design concept to stakeholders, clients, and team members during the web design process [8].

Mockups are generally more detailed than wireframes, but they still lack some interactive elements found in prototypes. They focus on the design and visual aesthetics rather than user interactions. Mockups can be created at different stages of the web design process, from the initial planning and wireframing stage to the final design stage [27]. Mockups can be presented in different formats based on the level of visual detail. Low-Fidelity Mockups are simple representations that may use placeholders for images and generic text for content. They are useful for early-stage design discussions and quick iterations. High-Fidelity Mockups are more detailed and visually polished. They use real images, actual content, and refined design elements to give stakeholders a more precise depiction of the final product.

The selection of tool will be highly dependent on the required fidelity level of mockup. Various design tools can be used to create mockups, ranging from graphic design software like Adobe Photoshop and Sketch to more specialized tools like Adobe XD, Figma, and InVision.

3.3.3. Prototype

A prototype is an interactive, functional representation of a web page, application, or digital product that simulates the user experience and demonstrates how the final product will work [17]. The primary goal of a prototype is to test and validate the functionality, user interactions, and overall usability of a digital product before investing resources in the development phase. Prototypes help identify potential issues and improvements early in the design process, leading to a more efficient and user-friendly final product. Unlike wireframes and mockups, prototypes include interactive elements that simulate user interactions and experiences. These interactions can be as simple as button clicks, dropdown menus, or scrolling, or as complex as navigating through different pages or completing form submissions.

The fidelity of prototypes can vary, depending on the design objectives and the stage of the project.

1. Low-Fidelity Prototypes are basic and quick prototypes with minimal visual design elements and limited interactivity. They are useful for early-stage concept testing and refining the user flow.
2. Medium-Fidelity Prototypes have more visual design elements and better representation of user interactions. They are useful for usability testing and gathering feedback on the user experience.
3. High-Fidelity Prototypes closely mirror the final product concerning visuals and interactions They are ideal for conducting detailed user testing and presenting a near-realistic experience to stakeholders.

There are different types of prototypes based on the level of interactivity and complexity. For example, Clickable Prototypes allow users to click through the design and interact with various elements [17]. They are often created using prototyping tools like Adobe XD, Sketch, Figma, or InVision. So, before selecting a tool it is important to know about the level of fidelity and type of web design.

The following table, Table 3.1 can serve as a guideline in deciding the type of web design according to specific needs and requirements of project.

Table 3.1 *Classification of types of web design*

Properties	**Wireframe**	**Mockup**	**Prototype**
Purpose	blueprint	visual representation	interactive and functional representation
Interactivity	non-interactive	non-interactive	interactive
User Testing	not suitable	not suitable	suitable
Usability Validation	validate layout	validate visual design	validate both visual design and user interactions
Development Handoff	provide basic layout structure for development	provide visual design guide	provide developers with a clear understanding of the design's interactive elements and user flow

3.4. Web Designing Tools

The diversity of web design tools exists due to the varying needs and preferences of web designers and developers. Different types of web design tools serve specific purposes and cater to different stages of the web design process. Ultimately, the existence of various web design tools reflects the dynamic and multifaceted nature of web design itself. As the web design landscape evolves and technology advances, new tools with innovative features and capabilities are likely to continue emerging to meet the evolving needs of web designers and developers. Before selecting a web designing tool, it is important to understand the functionality and features of that tool. Some of the most popular web design tools are Sketch, AdobeXD, Figma, InVision, and Balsamiq [28]. These tools are discussed in detail below.

3.4.1. AdobeXD

Adobe XD [29], short for Experience Design, stands out as a robust and versatile design and prototyping tool crafted by Adobe Inc. It is purpose-built for the creation of user interfaces (UI) and user experiences (UX) for both web and mobile applications. Introduced in 2016, Adobe XD quickly gained popularity among designers and design teams, due to its user-friendly interface, streamlined workflow, and seamless integration with other Adobe Creative Cloud applications. Some key features of Adobe XD are discussed below which can help in the selection process of web designing tool.

1. *Responsive Design:* With Adobe XD, designers have the capability to develop responsive designs that seamlessly adapt to diverse screen sizes and orientations. You can create multiple artboards for various devices and use responsive resizing to ensure elements scale proportionally.
2. *Design Components and Libraries:* Adobe XD supports design components that allow you to create reusable UI elements and maintain design consistency across projects. Libraries can be shared across files and updated globally, ensuring that changes are propagated everywhere.
3. *Interactive Prototyping:* Adobe XD excels in prototyping capabilities. Designers can link artboards with interactive hotspots and define transitions to simulate user flows and interactions. This feature is crucial for testing and refining the user experience before development.
4. *Auto-Animate:* Adobe XD's Auto-Animate feature enables designers to create advanced animations between artboards automatically. This allows for more engaging and dynamic prototypes without the need for complex animation tools.
5. *Voice Prototyping:* Adobe XD includes voice prototyping, allowing designers to create voice-enabled interactions and test voice commands within their prototypes.

6. *Real-Time Collaboration:* Adobe XD allows real-time collaboration on cloud documents, enabling multiple designers to work simultaneously and see changes in real-time. This fosters efficient teamwork and communication.
7. *Design Specs and Handoff:* Designers can generate design specs that developers can access to inspect design details, measurements, and CSS properties. This simplifies the handoff process and ensures accurate implementation of the designs.
8. *Plugins and Integrations:* Adobe XD supports an extensive range of plugins created by both Adobe and third-party developers. These plugins enhance the functionality of the tool and extend its capabilities further.
9. *Adobe Creative Cloud Integration:* Being part of the Adobe Creative Cloud ecosystem, XD seamlessly integrates with other Adobe tools like Photoshop, Illustrator, and After Effects, making it easy to transfer assets and designs between applications.

3.4.2. Figma

Figma [30], launched in 2016, is a cloud-based design and prototyping tool that has gained substantial popularity within the web and app design community. It enables designers to create, collaborate, and share user interfaces and interactive prototypes with ease. Figma is known for its versatility, real-time collaboration features, and accessibility across different platforms, making it a preferred choice for designers and design teams worldwide. Some of key features offered by Figma are summarized below, which can be helpful while selecting design tool:

1. *Cloud-Based Design:* Figma operates entirely in the cloud, meaning there's no need to install any software on your device. You can access Figma through a web browser or via desktop apps for Windows and macOS.
2. *Real-Time Collaboration:* An exceptional feature of Figma is its support for real-time collaboration. This allows multiple users to collaborate on the same design project simultaneously, with the ability to observe each other's changes in real-time. This feature greatly facilitates teamwork, enabling designers, developers, and stakeholders to collaborate seamlessly.
3. *Design Components and Libraries:* Figma allows users to create reusable design components, such as buttons, icons, and navigation bars. Designers can organize these elements into libraries, which can be shared and accessed across different projects, ensuring design consistency.
4. *Prototyping and Interactions:* Figma's built-in prototyping tools enable designers to create interactive prototypes and wireframes. You can define clickable links and transitions between design frames to simulate user interactions and test the user experience.

5. *Responsive Design:* Figma offers responsive design features, allowing designers to create adaptive designs that work well on various devices and screen sizes. Designers can create multiple layouts within the same file and use constraints to maintain consistency across different screen sizes.
6. *Version History and Comments:* Figma keeps track of design changes with version history, allowing users to revert to previous versions if needed. It also enables users to leave comments directly on specific design elements, facilitating effective feedback and communication among team members.
7. *Design Handoff:* Figma streamlines the handoff process between designers and developers. Design specs and assets can be easily extracted, making it easier for developers to implement the designs accurately.
8. *Plugins and Integrations:* Figma supports a wide range of plugins created by both Figma and third-party developers. These plugins extend the tool's functionality and can be used to automate repetitive tasks or add additional design features.
9. *Security and Permissions:* Figma prioritizes security, and it offers various permissions and access controls to manage team collaboration securely.
10. *Design Versioning:* Figma allows users to create design variants and compare different versions side by side, enabling easy A/B testing and iteration.

3.4.3. Balsamiq

Balsamiq [31] is a wireframing and prototyping tool used for creating quick, low-fidelity mockups and sketches of user interfaces (UI) and user experiences (UX). It was first released in 2008 and has since become a popular choice among designers, product managers, and development teams for its simplicity and focus on rapid prototyping. Here is the list of some features offered by Balsamiq:

1. *Low-Fidelity Sketches:* Balsamiq's main strength lies in its low-fidelity approach to wireframing. It provides a library of pre-built UI elements (known as "controls") that resemble hand-drawn sketches, giving the mockups a rough, unfinished appearance. This encourages stakeholders to focus on the layout and structure rather than getting caught up in visual details.
2. *Simple Interface:* Balsamiq's user interface is intentionally minimalistic and straightforward. The goal is to make it easy for users to quickly create wireframes without the need to spend time on learning complex design tools.
3. *Extensive Library:* Balsamiq comes with a vast collection of UI components and widgets commonly used in web and app design, such as buttons, text boxes, navigation bars, and more. Users can easily drag and drop these elements onto the canvas to create mockups rapidly.

4. *Custom Components and Templates:* In addition to the built-in library, Balsamiq allows users to create custom UI components and templates, enabling design consistency across different projects.
5. *Quick Iterations:* The simplicity of Balsamiq allows for rapid iteration and exploration of design ideas. It's particularly useful during brainstorming sessions and early design stages when multiple concepts need to be generated quickly.
6. *Collaboration and Feedback:* Balsamiq provides collaborative features, allowing multiple team members to collaborate on the same project. It also permits stakeholders to leave comments and feedback directly on the wireframes, streamlining the feedback process.
7. *Integration with Other Tools:* While Balsamiq is primarily used for low-fidelity wireframes, it can be integrated with other prototyping tools and design applications for more advanced interactions and high-fidelity mockups.
8. *Export Options:* Balsamiq allows users to export wireframes in various formats, such as PNG, PDF, and interactive PDF, making it easy to share and present designs with stakeholders and clients.
9. *Offline and Cloud Versions:* Balsamiq offers both desktop (offline) and cloud-based versions, giving users the flexibility to work on their projects regardless of their internet connectivity.

3.4.4. Sketch

Sketch [32] is a widely used vector-based design tool developed by Bohemian Coding, primarily designed for crafting user interfaces (UI) and user experiences (UX) for web and mobile applications. It was first released in 2010 and quickly gained traction in the design community, particularly among UI/UX designers, due to its focus on simplicity, flexibility, and efficiency. Some of key features offered by Sketch are:

1. *Vector-Based Design:* Sketch is built on a vector-based platform, allowing designers to create scalable and resolution-independent designs. This makes it suitable for designing assets that can be used across different screen sizes and resolutions.
2. *Artboards and Pages:* Sketch uses an artboard-based approach, where designers work on individual screens or pages. This structure makes it easy to manage and organize multiple screens within a single file, enabling efficient design iterations.
3. *Symbols and Shared Styles:* Sketch introduced the concept of Symbols, which are reusable design elements like buttons, icons, and navigation bars. By using

Symbols and Shared Styles, designers can maintain design consistency throughout the entire project and make changes globally.

