MODERN FRAMEWORKS FOR WEB-APPLICATION DEVELOPMENT

ANNOTATION

Currently, web programming technologies continue to develop at a rapid pace and it is important to know about the most popular and effective frameworks that are available today. Frameworks have become an integral part of the web development process due to the growing standards of web applications and the complexity of the technologies used. When choosing the right framework, there are many factors to consider: project requirements, team experience, and ever-changing industry trends. The framework, as a rule, contains only basic software modules, and all project-specific components are already implemented on their basis by the developer. Thus, not only high development speed is achieved, but also high productivity and reliability of solutions. If we compare frameworks with other classes of platforms - SaaS, CMS, or CMF - then frameworks are much more efficient to use in projects with complex business logic and with high requirements for speed, reliability, and security.

The article discusses the types of popular frameworks, technologies (MVC), and tools designed for developing web applications taking into account modern standards. The characteristics of architectures and the type of frameworks are given. The popular server and client frameworks and programming languages with which they work are considered. The advantages and disadvantages of each of the considered tools are noted.

Key words: web technologies, framework, web application development, MVC, artificial intelligence, artificial neural networks.

Introduction. In the modern world, it is difficult to imagine a large company and organization without its own web application or website. Along with the interest of organizations, the number of libraries and frameworks aimed at facilitating the development of high-quality web applications is growing. Frameworks have greatly changed the world of programming and have become an integral part of web development, as application standards are constantly being modified and, as a result, the complexity of the technology is growing. Using ready-made tools and libraries approved by thousands of developers around the world is a reasonable approach to create multifunctional and interactive web applications. Choosing a framework is not the most difficult task, but their merits and disadvantages should be taken into account when choosing.

It also describes approaches to the development of artificial intelligence through the training of artificial neural networks. A list of available services and programming languages that are used to develop artificial intelligence is provided. Using the capabilities of these frameworks, developers can create intelligent web applications that provide a smooth user interface and bring real benefits to both businesses and consumers.
**Architecture and types of frameworks.**

A framework (“framework”, "structure”) is a dynamically replenished library of a programming language, which contains its basic modules. Frameworks are created to simplify the processes of developing applications, websites, and services. In order not to write a module in an application from scratch, it is much easier to turn to ready-made framework templates, which form the developer's working environment.

The architecture of almost all frameworks is based on the decomposition of several separate layers (applications, modules, etc.) of the project. This means that you can extend the functionality of the application based on your needs and use the modified version together with the framework code or use third-party applications. This flexibility is one of the key advantages of using frameworks [1].

MVC (Eng. Model-View-Controller - "Model- View-Controller") is a software architectural pattern[1] commonly used for developing user interfaces that divide the related program logic into three interconnected elements. This is done to separate internal representations of information from the ways information is presented to and accepted from the user.[2][3] The structure of the software architectural pattern is shown in Figure 1.

![Fig.1: MVC - architectural software template](image)

The model contains all the data and levels of business logic, its rules and functions. The view is responsible for the visual display of data, for example: charts, graphs, etc. The controller simply transforms the data for the commands of the previous two components [2].

Frameworks have two main functions: working on the server side (backend) and working on the client side (frontend). The client part is mainly HTML, CSS and JavaScript and it is considered as a way of presenting information to users. The server part, as a rule, is the most complex and describes the entire business logic of the application. Web development of the backend can be performed in many languages, such as Java, Python, PHP, etc.

**Server frameworks**

The rules and architecture of such frameworks do not make it possible to create a web application with a rich interface. They are limited in their functionality, but you can still create simple pages and different forms. They can also generate output data and be responsible for security in case of attacks. All this can definitely simplify the development process. Server frameworks are mainly responsible for individual, but critical parts of the application, without which it will not be able to work normally. Here are some of the most popular frameworks and the languages they work with Django - Python, Zend - PHP.

**Django**

A high-level framework, which is not only a quick solution in web development, including everything necessary for high-quality code and transparent writing but also an excellent platform for working with the clientele of a particular business. At the same time, it is convenient for developers.

Django implements the DRY - Don't Repeat Yourself principle. That is, when using Django, you do not need to rewrite the same code several times. The framework allows you to create a website from components. Thanks to this, the time for creating websites is reduced.

