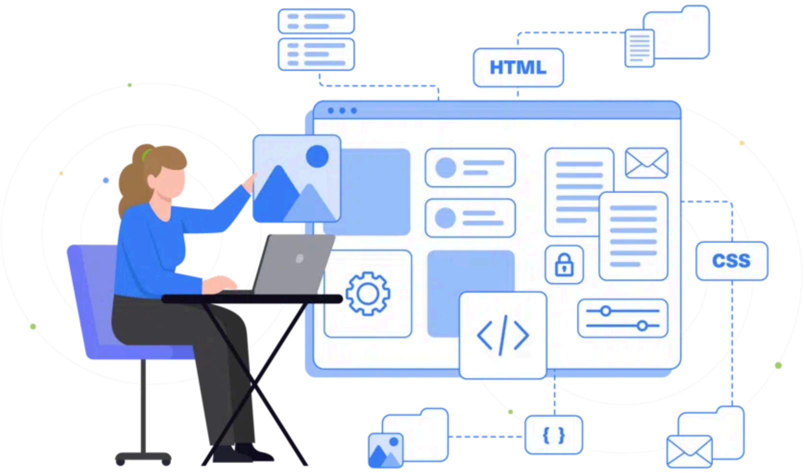
A Comprehensive Guide to Modern Web Development Stacks

#development



by Pavel Gorbachenko
Co-Founder, CEO

16 minutes

A typical web application consists at minimum of a database, web servers, operating system, and the development environment. Any firm developing websites should have the knowledge and human resources to implement these functions by a number of available platforms in the market. It is also important for clients of website development companies to understand the tech stacks used by their web developers, as they affect the user experience and application functionality. To help article we delve into the purpose of web frameworks, factors to consider while choosing one, and learn some of the most popular stacks along with their use-cases in modern web development.

Popular Web Development Stacks

Different stacks exist to handle different tasks related to web application development and operations. Following are some of the top stacks for front-end, back-end, and full-stack web development:

Front-End Frameworks

The choice for front-end development is typically limited to JavaScript frameworks as it is the only programming language that can be referenced within the Open Web Platform.
Popular Javascript frameworks for the front-end include:

1. Vue (Vue.js)

An open-source, model-view-view-model progressive Javascript framework for generating user interfaces. Vue relies on a user-friendly core library and a collection of supporting libraries that simplify the deployment and management of single-page applications. The framework can be adopted in increments, making it easy to scale up and down depending on workload and use-case.

maintained by Google and an open-source community for developing single-page websites. Angular is fully extensible and interfaces seamlessly with other libraries.

3. React

An open-source declarative Javascript used to develop UI components and generate User Interfaces. React is maintained by Facebook and an open-source community as a base of development for mobile applications and single-page websites.

Back-end Frameworks

Popularity is often considered a good measure of viability for back-end frameworks since the most used tech stacks get better community support. Some popular back-end frameworks include:

1. Django

A free, open-source, python-based web framework whose architectural pattern follows the model-template-views format. The high-level framework encourages clean, practical design while offering rapid development and deployment. With Django, developers can build every web application's web features as well as Forms, User Authentication & permissions, Caching, and Data Serialization.

2. Node (Node.js)

An open-source, cross-platform Javascript runtime that executes JS code externally through the chrome

3. Express (Express.js)

A minimal, flexible web application framework built upon Node.js to provide extensive sets of features for application back ends. Express.js enables the rapid development of APIs through implementing various HTTP utility methods and middleware.

Full-Stack Frameworks

In a full-stack development, developer teams are responsible for provisioning servers, the underlying platform, user interface and all other components required for a functional website.
Some popular full-stack frameworks include:

Django, Vue.js and Python

This stack builds upon Python, the general-purpose programming language, to create a framework that's efficient, easy to read, and easy to learn.
This OS-agnostic stack enables the building of quick prototypes through leveraging features such as **Class-Based Views** and **Object Relational Model (ORM)**. With Django ORM, teams can abstract the complexities of creating and managing databases so that they can build databases without writing SQL code. Using Vue.js for the front-end, this framework combines the best of React.js and Angular.js into one library for building Single Page Applications.

MEAN/MERN/MEVN

components are:

- **Database** - MongoDB
- **Server-side framework** - Express.js
- **Client-side framework** - Angular/React/Vue
- **Javascript Runtime** - Node.js

LAMP

A generic software stack model with interchangeable components for delivering high-performance web applications. Commonly used technologies of a LAMP stack consists of:

- **Operating System** - Linux
- **Server** - Apache
- **Database** - MySQL
- **Client-side programming** - PHP

Flutter

An open-source software development kit used to develop cross-platform web applications for Mac, Windows, iOS, Linux, and Android. The kit comes in a friendly User Interface that enables developers to create natively compiled applications for multiple devices from a unified codebase. Flutter is built with layered architecture for full customization, flexible design and rapid front & back-end rendering.