4. *Plugins and Extensions:* Sketch has a robust plugin ecosystem that allows users to extend its functionality and integrate with other tools and services. Designers can use plugins to automate repetitive tasks, export assets, and access additional design features.
5. *Responsive Design:* Sketch provides features for creating responsive designs, including resizing constraints, which allow designers to specify how elements should adapt when their parent container changes size.
6. *Prototyping and Interaction:* While Sketch is primarily a design tool, it offers basic prototyping capabilities with the ability to define clickable hotspots and transitions between artboards. However, for more advanced prototyping, designers often use other dedicated prototyping tools.
7. *Exporting Assets:* Sketch allows designers to export assets in various formats, including PNG, JPG, SVG, and more. Designers can also specify different export scales and resolutions to accommodate various device requirements.
8. *Integration with Design Systems:* Sketch is commonly used in conjunction with design systems, where designers create and maintain a centralized library of design elements that can be shared and reused across different projects.
9. *Collaboration and Version Control:* Sketch supports collaborative workflows through plugins or cloud-based solutions, enabling team members to work together on the same files. Additionally, some plugins provide version control features to track changes and revert to previous states.
10. *Accessibility Features:* Sketch includes features to check the accessibility of designs, ensuring that they meet certain standards for users with disabilities.

3.4.5. InVision

InVision [33] is a digital product design and prototyping platform that facilitates collaboration among designers, developers, and stakeholders. It was founded in 2011 and has since become one of the leading tools in the UX/UI design industry. InVision provides a diverse set of features catering to various stages of the design process, spanning from wireframing and design to interactive prototyping and design collaboration. Some of key features are discussed below:

1. *Prototyping:* InVision is known for its powerful prototyping capabilities. Designers can create interactive and animated prototypes by linking artboards and defining user interactions and transitions. This allows stakeholders and users to experience the design in an interactive way, making it easier to gather feedback and validate design decisions.

2. *Design Collaboration:* InVision facilitates seamless collaboration between designers, developers, and stakeholders. Team members can leave comments directly on design elements, making it easier to share feedback and iterate on designs.
3. *Version History and Design Handoff:* InVision keeps track of design versions, allowing designers to revert to previous states if needed. It also provides features for design handoff, allowing developers to access design specs, measurements, and assets needed for implementation.
4. *Design System Management:* InVision provides tools for creating and maintaining design systems. Designers can create libraries of reusable components and styles, ensuring design consistency across different projects.
5. *User Testing and Feedback:* InVision allows designers to conduct user testing and gather feedback through features like clickable prototypes and user testing sessions. This helps validate design decisions and improve the user experience.
6. *Mobile Preview:* InVision offers a mobile app that allows users to preview and test prototypes directly on their devices.
7. *Craft Plugin*: InVision's Craft plugin integrates with design tools like Sketch and Adobe XD, allowing designers to sync their designs to InVision and create prototypes directly from within their design tools.
8. *Animation and Micro interactions:* InVision supports more advanced animations and micro interactions through features like "Auto-Animate," which enables designers to create more dynamic and realistic prototypes.
9. *Security and Permissions:* InVision prioritizes security and provides granular permissions settings to control access and editing rights for team members.

The strengths and weaknesses of Adobe XD, Figma, Sketch, Balsamiq, and InVision are summarized in Table 3.2 below. This information can be very helpful in the selection process of web designing tool.

Table 3.2 *Summary of Strengths and Weaknesses of Web Designing Tools*

Tools	**Strengths**	**Weaknesses**
Adobe XD	Seamless integration with other Adobe tools, powerful prototyping features, design consistency through	Some advanced features might be limited compared to specialized prototyping tools

	Components, and shared assets via Creative Cloud	
Figma	Real-time collaboration, cross-platform accessibility, powerful design system management, and versatile prototyping capabilities	Occasional performance issues with large files, offline functionality limited compared to desktop apps
Sketch	Lightweight and fast, extensive plugin ecosystem, strong vector editing capabilities, industry-standard for macOS users	Limited cross-platform accessibility (macOS only), some prototyping features require third-party tools
Balsamiq	Speedy wireframing, minimal learning curve	Limited to low-fidelity wireframes
InVision	Advanced prototyping capabilities, user testing features, strong design collaboration tools.	Limited vector-based design capabilities

3.5. Key Strategies for Efficient Web Design Tool Utilization

When the web designing tools for web design of a web front-end application have been selected, it is important to know how to use the features offered by tool efficiently for web designing. This can significantly boost your productivity and streamline your design workflow.

It is recommended to get familiarize with the tool’s interface, menus, and shortcuts. Familiarity with the layout will save time and reduce errors in the long run. Many web design tools offer templates and pre-built design libraries. Utilize these resources to speed up your design process and maintain consistency across projects. Learn and use keyboard shortcuts for commonly used actions. This will help you navigate the tool faster and perform tasks more efficiently. Organize your design files with a clear folder structure and naming conventions. This will make it easier to find and manage assets and designs as your project grows. If your design tool supports it, create reusable design components for elements like buttons, icons, and navigation bars. This way, you can update these components globally, ensuring design consistency. Regularly check for updates and new features in your web design tool. Staying informed will help you take advantage of the latest improvements and functionalities. Keep your design files clean and organized by removing unnecessary elements and optimizing assets. This will help maintain file performance and prevent clutter. Web design tools often have hidden or advanced features that you may not be aware of, continuously seek learning resources, tutorials, and online courses to expand your skills and efficiency.

By implementing these tips, you can enhance your proficiency with the web design tool and ensure a smooth and productive design process. Efficient use of the tool will enable you to focus more on creativity and delivering high-quality designs.

3.6. Conclusion

By considering the factors mentioned in the beginning of this section 3, users can make well-informed decisions when choosing web design tools to streamline their web design process. The first step is to understand the specific requirements of the web application. Users should identify the desired features, functionality, and target audience to select a tool that best aligns with the project's needs. After that it is important to assess the resources available for web designing, including budget, team expertise, and time constraints. Choosing a tool that fits within these resource limitations ensures a smoother design process. Different web applications may require different types of web design, so it is crucial to select the tool according to the type of web design. There are numerous web design tools with varying features and capabilities. Users should research and compare different tools to find the one that offers the functionalities needed for their specific project. Once the web design tool is selected, users can maximize its potential by mastering the tool's interface, learning keyboard shortcuts, and leveraging features like templates, design components, and collaboration options. Continuous learning and staying updated on new features can further improve efficiency.

4. Selection of JavaScript Frameworks

JavaScript framework (JSF) selection is a challenging task for developers due to several reasons. There is an abundance of JSF available, each with its own set of features, approaches, and philosophies. This wide array of choices can be overwhelming for developers, making it difficult to determine which framework is the best fit for their project.

The JavaScript ecosystem is dynamic and constantly evolving. New frameworks and libraries emerge regularly, while existing ones receive updates and enhancements. Keeping up with the latest trends and evaluating the suitability of each framework requires continuous learning and research.

JSF selection involves a long-term commitment, as migrating to a different framework can be time-consuming and disruptive. Given these challenges, it is important for developers to thoroughly research and evaluate before making a final decision. Being able to answer the following questions provides a guide in the selection process.

- What type of requirements to consider before selecting a framework?
- What should be the architecture of web front-end application?
- What should be the type of web front-end application?
- What options of JavaScript front-end frameworks or libraries are available and what are their key features?
- How to do the analysis of selected JavaScript frameworks?

Below are justifications for the importance of considering these questions before selecting a JavaScript framework for a web front-end application.

4.1. Requirements

In the context of selecting a JavaScript framework for web front-end development, there are several types of requirements to consider. These requirements help ensure that the chosen framework is suitable for your project and aligns with your development goals. Here are some specific types of requirements for JavaScript framework selection:

1. *Functional Requirements*: These requirements pertain to the specific features and functionalities needed for web application. For example, UI components, data handling, state management, and any other capabilities required to build application. If a web application has a functional requirement to present data in various visual formats, such as charts, graphs, maps, and interactive data tables. Then selection of a JavaScript framework or library which has built-in data visualization components can significantly streamline the development process.
2. *Non-functional Requirements*: Non-functional requirements are quality attributes that describe how the system should behave or perform rather than what it should do. Examples include performance, reliability, security, scalability,

usability, and accessibility requirements. If it is the requirement to build a real-time collaboration tool or an interactive application that requires frequent updates and real-time data synchronization, a JavaScript framework that minimizes rendering and processing delays would be crucial. It should have optimized algorithms and rendering mechanisms to handle dynamic content efficiently. Keep in mind that different JavaScript frameworks have varying performance characteristics, and the best choice will depend on your specific application's requirements.

3. *User Requirements*: User requirements are declarations that encapsulate the needs and expectations of end-users or stakeholders. These statements emphasize what users anticipate the system to accomplish and how it should behave from their perspective. For example, if cross-browser compatibility is a user requirement it can significantly influence the selection of a JavaScript framework. In this requirement, user is expecting the web application to function and appear consistently across different web browsers and devices. Each browser may have slight differences in rendering and handling JavaScript code, leading to potential inconsistencies in the user experience. Therefore, selecting a JavaScript framework that ensures smooth operation and consistent behavior across major browsers is essential to meet user expectations.
4. *Business Requirements*: Business requirements focus on the overall business goals and objectives that the software system should support. They help ensure that the software aligns with the organization's strategic direction. If it is a business requirement for an application to have rapid development and quick time-to-market, then this requirement will have an impact on the selection of framework. Some frameworks offer features, tools, and abstractions that enable developers to build complex applications with less code and in a shorter timeframe. Rapid development can lead to quicker iteration cycles, faster feedback from users, and earlier deployment of valuable features.

4.2. Architecture of Web Front-end Application

The choice of architecture can significantly impact the selection of a JavaScript framework for your web front-end application. Different architectures have varying requirements and best practices, and certain JavaScript frameworks are better suited for specific architectures. Architecture of an application can provide valuable insights to make an informed decision.

For example, if someone is working with an architecture like Model-View-Controller (MVC) [34] or Model-View-ViewModel (MVVM) [34], you might want a framework that enforces a clear separation between the Model, View, and Controller/ViewModel. If someone is adopting a unidirectional data flow pattern like Flux or Redux, which helps manage complex state in large applications, you might consider React with Redux or frameworks that provide built-in support for Flux architecture.

Architecture can provide some types of key information to help in the selection of a JavaScript framework. For example, architecture can help in evaluating the complexity of your application's user interface and the interactions required. How your application's components will be structured and whether component reusability is important for your application. It can also help in analyzing the data management requirements of your application. Does it need advanced state management, such as Redux or Flux patterns? Some frameworks come with built-in solutions for state management, while others rely on external libraries.