The framework copes with a large number of tasks and increased loads. It is used to create:

- CRM systems (Customer Relationship Management) - customer relationship management systems;
- CMS-systems (Content Management System) - content management systems [3];
- communication platforms;
- room booking services;
Django is also suitable for creating algorithmic generators, platforms for electronic mailings, verification systems, filtering systems with dynamic rules and complex parameters, platforms for data analysis and complex calculations, machine learning.

Object-Relational Mapping (ORM) is implemented, which ensures the interaction of the application with databases (DB). ORM automatically transfers data from a database, such as PostgreSQL or MySQL, to objects that are used in the application code. It includes mechanisms to prevent widespread attacks like SQL injection (XSS, English Cross-Site Scripting) and forgery of cross-site requests (CSRF, or XSRF, English Cross-Site Request Forgery) [4].

Django REST Framework, which is often abbreviated to DRF, is a library for building an application programming interface, API (Application Programming Interface). It has a modular and customizable architecture that works well for creating both simple and complex APIs. In DRF, authentication and permission policies are available "out of the box". It comes with basic classes for CRUD operations and a built-in utility for testing the API being developed [5].

Advantages:
- a lot of libraries, which allows you not to write basic functionality, but only to import the necessary;
- detailed documentation and friendly communication, so it is always much easier to find ready-made solutions or get help from the community;
- Django lets you start small and scale as needed.

Disadvantages:
- Django does not support WebSockets, so it is poorly suited for real-time operation;
- ready-made libraries often reduce flexibility, Django is a large and monolithic framework, and its components are deployed together.

Zend Framework is an open-source object-oriented web application environment, which is a set of professional PHP-based packages developed and maintained by Zend. This is a set of classes, most of which are not tied to each other, which means that they can be used as spare parts in other projects, or completely. This framework has everything you need in a modern web project. This framework is usually most used in the development of large commercial projects. Companies such as Google, Microsoft, and StrikeIron collaborate with Zend, providing interfaces for web services and other technologies that they want to make available to Zend developers. The company participates in the development of the core of the PHP programming language [6].

The framework uses various packages, using Composer as part of its package dependency managers. Some of them are PHPUnit for testing all packages, and Travis CI for continuous integration services. The Zend Framework provides users with MVC support in combination with the Front Controller solution. Zend Framework follows PHP-FIG standards and includes a PSR-7 implementation for HTTP message interfaces. Support for several database systems and vendors, including MariaDB, MySQL, Oracle, IBM DB2, Microsoft SQL Server, PostgreSQL, SQLite, and Informix Dynamic Server [7].

With the help of Remote Procedure Call (RPC) and REST services (English: Representational State Transfer - "transfer of the representation state") Zend Apigility helps developers create and document APIs.

Zend Server provides improved performance for PHP, and especially Zend Framework applications, by speeding up the opcode and multiple caching capabilities, and also includes application monitoring and diagnostics tools. Zend Studio is an Integrated Application Development Environment (IDE), which includes functions specifically designed to simplify working with the Zend Framework. It provides MVC representation, MVC code generation based on Zend_Tool (a component of the Zend Framework), provides formatting tools, code analysis and correction, parameter assistance, and much more [8].

Advantages:
- great for developing commercial web applications;
- object-oriented approach to development;
- unrelated components for reuse in projects.

Disadvantages:
- heavy and resource-intensive, as it provides the developer with a lot of features and functions;
- not suitable for rapid project development;
- there are not enough materials for the Russian-speaking segment of developers.
Client frameworks

Unlike server-side frameworks, client frameworks have nothing to do with application logic. This type of framework works in the browser. With their help, you can improve and implement new user interfaces. Front-end frameworks allow you to create different animations and single-page applications. All client frameworks differ in functionality and usage. Let's take a closer look at some of them using the JavaScript language.

React

ReactJS is a JavaScript library created by Facebook in 2013, it is excellent for creating large-scale web applications where data can change on a regular basis.