Serverless

Serverless Stack (SST), organizations can focus on building clean code while a cloud service provider executes functions that allocate resources dynamically.

Factors to Consider When Choosing a Web Development Stack

Choosing the right tech stack to operationalize a web application may seem a herculean task, given the complex and dynamic nature of modern software production. It is important to select the right combination of tools and techniques to operationalize an application as these determine the success of the product in deployment.
Here are a few aspects worth factoring in when deciding on a web development tech stack:

Project Type

The scope and type of project determine the time and effort investment required to deploy the application.
For instance, Minimum Viable Products require simple, rapid development with flexible syntax, so organizations developing MVPs could benefit from Ruby on Rails. On the other hand, projects on Machine Learning, Big Data, and Artificial Intelligence are best handled with Python since it

build upon the Java programming language for the range of features and benefits it offers.

Human Resource Talent Pool

It is important to have skillsets in the organization who are familiar with the chosen technology stack. Adding new techniques to the toolchain requires staff upskilling or recruitment, thereby raising operational expenses. Some tools, however, require transferable skills which can be picked up by staff during on-the-job training or transitioning to a new role.

Support for the Stack

Each popular technology has a community that turns out to be the biggest contributor through various phases of the technology's growth. When selecting a tech stack, organizations should assess whether the technology is supported with sufficient documentation, information, and remediation forums. Additionally, tech stacks should enforce strict source control, with a formalized and standardized versioning system for updates and patches.

Time-to-Market

Every tech startup and software company considers Time-to-Market a critical factor in development since quicker development and deployment offer a competitive advantage.

include:

- out-of-the-box functionalities
- developer availability
- third-party integrations
- documentation and support

Components of a Web Development Stack

A web application is generally divided into two components: the server-side (back-end) and the client-side (front-end).
Each of these components is composed of consecutive layers of technology implementing different functionalities built atop each other forming a stack. The tech stack is, therefore, the combination of multiple technology services used in the creation of the data ecosystem that helps to run an application. Due to its importance in building an application, choosing the tech stack is often considered one of the most critical factors in application design and development.
Besides influencing the technical factors, the tech stack also helps with the following -

- Identify the human skills the organization should hire
- Projected efficiency of development teams

of a web application and how they relate to web development tech stacks.

The Front-End

An application's front-end encompasses various components that the end-user can see and interact directly with.

Apart from the user interface, front-end development also utilizes web development technologies running on the **Open Web Platform** to architect websites and applications.

An application's front-end is composed of three main components:

1. **User Interface -** the visual representation of information generated by front-end software, providing a graphical means of viewing, accessing, and editing website data.

2. **Markup Language -** the set of symbols, rules for how they should be used, and the instructions implemented by each symbol to deliver information to the user. Typically, an HTML markup language is widely used in websites to inform browsers and viewing software how to read and display content.

3. **Scripting/Programming Language -** these are the tools that give life to dynamic websites. These languages are meant to run as part of the application, inserting interactive elements that make the user experience more engaging.

programming are constantly changing. Additionally, users rely on different devices with varying screen sizes, resolutions and refresh rates, making them a complex consideration during development.

The Back-End

An application's back-end or server-side contains various processes that run under the hood to service user requests and transactions. Back-end frameworks enable communication between the web service and a database where the site's content is stored and fetched from.

A back-end also involves processes involved in implementing business logic and accessing other IT resources such as cloud services and file servers.

An application's back end is a stack of multiple layers, including:

1. **Application Logic -** an application logic uses a server-side programming language to develop the application. These are coding platforms used to develop the instructions that guide web servers on how to handle user requests. These languages are usually part of a framework that simplifies coding by eliminating repetition and the need to reinvent the wheel.

2. **File System -** every web application needs a file storage platform to manage the system and user data. Modern web applications are data-driven, where IT operations and development

3. **Database -** a structured data set that enables the organization, storage, and access of information on products, transactions, user identities and other data generated by the application.

4. **Hosting -** the service that allows organizations to publish and expose their web services to the internet. Applications could be hosted in-house, using public cloud offerings or through third-party hosting. The right hosting provider is chosen based on the security, speed and reliability of their services.