4.3. Type of Web Front-end Application

The choice of the type of web front-end application can have a significant impact on the selection of the JavaScript framework for development. Different front-end application types have distinct requirements and characteristics, and certain JavaScript frameworks are better suited to address those needs. Here are some ways the type of front-end application can influence the selection of the JavaScript framework. SPAs require a robust framework to handle complex user interactions and UI updates without page reloads. Frameworks like React, Angular, and Vue.js are well-suited for SPAs due to their component-based architecture and virtual DOM handling, which improves performance and makes it easier to manage complex UIs. For simpler static web applications or multi-page applications that don't require extensive UI interactions, developers might opt for lightweight or minimal JavaScript frameworks or libraries. Vanilla JavaScript or smaller libraries like jQuery might be sufficient for these cases. Progressive Web Applications (PWAs) benefit from a combination of a front-end framework and service workers for caching and offline capabilities. React and Angular have strong PWA support, making them popular choices for building progressive web apps. Responsive web applications need a framework that supports responsive design, ensuring the UI adapts smoothly to various screen sizes and devices. Most modern JavaScript frameworks provide tools and techniques for responsive web development.

4.4. JavaScript Framework and Libraries

There is a wide range of JavaScript frameworks and libraries. In the selection process of these frameworks, it is important to know what options of frameworks and libraries we have and what functionalities they are offering. This understanding can help developer in decision making. Some of these frameworks and libraries are introduced below.

4.4.1. React

React [35] is commonly employed for building user interfaces in web applications. Developed by Facebook, it stands as a popular JavaScript UI framework. Operating on a component-based architecture, React divides the UI into reusable and self-contained components. These components possess their properties and state, facilitating the management and updating of the UI in response to changes in data.

React employs JSX (JavaScript XML), which serves as a syntax extension of JavaScript. This extension enables developers to write HTML-like code directly within their JavaScript files, simplifying the creation and manipulation of UI components [23]. React promotes the development of reusable and modular components, where each component handles its own state and properties. These components can be composed together to construct intricate UI hierarchies.

In React, the virtual DOM compares the current virtual representation of the UI with the previous one, identifying differences and efficiently updating only the necessary parts of the actual DOM. This approach minimizes the number of actual DOM manipulations, resulting in improved performance. When a React component is initially rendered, it generates a virtual DOM representation of the component's UI. This virtual DOM serves as a lightweight copy of the actual DOM tree, enabling React to calculate differences (diffing) between the previous and current states. When the state or props of a React component change, a new virtual DOM representation is created for the updated component.

React then initiates a process called "diffing" or "reconciliation," involving a comparison between the previous virtual DOM and the updated one. React identifies the differences, determining the necessary changes to be made to the actual DOM. Once these differences are calculated, React selectively updates only the specific parts of the actual DOM that have changed. This targeted update is significantly more efficient than updating the entire DOM tree. The virtual DOM representation in React is illustrated in Figure 4.1.

In React, the data flow follows a unidirectional path, with data passed through properties (props) from parent to child components. This design choice enhances the clarity and ease of debugging in understanding the flow of data within the application. With React version 16.8, the introduction of hooks enables developers to incorporate state and lifecycle functionalities into functional components. Hooks provide a simpler and more concise approach to managing component state and handling side effects [36].

React has an extensive ecosystem comprising libraries, tools, and robust community support. Developers can leverage various third-party libraries and tools to enhance React applications, like React Router for routing [26], Redux for state management [37], and Material-UI [38] for ready-to-use UI components.

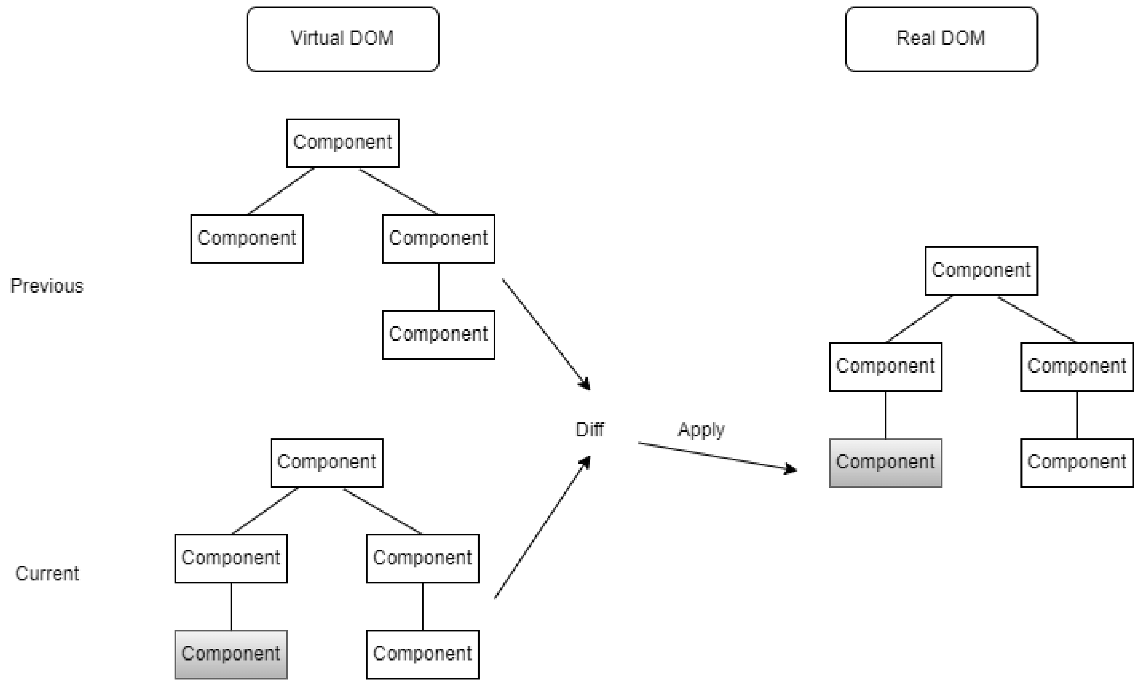


***Figure 4.1** Virtual DOM in React [45, Fig. 2.3]*

4.4.2. Angular

Angular [39] is a widely used JavaScript UI framework developed by Google. It is a comprehensive framework suitable for the complete development of web applications. Angular follows a component-based architecture and offers a variety of tools and features to simplify the building process. It includes features like forms handling, dependency injection, routing, data binding, and testing utilities out of the box. This comprehensive approach reduces the need to rely on external libraries or tools for many common development tasks.

Like React, Angular adheres to a component-based architecture. The UI is segmented into reusable and modular components, each encapsulating its own logic, styling, and templates. Angular's components facilitate two-way data binding [40], simplifying the management and updating of the UI based on changes in the underlying data. Two-way data binding ensures synchronization between the data model and the view. Any alteration in the data model is reflected in the view, and vice versa. This process is immediate and automatic, ensuring both the model and view stay updated consistently.

Angular employs TypeScript, a strongly typed superset of JavaScript developed by Microsoft. TypeScript introduces optional static typing to JavaScript, enabling developers to identify errors and bugs during development rather than at runtime. TypeScript code is transpiled to standard JavaScript, making it executable in any browser or Node.js environment. TypeScript brings features like static typing, enhanced tooling, and better code organization to the development process. It provides improved maintainability, refactoring capabilities, and better developer experience.

Angular offers a rich set of directives that extend HTML with additional functionality. Directives allow developers to add custom behaviors, manipulate the DOM, handle events, and create reusable components. Angular offers a robust dependency injection system, providing an efficient means to manage dependencies and encourage modularity. This feature facilitates the creation of more testable and maintainable code by decoupling components and services.

Additionally, Angular harnesses the capabilities of Reactive Extensions for JavaScript (RxJS) to handle asynchronous operations and support event-based programming. RxJS enables the effective management of data streams, simplifying the handling of asynchronous tasks like HTTP requests and real-time updates.

Furthermore, Angular is equipped with the Angular Command Line Interface (CLI), a potent tool automating common development tasks. This includes activities such as scaffolding components, code generation, test execution, and application building for production.

4.4.3. Vue

Vue.js [41] stands out as a popular open-source JavaScript framework primarily utilized for constructing user interfaces and single-page applications (SPAs). Recognized for its flexibility, accessibility, and incremental adoptability, Vue is suitable for both small and large-scale projects. It often receives praise for its simplicity, ease of learning, and seamless integration with existing projects.

Vue employs a reactive data binding system, ensuring that alterations in the underlying data automatically update the corresponding parts of the UI. This facilitates efficient and intuitive handling of data-driven UI updates [42]. Like React and Angular, Vue adheres to a component-based architecture, allowing Vue components to be combined to construct intricate UI hierarchies.

Vue supports Single File Components (SFC). It is a file format and component organization approach used to encapsulate the style, template, and script of a Vue.js component within a single file. Traditionally, in front-end development, the template, script (JavaScript), and styles (CSS) of a component were often separated into three different files. However, as the complexity of applications grew, managing these separate files became cumbersome. SFC were introduced to address this issue by consolidating all the related code for a component into a single, self-contained file with a '.vue' extension. This helps in organizing and maintaining components by keeping related code in one place.

Vue provides an official routing library called Vue Router. It helps in building SPA with multiple views and seamless navigation between them. In Vue, the official state management library is Pinia. It provides a centralized store to manage application state, enabling efficient data sharing and communication between components.

4.4.4. Strengths and Weaknesses of JS frameworks

The strengths and weaknesses of few popular JavaScript frameworks are summarized below to provide a structured and concise information which can facilitate in comparison, comprehension, and decision-making.

Strengths:

React:

- React's Virtual DOM optimally updates only the necessary parts of the actual DOM, resulting in enhanced rendering performance and a smoother user experience.
- React's component-based approach promotes reusability, modularity, and maintainability of code.
- React has a massive and active community, which means abundant resources, libraries, and third-party packages are available for developers.
- React is primarily a view library, allowing developers to choose additional libraries and tools for state management, routing, etc.
- React's architecture and concepts can be easily extended to build mobile applications using React Native.

Angular:

- Angular is a comprehensive framework that offers built-in solutions for many aspects of development, including state management, forms, HTTP requests, etc.
- Angular is built with TypeScript, which provides help catch errors during development.
- Angular's dependency injection system allows for better modularity and testability of code.
- Angular's two-way data binding simplifies data synchronization between the UI and the underlying model.

Vue:

- Vue is praised for its easy learning curve, making it accessible for developers with basic HTML, CSS, and JavaScript knowledge.
- Vue provides a balance between React's flexibility and Angular's opinionated approach, making it versatile and adaptable to various projects.
- Vue's reactivity system and efficient rendering mechanisms contribute to excellent performance.
- Vue's smaller bundle size and faster initial rendering make it a suitable choice for small to medium-sized projects.

Weaknesses:

React:

- React can have a steeper learning curve for beginners, especially when dealing with advanced concepts like state management.
- React might require additional setup and boilerplate code for features that are built-in or easier to implement in other frameworks.

Angular:

- Angular's extensive feature set and opinionated architecture can lead to a steeper learning curve and increased complexity for small to medium-sized projects.
- Angular's aggressive change detection strategy can result in slightly higher performance overhead compared to other frameworks.
- Angular's core is larger compared to other frameworks, leading to larger initial bundle sizes.