React introduced the concept of a virtual DOM, which is a web page in a browser. React has its own virtual DOM, which controls the actual browser DOM and, since it is much faster than the browser DOM, significantly improves performance. DOM React can create more than 200,000 nodes per second, which exceeds the average number of nodes for most sites. It can also recreate changes by using the Diffing algorithm, which is able to reduce the calculation of the difference from complexity O(n^3) to O(n). DOM allows you to combine HTML, XHTML, or XML documents according to certain criteria, most often into a tree, so React is great for web browsers when analyzing various elements of web applications [13].

One of the great features of React is the introduction of JSX (JavaScript programming language extensions). It is necessary to understand: JSX is not HTML and not JavaScript. The beauty of JSX is that it allows the developer to visualize the contents of pages, it is much easier to write on it than on traditional JavaScript [12].

From the point of view of the level of complexity, React is one of the easiest to learn. It is based on light programming languages - it is enough to remember how HTML works. There is no need to study TypeScript deeply, as in Angular, there is enough superficial knowledge. React became famous for its component-based architecture, which other platforms began to use much later. This structural approach allows you to create an interface relatively quickly and simply. It is worth noting that the library is aimed not only at creating a user interface for a single-page web application (Single Page Application, SPA) but also for mobile development.

React Native platform is a framework aimed at developing cross-platform high-quality applications for iOS and Android. The advantages of this tool are a variety of compatible modules (Browserify, RequireJS, ECMAScript 6, and others), installed components, unidirectional code flow, and Redux library [14].

Advantages:
- Lightweight of the database; processes occurring on the part of the user can be easily involved on the server;
- Data binding guarantees the immutability of parent data, child elements cannot change the data chain;
- Open data library is being improved daily thanks to open-source code;
- High level of flexibility and maximum responsiveness.

Disadvantages:
- Lack of ordered documentation, ultra-fast exchange of solutions in ReactJS leaves no room for ordering documentation, documents are placed a little chaotically;
- Too much choice of actions to solve certain problems - the number of these solutions can be confusing;
- Poor cross-browser support.

Vue

According to the developers, Vue is a progressive JavaScript framework for user interface development. A distinctive feature of Vue from big frameworks is a high degree of adaptability. It consists of orientation to the View level and ease of integration of other libraries or into existing projects [15].

Vue allows you to develop complex single-page applications (SPA) by extending HTML attributes with so-called directives. There are both built-in directives and those defined by the programmer.

Vue and the previously discussed React are very similar. The speed of operation of both frameworks is very high, however, there are some nuances of the functioning of Vue that are worth paying attention to.

Unlike React, in which changing the state of a component leads to redrawing the entire subtree of that component, Vue automatically tracks component dependencies. This approach allows the system to
know exactly which components need to be redrawn. This eliminates the need for a whole class of optimizations.

In React, everything is built on JavaScript, while Vue covers classic web technologies and is based on them.

React is difficult to learn because to use this framework, you need to have knowledge about JSX and ES2015+ and build systems. All this knowledge is not required to start development on Vue [16].

Compared to Angular, Vue has slightly higher performance [17].

As mentioned earlier, working with Angular requires knowledge of TypeScript. The use of TypeScript has its advantages, such as checking static types, but this leads to an increase in overhead costs. Vue provides official type declarations and an official decorator for those who want to use TypeScript with Vue.

The Angular API contains many elements and concepts, which take a lot of time to learn. Everything you need to know to work with Vue is HTML and standard JavaScript.

The size of a fully functional project with Vue (Vue Router + Vuex) is significantly smaller than with Angular (~30 kB vs. 65 kB).

Unlike Angular, Vue allows you to use different build systems without limiting developers on which structure to use for the application.

The development of the framework began in July 2013 and is still actively continuing.

**Advantages:**
- Enhanced HTML. This means that Vue.js has many characteristics similar to Angular. This can help optimize the processing of HTML blocks using different components.
- Detailed documentation. Vue.js has very good documentation that can increase the learning speed of developers and save a lot of time in developing an application using basic knowledge of HTML and JavaScript.
- Adaptability. Vue.js provides a quick transition period from other frameworks to Vue.js in view of its similarities with Angular and React in terms of design and architecture.
- Amazing. Vue integration.js can be used both for creating single-page applications and for more complex web application interfaces. The most important thing is that small interactive parts can be easily integrated into the existing infrastructure without adversely affecting the entire system.
- Large scaling. Vue.js helps to develop fairly large templates for multiple uses, which can be developed without spending a huge amount of time in view of the simple structure.
- Tiny size. Vue.js can weigh about 20 KB and at the same time maintain its own speed and flexibility, which allows you to achieve much higher performance compared to other frameworks.