5. **Web Server -** the component of the web application that coordinates communication between users and the back-end service. The web server relies on transfer protocols (HTTP/HTTPS) to transmit data between the front and back ends.

6. **Web Development Frameworks -** these frameworks provide templates, structures and components to reduce the time required for building back-ends and front-ends.

7. **Operating System -** every server has an Operating System installed that acts as an interface between various development software and hardware resources.

The growth of application-driven business processes has led to the emergence of different tools, techniques, and mechanisms of implementing web applications for different use-cases. As a result, every organization/developer has its own preference for a web development tech stack that solves its purpose.

With the disappearance of the classic web model, various stacks have evolved and embraced to solve common web application use-cases in recent times. In the current technology landscape, there are a plethora of developed frameworks and tools that focus on various critical aspects, such as scalability, user-friendliness, expandability, and stability of web applications. For beginners, while this offers a range of options, this often leads to confusion when it comes to choosing the right tech stack. This section explores key considerations of choosing a web stack, along with outlining some of the popular web development stacks in the market today.

Summary

Finding the perfect tech stack for an application's deployment environment depends on a number of factors. These include the type of application, time-to-market, skill pool, and support for the tech stack. While there are a number of full-stack frameworks to choose from, the Django, Vue.js, Python framework is increasingly gaining traction for data-driven since Vue handles Javascript, reducing the workload on developers. This post has explored the various components of a web application, and how they are implemented with a tech framework. Talk to our representatives to learn more about how choosing the right stack can help improve product development for your organization.

At Enkonix, we take pride in having helped multiple organizations develop complex web applications or help with expert consulting in their web development journey. Whether you are looking to choose the right tech stack, hire a team of skilled programmers, or need help to manage your project, Enkonix can help. Contact us today to know more.

Recent topics

#development

Everything You Need to Know About Legacy Code

The term Legacy Code usually refers to an application source code that has reached or crossed an end of the support cycle. Often, this may also imply that parts or whole of the application's supported technology stack are either too complex, outdated, or irrelevant to

As applications rely on specific languages, libraries, compilers, and operating architecture, a Legacy Code relates to either of such components to be superseded by modern technologies. Besides these, developers often treat a source code as Legacy which is complex to debug or change.

10 minutes

Keep reading →

Pavel Gorbachenko

Co-Founder, CEO

#development

Code

Refactoring: Why, When, and How Explained

correctly, the end-user will not notice that the code has changed other than seeing improvements in responsiveness. The investment required for refactoring could be recovered by reducing maintenance activities and the associated costs of regression testing activities. The refactoring exercise should have a clear goal and a plan of how to achieve that goal. Without careful planning, refactoring can result in the expenditure of effort that does not deliver noticeable results.

12 minutes

Keep reading →

Pavel Gorbachenko

Co-Founder, CEO

#development **#product strategy**

Technology Consulting for Startups: For Operational Efficiency and Growth

Until a few years ago, tech consulting was aimed towards optimizing technical opportunities, reducing costs, etc. However, it has now evolved to cover much more, ranging from initial phases of Digital a critical enabler to achieve operational efficiency while they compete with other established rivals. By providing a comprehensive consultancy across multiple technology segments, we help organizations achieve a strong foundation to commercialize their business ideas.

11 minutes

Keep reading

Pavel Gorbachenko

Co-Founder, CEO

Table of contents

#development

What is the Rapid Application Development Process?

Rapid Application Development, or RAD for short, is a methodology for fast application production using an iterative software development process. We've listed the pros and cons of the rapid application development methodology in this article.

#development **#project management**

Agile or Traditional? Forming the Right Software Development Team Structure

One common factor identified as the cause of failure of several IT projects is an inefficient software development team structure. In this responsibilities.

#design #development #product strategy

Project Scope: What is it, and how is it defined?

The project scope provides the foundation for the planning process, defining the endpoints that show how to define if completion is successful while cognizant of interdependencies between processes and resourcing and any critical stages in the project's execution.

Pick category to explore more

#company news #design #development #product strategy

#project management

away from putting that project to work. Send us a note

Lobortis elementum nibh tellus molestie nunc non blandit massa. Augue ut lectus arcu bibendum at. Ultrices neque ornare aenean euismod elementum. Enim nunc faucibus a pellentesque sit.

Contact us

Expertise

eLearning Software Development

Custom Sports Software Development

Real Estate Software Development

Services

Uber-like App Development

Custom Software Development

Custom Mobile App Development

Blog

Design

Development

Company news

Product strategy

Project management

Careers

Lead Generation Specialist

Manual QA Engineer

Business/System Analyst