Vue:

- Vue's ecosystem, while growing, might not be as extensive as React's or Angular's, leading to fewer choices for third-party packages and libraries.
- While Vue is gaining popularity, it might not be as widely adopted in large enterprise-level applications compared to Angular.

4.5. Analysis of selected framework

To perform the analysis of selected framework, it is important to define an evaluation criterion. The answers to the questions mentioned in start of this section 4 will help in formulating an evaluation criterion according to the needs of project. This evaluation criteria for the JavaScript framework will depend on the project requirements, architecture, and type of application. The key features of the selected framework should satisfy the evaluation criterion so that the developed application will meet all types of project requirements.

In summary, choosing the right framework for development positively impacts productivity, code quality, performance, and scalability. It empowers developers to build robust applications efficiently and maintain them effectively over time. A well-chosen framework sets the foundation for a successful development process and contributes to the overall success of the project.

5. Implementation

To study and analyze the usefulness of proposed approaches for the selection of web designing tool and JavaScript framework, this approach was used for the implementation of a software application named Matchmaking GUI application.

5.1. Matchmaking GUI Application

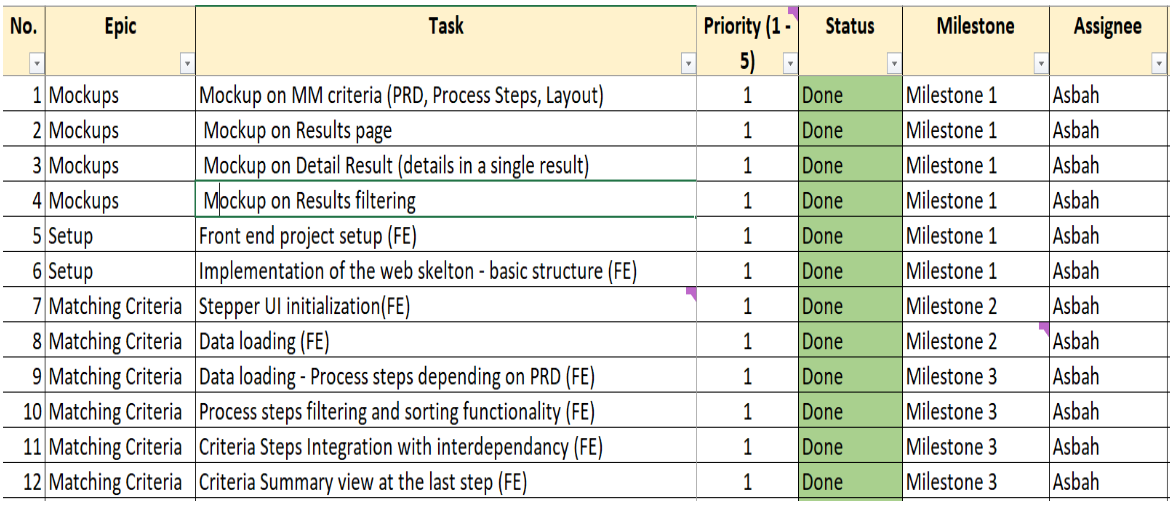
In manufacturing industry, traditionally system designer performs manually reconfiguration planning and system design. This manual process depends on the system designer's expertise and knowledge. They must find the appropriate configuration solutions by comparing and finding the suitable production/manufacturing resources and resource combinations for product requirements. There was a need to automatize this manual process to avoid human-caused errors and for efficient results. The capability matchmaking software was developed to resolve this issue [43]. This matchmaking service was implemented using RESTful architecture. For capability matchmaking service, a user interface was needed to allow the system designer to interact with capability matchmaking software running on a remote server.

There was a need to develop a web application, so to overcome this issue, Matchmaking Graphical User Interface (MM GUI) application is designed and developed which allows the user to interact with matchmaking service from the browser of its personal computer. The purpose of this MM GUI is to allow the user to focus on system design and reconfiguration planning without getting worried about software installations and updates. This MM GUI makes it easier for the system designer to access matchmaking service from browser without any local setup and software installations. The system designers or users will not get bothered by any update on remote server and they can interact or access matchmaking software from any combination of hardware and operating system via browser. This MM GUI assures that system designer can utilize the services in diverse circumstances.

5.1.1. Requirements Gathering

The first stage to develop MM GUI was requirements gathering phase. It starts with kick-off meetings and workshops. In these meetings, brainstorming sessions are performed and different ideas and perspectives are collected within a team. In these meetings all types of requirements were gathered including functional and non-functional requirements.

After identification of these requirements, these requirements were categorized into main features and functionalities which need to be implemented in MM GUI. After taking notes of these identified functionalities and features, they are divided into smaller tasks and these tasks are prioritized and listed in an excel sheet. The excel sheet contains different columns including tasks, time estimation (how much time in days a particular task required), priority (1 highest priority – 5 lowest priority), status (not started, in progress, done), milestone, and assignee. This sheet helped in keeping track of the progress of MM GUI. Few tasks from excel sheet are shown in Figure 5.1 below:



No.	Epic	Task	Priority (1 - 5)	Status	Milestone	Assignee
1	Mockups	Mockup on MM criteria (PRD, Process Steps, Layout)	1	Done	Milestone 1	Asbah
2	Mockups	Mockup on Results page	1	Done	Milestone 1	Asbah
3	Mockups	Mockup on Detail Result (details in a single result)	1	Done	Milestone 1	Asbah
4	Mockups	Mockup on Results filtering	1	Done	Milestone 1	Asbah
5	Setup	Front end project setup (FE)	1	Done	Milestone 1	Asbah
6	Setup	Implementation of the web skelton - basic structure (FE)	1	Done	Milestone 1	Asbah
7	Matching Criteria	Stepper UI initialization(FE)	1	Done	Milestone 2	Asbah
8	Matching Criteria	Data loading (FE)	1	Done	Milestone 2	Asbah
9	Matching Criteria	Data loading - Process steps depending on PRD (FE)	1	Done	Milestone 3	Asbah
10	Matching Criteria	Process steps filtering and sorting functionality (FE)	1	Done	Milestone 3	Asbah
11	Matching Criteria	Criteria Steps Integration with interdependancy (FE)	1	Done	Milestone 3	Asbah
12	Matching Criteria	Criteria Summary view at the last step (FE)	1	Done	Milestone 3	Asbah

Figure 5.1 *Division of tasks for MM GUI application*

There were different features in MM GUI application. But in this section only one main feature is highlighted which is making a new match request. The feature of making a new match request includes:

- selection of Product details
- selection of Process steps based on the product detail
- selection of Resource pools and Layout
- sending of new Match request to server

5.1.2. Design Stage

For the design stage of MM GUI application, the selection approach proposed in section 3 has been adopted.

Requirements:

According to requirements gathering sessions mentioned above in section 5.1.1. A few requirements gathered which were relevant to web design stage are mentioned below:

- The tool's user interface should be clean, modern, and easy to navigate. It should provide clear instructions and visual cues for various design actions.
- The tool should provide a library of design elements, such as images, icons, buttons, and graphics, for designers to use in their project.
- The tool should offer a selection of pre-designed templates and themes.
- Users should be able to upload their custom assets and reuse them across different projects.

- Designers should have the ability to customize various design aspects, including colors, fonts, layouts, and styles. The tool should support custom CSS styles for advanced customization.
- The tool should allow designers to export the design project into various file formats
- The tool should support collaboration among designers, clients, and team members. Users should be able to share design previews and receive feedback directly within the tool.
- There should be option of restrict access to protect user work.

Resources:

For MM GUI application, two most important resources to consider for selection of web designing tool were budget of project and time. It was necessary to select a tool which would offer a free plan and it was also important to adopt a tool which would be easy to learn. Another option was to select a tool which is familiar to the team. An already familiar tool will significantly help in productivity, efficiency, and collaboration.

Types of web design:

Two types of web designs were created for MM GUI to facilitate development stage. So, the first method selected was of wireframes and the second method was low-fidelity prototype, details of these web designs has already been discussed in section 3.

For wireframing of MM GUI application, paper wireframes were sketched. For paper wireframes, pen and paper was used to draw sketches of MM GUI application. These wireframes helped in understanding the logical flow and layout of application. These sketches were the root of interface design. These paper wireframes saved time, energy, and money and it was easy to share them after scanning and copies can also be printed out if needed.

The initial wireframes of product selection, process steps selection, resource pools and layout selection are shown in Figures 5.2, 5.3, and 5.4 respectively. These wireframes go through an iterative process of feedback and improvements. After wireframing stage, it was decided to create low-fidelity prototype to get a clear picture of visual design and it was also required to create limited user interactions.

