



## How to Properly Setup a Web Application UI stack

### How to Properly Set Up a Web Application UI Stack

Before you can develop the perfect web application, you need to choose and set up an appropriate user interface (UI) technology stack. This process involves several key decisions, with the technology you choose laying the foundations for the success of your project. From initial research and setup considerations through to choosing relevant tools to enable continuous development, it's important to make the right decisions to ensure best practice outcomes.

#### Problem Statement

Small businesses and startups often struggle to transform their creative ideas into robust applications. While having a good idea is a great first step, limited budgets and resources can be major hurdles when it comes to application development. Inconsistencies often exist between UI stacks and the final goals of the application, with multiple standards, tools, libraries, and frameworks often getting in the way of a harmonious user experience.

When you're deciding how to build a software product, it's essential to simplify the process as much as possible and choose a UI stack that's capable of meeting the unique demands of your project. It's important to be aware of the limitations of particular technologies and put the user front and center with every decision you make.

Unlike server-side programming, the UI stack is entirely based on the end user's needs and almost completely dependent on client-side programming. This side of development focuses on how the website is presented to the end-user, including all interactions between the end user and the application that involve an exchange of information with the server.

Key problems to address:

- Inconsistent user experience across a web application
- Inconsistent standards, tools, libraries, and frameworks between technologies
- Inconsistent demands between server-side and client-side development

UI development is about creating the best possible user experience, with UI components designed to present information most beneficially and interact efficiently with backend operations where needed.



## What is a Web Application UI Stack?

The web development process can be extremely varied, with different tools and methodologies applied throughout the process. There are two sides to web development: the client-side or front-end based on the user experience, and the server-side or back-end providing the framework for the application. A web application UI stack is a combination of individual software components used to create and enhance the user experience.

## Solution

With new buzzwords and methodologies being introduced all the time, developers need to sit back and make careful decisions before moving forward with any project. Rather than relying on past experience or personal preferences, it's important to find relevant new solutions that meet the specific demands of the project in question.

The choice you make should be dependent on the type of application and the intended processing load. Security and scalability are also important, as is the overall development cost and time to market considerations.

## Initial setup

When choosing a web application UI stack for your project, it's essential to take time with the initial setup. Early mistakes can have a snowball effect, so make considered choices and implement technology relevant to the problem at hand. It's important to avoid unnecessary components during this early phase, and only implement libraries and frameworks that are relevant.

### a) Ensure best practices

Following these best practices will help you develop a consistent user experience that adds value:

- **Keep it simple** - From the design to the experience, a good user interface needs to be minimal and elegant.
- **Communicate with specific intent**- Talk directly to the end user and apply a problem-solving mindset.
- **Reduce the learning curve** - Find novel ways to engage new users and introduce them to key processes.
- **Stay consistent**- Find underlying abstractions and conform to known industry standards and controls.
- **Leave breadcrumbs**- Let the user know exactly where they are so they can navigate effectively.



# Owl Branch Development

- **Follow the principles of design** - Use contrast, repetition, alignment, and proximity to your advantage.

## b) Apply relevant tools

There are two prominent technology stack components used for client-side application development, with additional technologies needed to handle the request and exchange process between the end user and the backend operations. Along with these three core components, several open-source tools are available to enable continuous development.

- **Hypertext Markup Language (HTML) and Cascading Style Sheets (CSS).** This technology is concerned with presentation, with HTML formatting web pages and CSS presenting your content in a range of styles. HTML and CSS are responsible for what the end user sees.
- **Javascript (JS).** This technology is used to make your web pages and applications interactive. Several popular JavaScript libraries can be used to enhance and personalize this process, including jQuery and React.js. JS frameworks such as Vue and Angular can also be employed to speed up development. JS is responsible for how the end user acts.
- **Technology for client requests.** Applications are increasingly concerned with client requests, with individual technologies needed to handle the exchange between back-end and front-end operations. There are currently two major players in this field: Apache, and Nginx.

## Continuous Development

Agile software development aims to deliver value, accelerate time to market, and improve code quality across iterations. Several approaches have been developed to enable these goals, including continuous development. Continuous development is an umbrella term that includes continuous integration, continuous delivery, and continuous deployment.

This entire process is dedicated to small build cycles with short sprints and immediate feedback between processes. The goal of this strategy is to have the code ready at any time, with feature sets available, tested, debugged, and ready to deploy without necessarily being complete.

Smaller and more manageable cycles allow for a quicker turnaround time to identify and fix bugs when new features are developed. While this process can be deployed in any software and production environment, open-source tools and technologies, such as Jenkins, can be very advantageous.



# Owl Branch Development

## Continuous Integration Setup

The overall concept of continuous integration is to enable multiple developers working on different aspects of a project to function as a whole. Rather than testing software on individual developer's machines, building and testing procedures are automated and integrated as code is changed. This is crucial for websites and online applications where regular changes are a normal part of the operation.

## Continuous Delivery Setup

Continuous delivery goes one step further, with the actual software release automated to improve the iteration cycle. This step typically involves packaging the software for deployment in a production-like environment. This is highly useful for web applications because it allows code to be released as soon as it's been tested. Once again, a continuous feedback loop is the end goal.

## Continuous Deployment Setup

In a typical continuous deployment model, each change made to the code is automatically deployed through to the production environment. While treating the end user as the tester can be problematic in some situations, it allows for a much quicker release and also benefits end users by giving them access to new features and applications. In many ways, continuous deployment is the perfect fit for web sites and APIs.

## Conclusion

Choosing and setting up a web application stack is a critical part of the UI development life cycle. Each stage is important, from the early days of research and best practice analysis through to obtaining stack components for continuous client-side development. To be effective, a UI web application stack needs to meet the current demands of end users and stay agile as technologies and user expectations change over time.

Ready for more? Ready to get started?

[Contact](#) us today!