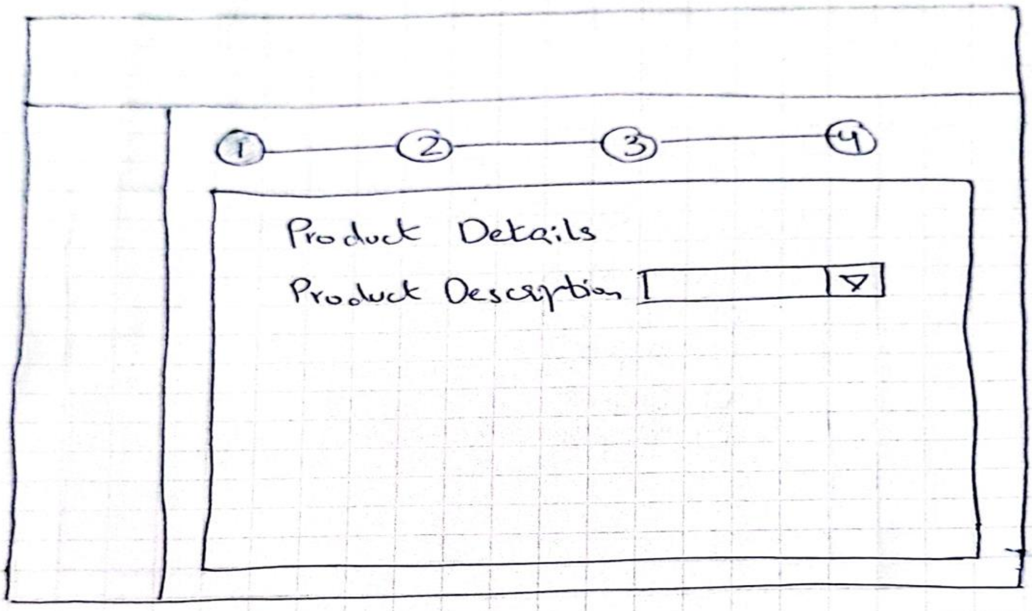


Figure 5.2 *Wireframe for Product selection in MM GUI application*

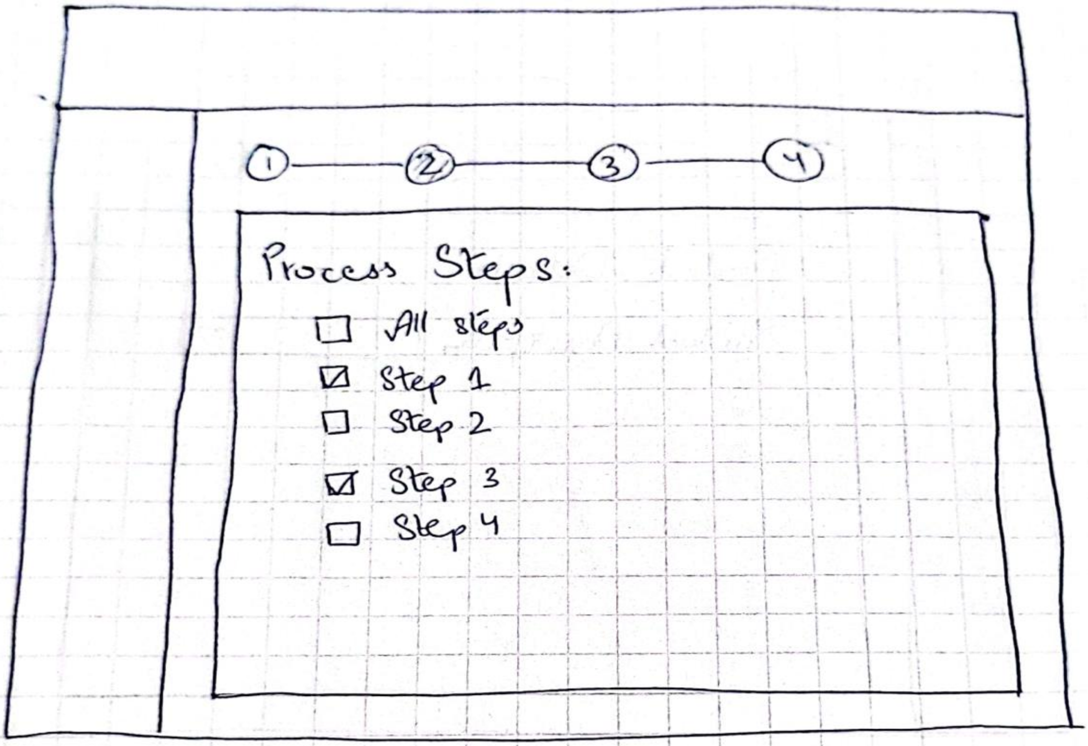


Figure 5.3 *Wireframe for Process steps selection in MM GUI application*

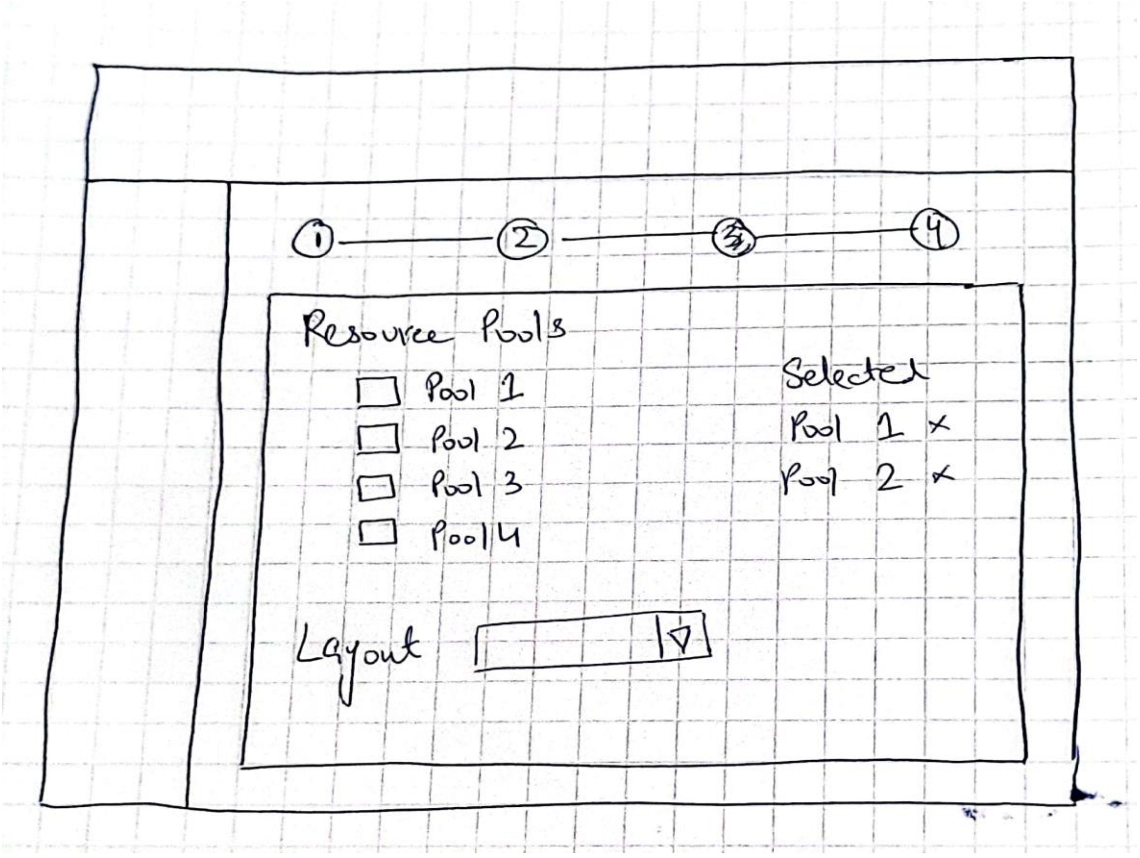


Figure 5.4 *Wireframe for Resources and Layout selection in MM GUI application*

Tool selection:

For prototyping, it was important to select right and suitable web designing tool by keeping in mind the requirements and resources of the project. The tool should be easy-to-use, efficient, and feature packed. **Figma** was selected for prototyping because of the following reasons:

- Team members were already familiar with this tool. So, it saves lots of time.
- It offers a free plan.
- Figma is a web based online tool. So, no need to install anything.
- Team members were working remotely so its real time collaboration feature makes it easy to work together on the design.
- It offers features like previewing and adding comments. This makes it easy to get feedback on the design.
- It offers a variety of pre-designed templates and libraries of design elements.

Tool usage:

Starting to design on Figma is relatively straightforward. Figma has an active community with plenty of tutorials and resources available online. So, the first step to start working with Figma was to visit the Figma website (https://www.figma.com/) and sign up for a free account. An example of Figma workspace is shown in Figure 5.5 below. After logging in, user should explore the Figma interface and get familiarize with the main elements, such as the toolbar etc. Figma offers a range of design tools on the left-hand toolbar. Tools include the frame tool, shape tool, text tool, pen tool, and more. Experimenting with these tools will help in understanding how they work.

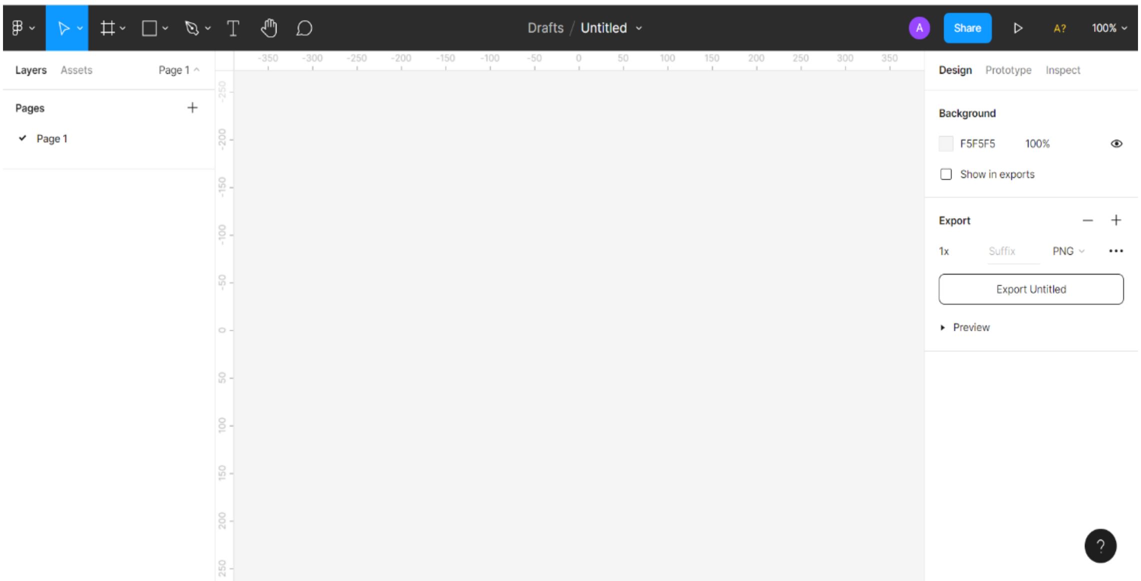
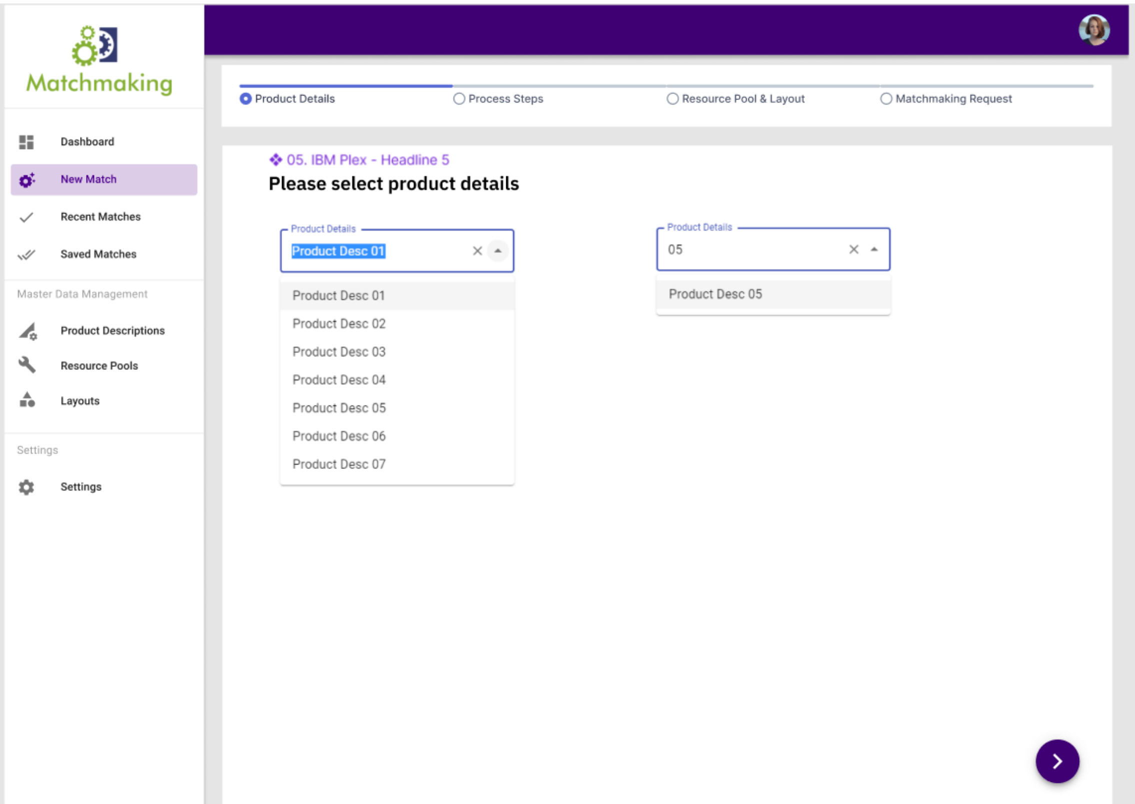


Figure 5.5 *Workspace of Figma*

In Figma, designs are organized using frames and layers. Frames are used to group and contain elements of your design and layers are used to manage the hierarchy and visibility of elements. Different design elements can be added to your canvas, such as shapes, text, images, and icons. These design elements can be customized using fill colors, stroke styles, shadows, and gradients. Figma supports constraints and auto layout, which make it easier to design responsive and flexible components. Once you're satisfied with your design, save your work by clicking the "Save" button. Figma allows you to export your designs in various formats, such as PNG, JPG, SVG, PDF, and more.

Few examples of prototypes developed by Figma are shown below in figures 5.6, 5.7, and 5.8.



❖ 05. IBM Plex - Headline 5

Please select product details

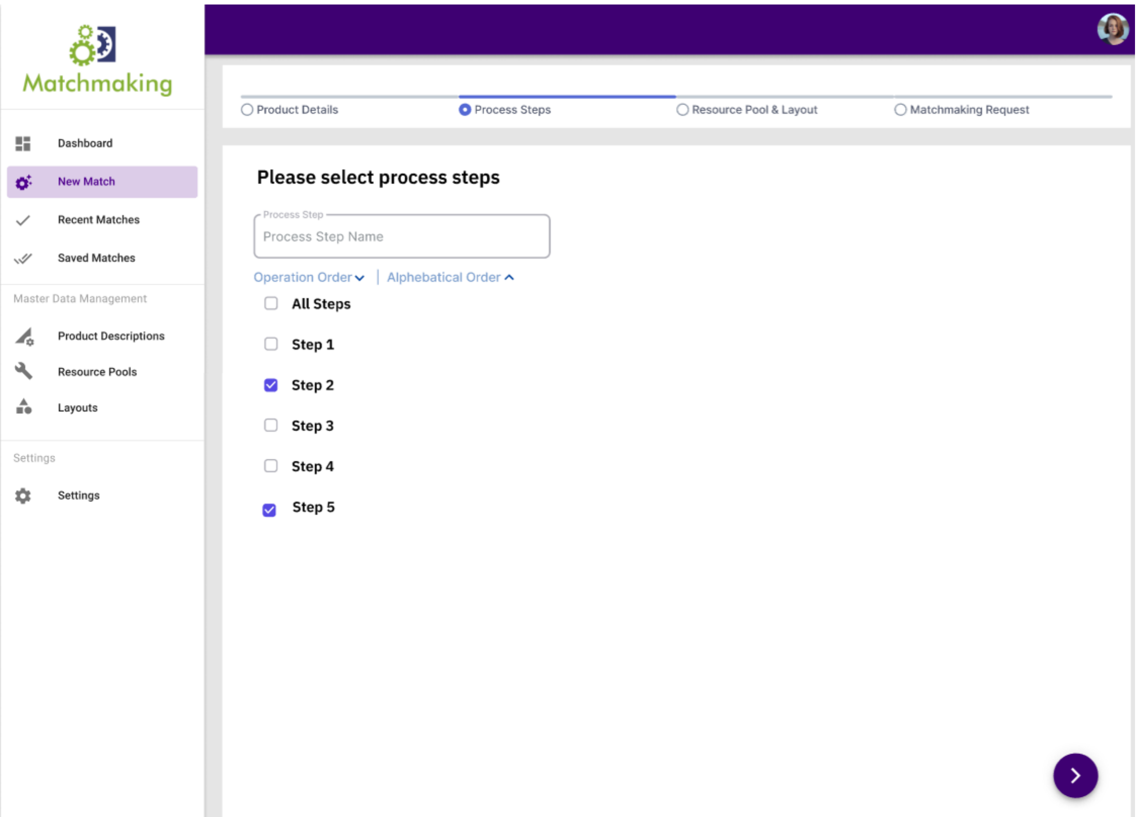
Product Details: Product Desc 01

Product Desc 01
Product Desc 02
Product Desc 03
Product Desc 04
Product Desc 05
Product Desc 06
Product Desc 07

Product Details: 05

Product Desc 05

__Figure 5.6__ Prototype for Product selection feature in MM GUI application



Please select process steps

Process Step: Process Step Name

Operation Order ⌄ | Alphabetical Order ⌃

□ All Steps
□ Step 1
☑ Step 2
□ Step 3
□ Step 4
☑ Step 5

__Figure 5.7__ Prototype for Process steps selection feature in MM GUI application

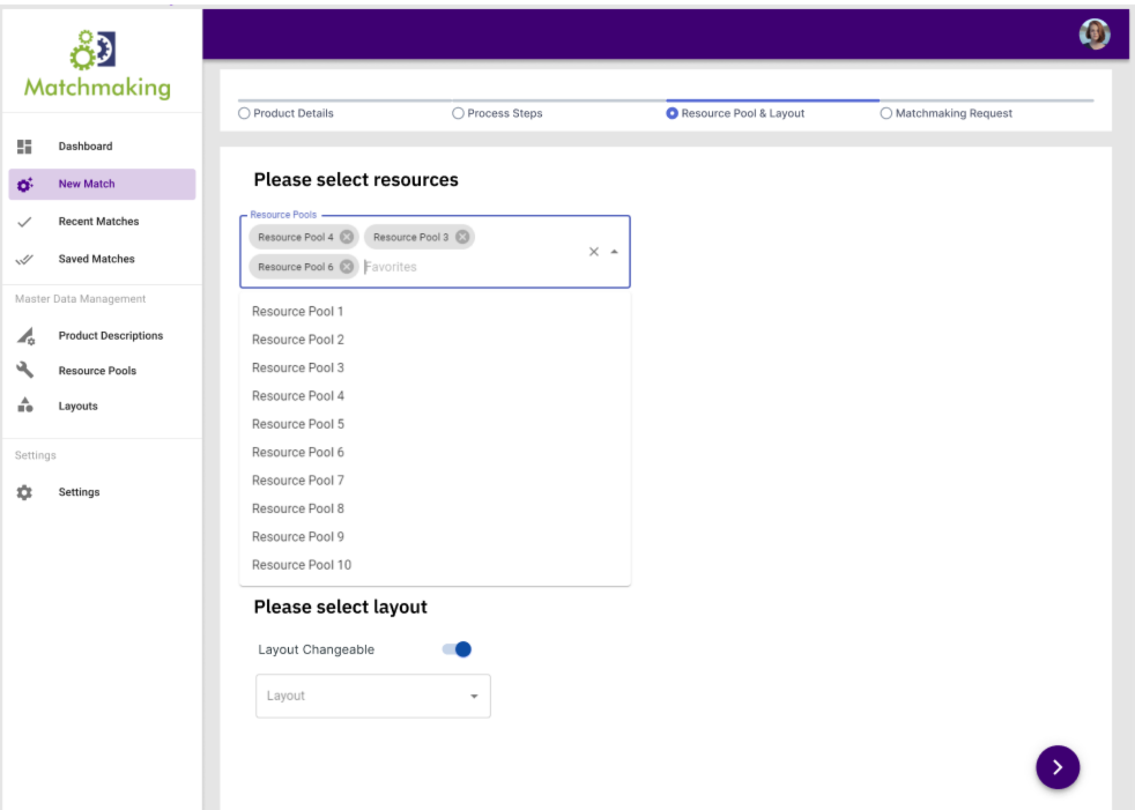


Figure 5.8 Prototype for Resources & Layout selection feature in MM GUI application

5.1.3. Development Stage

After creating web designs for MM GUI application using Figma tool, it was required to select a JavaScript UI framework and start development work. In section 4, a strategy has been proposed for the selection of JavaScript UI framework which can be used for the selection of framework for the development of MM GUI application.

Requirements:

The process of requirements gathering has been discussed in section 5.1.1. All types of functional and non-functional requirements were gathered, and they were prioritized according to need of the project. Some of the gathered requirements relevant to the selection of JavaScript framework are mentioned below:

- Project requires a lightweight front-end library or framework.
- It should allow developers to build reusable and modular UI components to efficiently manage the application's complexity.
- Front-end framework should support real-time data updates and frequent UI changes.
- Framework should have large and active community.

Architecture:

As discussed in section 5.1, MM GUI application needs to communicate RESTful APIs to capability matchmaking service where data is processed, returned, and stored. For MM GUI application, there was a need to store data in a centralized state so that data can be shared easily between different parts of application. So, for MM GUI application it was decided to go for either Flux or Redux architecture to maintain unidirectional data flow in application.

Type of web application:

To maintain high interactivity and real-time updates, it was needed to develop MM GUI application as a single page application (SPA). The main reason to select SPA was separation of concerns. In SPA there is a clear separation between front-end and backend development [44]. Front-end developers can work on the UI part using modern JavaScript frameworks, while backend developers can concentrate on designing robust APIs to serve data. This separation of concerns can lead to more efficient development workflows.

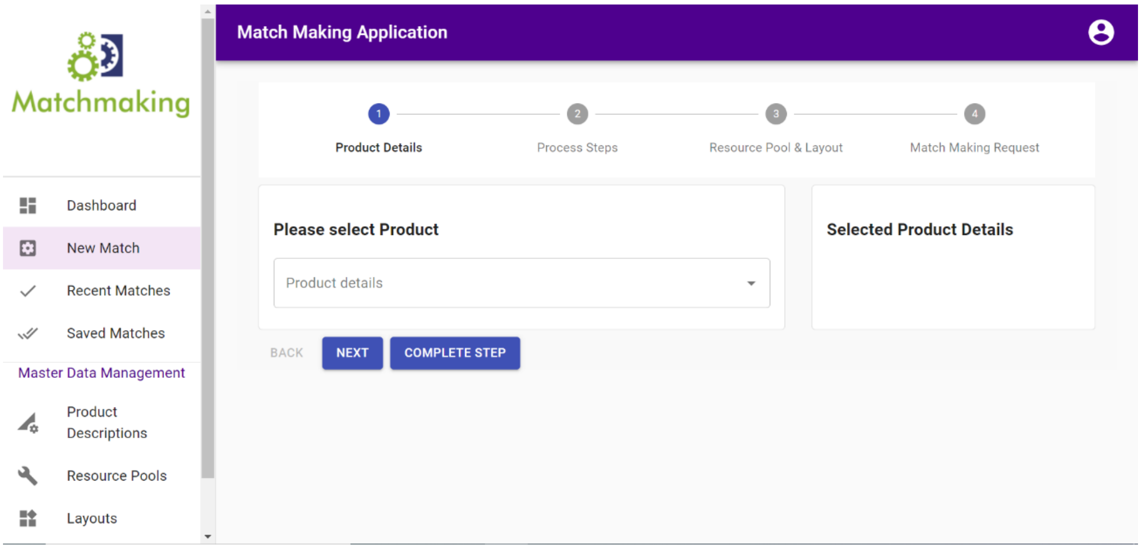
Framework selection:

Considering requirements, architecture and type of web application, React was selected for the development of MM GUI application. React is a popular JavaScript library for building user interfaces, while Redux is a state management library that complements React by providing a predictable and centralized way to manage application state. When building a SPA with React, integrating Redux can enhance your application's state management capabilities.

Result:

By following the proposed approach for the selection process, React was selected for the development of MM GUI application. React aligned well with the specific requirements of our project and it contributes to a more efficient development process. The learning curve of React was manageable and developers were able to leverage their skills to achieve project goals. The active and supportive community of React also helped in finding resources, documentation, and third-party libraries that enhanced development process. The main objective of adopting the proposed approach was to streamline the development process and it was achieved because this approach worked well for MM GUI application and deliver the project on time and covered all required functionalities of application.

Some functionalities of MM GUI application are shown in figures 5.9, 5.10, and 5.11 and their prototypes are displayed in figures 5.6, 5.7, and 5.8 respectively. Final web application seems very close to initial wireframes and prototypes of the application. For evaluation, design elements (colors, typography, icons, buttons etc.) in the final web application are compared with the corresponding elements in the initial design. Besides that, the layout and structure of the final web application is assessed against the wireframes. The positioning of page element is also checked. The user interactions and animations implemented in the final web application is compared with the intended interactions from the initial design. The design consistency was maintained across different pages or sections of the final web application. Any discrepancies which can be seen between the final web application and the initial design is due to scope changes.

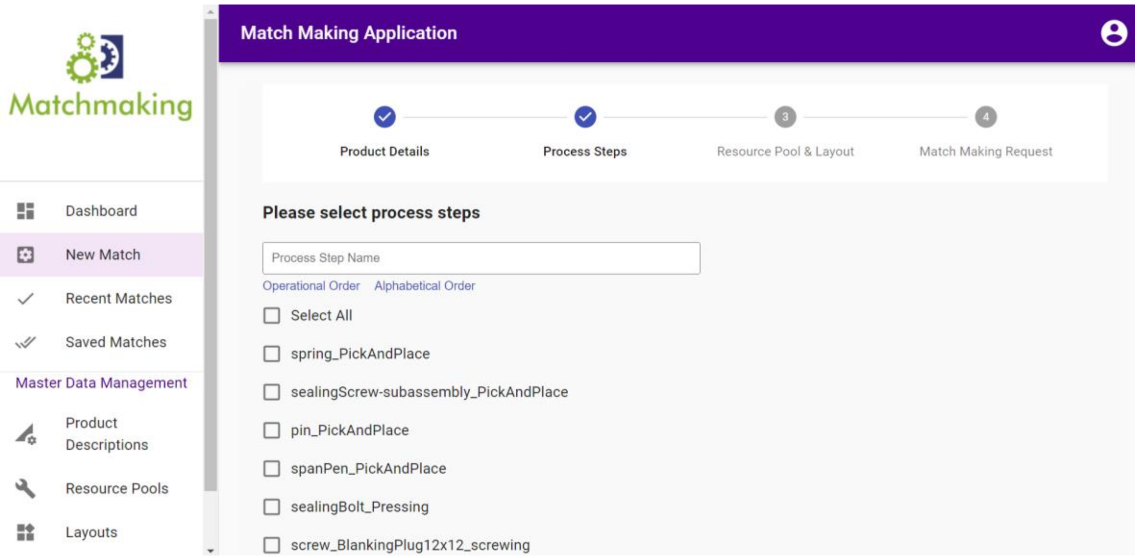


Please select Product

Product details

Selected Product Details

Figure 5.9 *Product selection feature in MM GUI application*

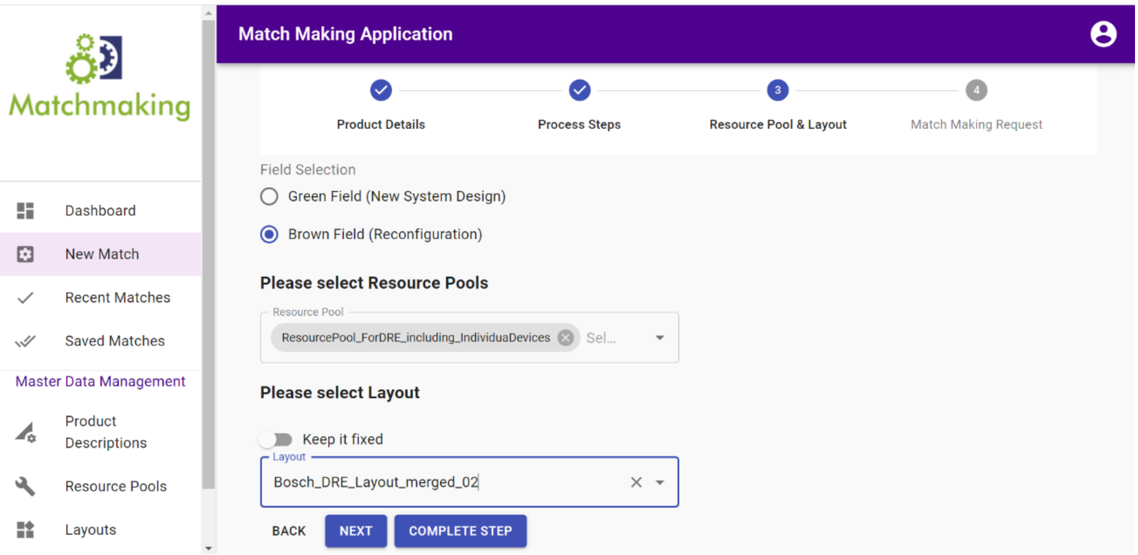


Please select process steps

Process Step Name

□ Select All
□ spring_PickAndPlace
□ sealingScrew-subassembly_PickAndPlace
□ pin_PickAndPlace
□ spanPen_PickAndPlace
□ sealingBolt_Pressing
□ screw_BlankingPlug12x12_screwing

Figure 5.10 *Process steps selection feature in MM GUI application*



Field Selection

○ Green Field (New System Design)

● Brown Field (Reconfiguration)

Please select Resource Pools

Resource Pool: ResourcePool_ForDRE_including_IndividuaDevices Sel...

Please select Layout

Keep it fixed

Layout: Bosch_DRE_Layout_merged_02

Figure 5.11 *Resource Pools & Layout selection feature in MM GUI application*

6. Discussion

Web designing tools are used to streamline the process of creating visually appealing and interactive user interfaces for web front-end applications. By using web designing tools, designers can enhance their workflow, collaborate more effectively, and create visually appealing and functional front-end interfaces for websites and web applications. Similarly, JavaScript frameworks are used to optimize the development process of web front-end applications, enhance the structure of code, and provide tools and conventions for building scalable, maintainable, and feature-rich user interfaces.

The selection process of web designing tool and JavaScript framework is an important step. It is crucial to carefully consider the specific needs of the project and long-term goals to make informed decisions and avoid potential issues in the design and development process. Any misalignments can hinder development speed and negatively impact the scalability and overall success of the web application. Imagine a team has limited time for web designing but they decide to use a complex and feature-rich web design tool that has a steep learning curve. Designers will find it challenging to adapt, leading to inefficiencies in the design process. On the development side, if a team is required to build an application which should be scalable and maintainable further in future, but they ended up using a JavaScript framework known for its performance optimizations but with a relatively small and decreasing community. As the project progresses, the team will encounter compatibility issues with other tools and libraries, and the lack of community support makes it difficult to find timely solutions. The chosen framework, although powerful, comes with unnecessary overhead for the project's scope, slowing down the development process and hindering overall scalability.

In section 3 and 4, guidelines are proposed for the selection of web design tools and frameworks for web development. This thesis work also helped in gaining valuable insights into the strengths and weaknesses of various design tools and JavaScript frameworks, as well as their impact on the development process and end-user experience.

The proposed selection guidelines helped in the selection process of Matchmaking (MM) GUI application. Regarding design tools, approach proposed in section 3 was followed. This approach helped in extracting useful information for the selection of web designing tool from the relevant project requirements, available resources for the project, and type of web designs. After collecting this information, it was easy to select a tool which satisfied all these criteria of the project. After careful examination, it was evident that Figma stands out as the most suitable tool for web designing of MM GUI application. The reason to select Figma was its real-time collaboration, team familiarity, and its features mapping to the requirements and resources of the project.

On the other hand, for the selection of JavaScript framework for MM GUI application React stands out because it was satisfying the evaluation criteria based on project requirements, architecture, and type of web application. React was a suitable choice because it was easy to achieve most of project requirements by following Redux architecture.

This thesis emphasizes the importance of thoughtful and informed decision-making when selecting design tools and JavaScript frameworks. Understanding the unique strengths of each option and aligning them with project goals will lead to more efficient development processes and the delivery of high-quality, user-centric web applications.

The results obtained after following these guidelines were quite satisfactory. The project was delivered on time without any unforeseen circumstances. These guidelines saved lots of time and speed up the overall process. The chosen web designing tool demonstrated exceptional efficiency and productivity, streamlining the design process, and fostering collaboration among team members. The visual consistency was good, reflecting the aptness of the selection approach. Likewise, the chosen JavaScript framework seamlessly integrated with other tools, mitigating compatibility issues. Furthermore, after comparing the final results of web application and initial web designs, it was evident that selection approach was very useful and helped in meeting project goals.

7. Conclusion

In conclusion, this thesis aimed to investigate and evaluate the selection of design tools and JavaScript frameworks for web development. The approach was to cover different factors that influence the web designing and development stages of a front-end application. For selection of web designing tool, the approach proposes to extract useful information from application requirements, available resources, identify and analyse types of web designs required by the application, analyse the available tools to get familiarize with their features and then select the most suitable tool for your application.

To choose a JavaScript UI framework, the guideline proposes to consider the following factors: requirements, select the most suitable architecture for your application, analyse the type of your web application, study available JavaScript framework to know their key features and then choose the most suitable one.

To determine the usefulness of these guidelines, Matchmaking GUI application was implemented. By following the proposed guidelines, Figma was selected as a web designing tool and React was selected for the development of application. This selection guideline makes it easy to achieve project's intended objectives and goals. Most of the required features and functionalities of the project were delivered within the agreed-upon scope, budget, and timeline.

This research work has provided valuable insights into the importance of structured decision-making in the dynamic landscape of web development. However, further research work can be performed in the comprehensive examination of how strictly adhering to these guidelines influences the overall success and long-term maintainability of web front-end applications. While there is recognition of the significance of guidelines in tool and framework selection, there is a lack of in-depth investigation into the real-world implications of guideline adherence. Addressing this research gap will provide nuanced insights into the practical outcomes of following or deviating from established guidelines, offering valuable guidance for developers and decision-makers navigating the complex landscape of web front-end development tool and framework choices.

References

[1] Chris D'Souza, Vincenzo Deufemia, Athula Ginige, Giuseppe D’Souza Polese. “Enabling the generation of web applications from mockups.” *Software, practice & experience,* 2018, Vol.48 (4), pp. 945-973. DOI: 10.1002/spe.2559.

[2] Santosh Maurya, Céline Mougenot, and Yukio Takeda. “Impact of Mixed Reality Implementation on Early-Stage Interactive Product Design Process.” *Journal of engineering design* 32.1 (2021): 1–27. DOI: 10.1080/09544828.2020.1851662.

[3] Andreas Gizas, Sotiris Christodoulou, and Theodore Papatheodorou. “Comparative Evaluation of Javascript Frameworks.” *Proceedings of the 21st International Conference on World Wide Web. ACM*, 2012, pp. 513–514. DOI: 10.1145/2187980.2188103.

[4] Daniel Graziotin, and Pekka Abrahamsson. “Making Sense Out of a Jungle of JavaScript Frameworks: Towards a Practitioner-Friendly Comparative Analysis.” *Lecture Notes in Computer Science (including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2013, Vol. 7983, pp. 334-337. Berlin, Heidelberg: Springer Berlin Heidelberg. ISBN: 9783642392580.

[5] Amantia Pano, Daniel Graziotin, and Pekka Abrahamsson. “Factors and Actors Leading to the Adoption of a JavaScript Framework.” *Empirical software engineering: an international journal,* 2018, Vol.23 (6), pp. 3503-3534. DOI: 10.1007/s10664-018-9613-x.

[6] Anisa Stefi. “Do Developers Make Unbiased Decisions? - The Effect of Mindfulness and Not-Invented-Here Bias on the Adoption of Software Components.” *23rd European Conference on Information Systems, ECIS 2015.* Vol. 2015-.

[7] Jia Zhang, and Jen-Yao Chung. “Mockup-driven fast-prototyping methodology for web application development.” *2003 Symposium on Applications and the Internet, 2003. Proceedings,* 2003, p.410-413, Article 1183086. ISBN: 0769518729. DOI: 10.1109/SAINT.2003.1183086.

[8] Giles Turnbull. *Your Life in Web Apps*, 1st edition. Sebastopol: O’Reilly Media, 2006.

[9] Pawan Vora. *Web Application Design Patterns*, 1st edition. Amsterdam ;: Morgan Kaufmann Publishers/Elsevier, 2009, ISBN : 9786612737350.

[10] D. P. Voorhees, *Guide to Efficient Software Design: An MVC Approach to Concepts, Structures, and Models*. Cham: Springer International Publishing AG, 2020, ISBN: 3030285006.

[11] Adam Boduch, *Flux architecture: learn to build powerful and scalable applications with Flux, the architecture that serves billions of Facebook users every day* (Community experience distilled.), 1st ed. PACKT Publishing, 2016, ISBN: 9781786465818.

[12] redux.js.org. “Redux essentials, part 1: Redux overview and concepts | redux.” (2022), [Online]. Available: https://redux.js.org/tutorials/essentials/part-1-overview-concepts (visited on 01/08/2023).

[13] Michele Aponte. *Building Single Page Applications in . NET Core 3: Jumpstart Coding Using Blazor and C#,* 1st edition, Berkeley, CA: Apress, 2020.

[14] Yongkang Xing, Jethro Shell, Conor Fahy, Tiande Xie, Ho Yan, Kwan, Wenqiang Xie. “Web XR user interface research: design 3D layout framework in static websites.” *Applied sciences,* 2022, Vol.12 (11). DOI: 10.3390/app12115600.

[15] Carlos Rojas. *Building Progressive Web Applications with Vue.js : Reliable, Fast, and Engaging Apps with Vue.js*. Berkeley, CA: Apress, 2019. ISBN: 9781484253342.

[16] Kwang Myoung Joo, Hun Kim. “A study on mobile application UI/UX design of color conversion for the color vision defectives.” *Journal of the Korean Society Design Culture,* 2017, Vol.23 (2), pp. 669-682. DOI: 10.18208/ksdc.2017.23.2.669.

[17] Inayaili de Leon. *Moving to Responsive Web Design: Bring Existing Static Sites into Today’s Multi-Device World with Responsive Web Design*. Berkeley, CA: Apress L. P, 2016. ISBN: 9781484219867.

[18] Andreas Kurniawan, Nunnun Bonafix, and Hendri Hartono. “Design UI/UX mobile games for left hand dominant people.” *Journal of Games, Game Art, and Gamification,* 2021, Vol.5 (2), p.48-53. DOI: 10.21512/jggag.v5i2.7476.

[19] Shane Loeffler, Robert E. Roth, Simon Goring, Amy Myrbo. “Mobile UX design: Learning from the flyover country mobile app.” *Journal of maps,* 2021, Vol.17 (2), pp. 39-50. DOI: 10.1080/17445647.2020.1867247.

[20] Omar Azouz, Nouamane Karioh, and Youssef Lefdaoui. “A systematic mapping study: how can UX design be adapted to improve the design of meaningful gamified solutions?” *International journal of innovation and technology management*, 2021, Vol.18 (6), Article 2130006. DOI: 10.1142/S0219877021300068.

[21] Thamsanqa Keith Miya, and Irene Govender. “UX/UI design of online learning platforms and their impact on learning: a review.” *International Journal of Research In Business and Social Science,* 2022, Vol.11 (10), pp. 316-327. DOI: 10.20525/ijrbs.v11i10.2236.

[22] Russ Ferguson. *Beginning JavaScript: The Ultimate Guide to Modern JavaScript Development*, 3rd edition, Berkeley, CA: Apress L. P, 2019.

[23] Sufyan bin Uzayr, Nicholas Cloud, and Tim Ambler. *JavaScript Frameworks for Modern Web Development: The Essential Frameworks, Libraries, and Tools to Learn Right Now.* Berkeley, CA: Apress L. P, 2019. ISBN: 9781484249949.

[24] Karl E. Wiegers. *Software Requirements*, 2nd ed. Sebastopol: Microsoft Press, 2009, ISBN: 0-7356-3518-8.

[25] Catherine. Courage, and Kathy. Baxter. *Understanding Your Users: a Practical Guide to User Requirements Methods, Tools, and Techniques*, 1st edition. San Francisco, CA: Morgan Kaufmann Publishers, 2005, ISBN: 1-4175-6157-2.

[26] Sagar Ganatra. *React Router Quick Start Guide: Routing in React Applications Made Easy*, 1st edition, Birmingham, UK : Packt Publishing, 2018.

[27] Jia Zhang, C.K. Chang, and Jen-Yao Chung. "Mockup-driven fast-prototyping methodology for web requirements engineering." *Proceedings - IEEE Computer Society's International Computer Software and Applications Conference*, 2003, pp. 263-268. ISBN: 9780769520209. DOI: 10.1109/CMPSAC.2003.1245352.

[28] Maya. Stoeva. "Model and Prototype of Interactive Assistant for Compliant Interface Development - MayUI Tool." *2021 International Conference Automatics and Informatics (ICAI). Piscataway: IEEE,* 2021. pp. 295–300. ISBN: 9781665426619. DOI: 10.1109/ICAI52893.2021.9639601.

[29] adobe.com. "Get Started with Adobe XD | Adobe." (2023), [Online]. Available: *https://helpx.adobe.com/xd/get-started.html* (visited on 07/18/2023).

[30] figma.com. "Prototyping | Figma." (2023), [Online]. Available: *https://www.figma.com/prototyping/* (visited on 07/20/2023).

[31] balsamiq.com. "Balsamiq Wireframes | balsamiq." (2023), [Online]. Available: *https://balsamiq.com/wireframes/* (visited on 07/20/2023).

[32] sketch.com. "Prototyping | Sketch." (2023), [Online]. Available: *https://www.sketch.com/prototype/* (visited on 07/20/2023).

[33] invisionapp.com. "Prototype | invision." (2023), [Online]. Available: *https://www.invisionapp.com/product/prototype/* (visited on 07/20/2023).

[34] Florent Vilmart, Giordano Scalzo, and Sergio De Simone. *Hands-On Design Patterns with Swift.* 1st edition. Packt Publishing, 2018. ISBN : 1-78913-556-7.

[35] react.dev. "Quick Start | React." (2023), [Online]. Available: *https://react.dev/learn/* (visited on 08/10/2023).

[36] Daniel Bugl. *Learn React Hooks: Build and Refactor Modern React. Js Applications Using Hooks*. Birmingham: Packt Publishing, Limited, 2019. ISBN: 9781838641443.

[37] Azat Mardan. *React Quickly: Painless Web Apps with React, JSX, Redux, and GraphQL*, 1st edition, Shelter Island, New York : Manning Publications, 2017.

[38] Adam Boduch. *React Material-UI Cookbook*. Birmingham: Packt Publishing, Limited, 2019. ISBN: 9781789615227.

[39] angular.io. "What is Angular | Angular." (2023), [Online]. Available: *https://angular.io/docs/* (visited on 08/20/2023).

[40] Rodrigo Branas. *AngularJS Essentials*. Olton: Packt Publishing, 2014. ISBN: 9781783980086.

[41] vuejs.org. “Introduction | Vue.js.” (2023), [Online]. Available: *https://vuejs.org/guide/introduction.html/* (visited on 08/20/2023).

[42] John Au-Yeung. *Vue. Js 3 by Example: Blueprints to Learn Vue Web Development, Full-Stack Development, and Cross-Platform Development Quickly*. Birmingham: Packt Publishing, Limited, 2021. ISBN: 9781838826345.

[43] Eeva Järvenpää, Niko Siltala, Otto Hylli, Minna Lanz. ”Implementation of Capability Matchmaking Software Facilitating Faster Production System Design and Reconfiguration Planning.” *Journal of manufacturing systems*, 2019, Vol.53, pp. 261-270. DOI: 10.1016/j.jmsy.2019.10.003.

[44] Hroje Puškarić, Aleksandar Đorđević, Miladin Stefanović, Marija Zahar Đorđević. “Development of Web based application using SPA architecture.” *Proceedings on engineering sciences (Online),* 2019, Vol.1 (2), pp. 457-464. DOI: 10.24874/PES01.02.044.

[45] Chris Minnick. *Beginning React JS Foundations Building User Interfaces with ReactJS*. Newark: John Wiley & Sons, Incorporated, 2022, pp. 11–22, ISBN: 9781119685548.