**Disadvantages:**
- Lack of resources. Vue.js still has a fairly small market share compared to React or Angular. This means that knowledge exchange within the framework is still being formed.
- The risk of excessive flexibility. Sometimes Vue.js may have problems when integrating into huge projects, and there is still no experience with possible solutions. But they will definitely appear in the near future.
- Lack of complete English-language documentation. This leads to some difficulties at various stages of development. However, more and more materials are being translated into English.

**AI frameworks for efficiency**

There are several modern frameworks for web application development that play a crucial role in AI, particularly in building intelligent web applications.

Some of the popular frameworks include:

**React:** React is a JavaScript library for building user interfaces. It is widely used for building single-page applications, and its virtual DOM makes it very fast and efficient. React can be used in combination with other libraries to build AI-driven web applications.

**Angular:** Angular is a TypeScript-based framework for building web applications. It provides a robust set of tools and features for building complex applications. Angular can be used with TensorFlow.js, a library for building and training machine-learning models in the browser.

**Django:** Django is a high-level Python web framework that enables rapid development and clean, pragmatic design. It includes built-in tools for web development, such as an ORM, template engine, and routing system, making it easy to build AI-driven web applications.
Flask: Flask is a lightweight Python web framework that is easy to learn and use. It provides a simple way to create web applications and can be easily extended with Python libraries for machine learning, such as scikit-learn and TensorFlow.

Ruby on Rails: Ruby on Rails is a popular web application framework that follows the Model-View-Controller (MVC) architecture. It is known for its simplicity and elegance and can be used with machine learning libraries, such as Keras and PyTorch.

These frameworks provide developers with the tools they need to build intelligent web applications that incorporate machine learning and artificial intelligence algorithms. For example, TensorFlow.js can be used with React to build intelligent chatbots and recommendation systems, while scikit-learn can be used with Flask to build predictive analytics tools. By leveraging the power of these frameworks, developers can build intelligent web applications that provide a seamless user experience and deliver real value to businesses and consumers alike.

Also, frameworks in AI are essential tools that help developers build and train machine learning models more efficiently. AI frameworks provide a set of pre-built functions and libraries that make it easier for developers to build complex models without having to write all of the code from scratch [25].

Some popular AI frameworks include:

1. TensorFlow: Developed by Google, TensorFlow is one of the most widely used AI frameworks. It is an open-source library for dataflow programming across a range of tasks. TensorFlow is known for its flexibility and scalability, making it an excellent choice for large-scale machine-learning applications.
2. PyTorch: Developed by Facebook, PyTorch is another popular AI framework that is growing in popularity. PyTorch is known for its dynamic computational graph, which allows developers to modify their models on the fly, making it ideal for research and experimentation.
3. Keras: Keras is a high-level neural network API written in Python. It is designed to be user-friendly, modular, and extensible, making it an excellent choice for beginners and experienced developers alike.
4. Scikit-learn: Scikit-learn is a popular machine-learning library for Python. It provides simple and efficient tools for data mining and data analysis, making it an excellent choice for beginners and experts alike.

Using an AI framework can save developers a lot of time and effort, as they do not need to write all of the code from scratch. Moreover, the frameworks come with pre-built functions and libraries that can significantly improve the performance and accuracy of machine learning models.

Conclusion

Web development frameworks are similar in many ways, even if they are implemented in different programming languages. Nevertheless, each of the listed frameworks is individual. They have different approaches, methods, and behavior in development. It is impossible to choose the best framework among the available ones, it all depends on what you plan to work with [21-23]. Frameworks have a lot of different advantages and disadvantages. You can compare all the functions and technologies listed in this article to make it easier to understand which of the frameworks is best suited for business or the needs of a particular project, and make the right choice, observing in the end how the selected tools and libraries are of great help in the work. It can also be added that the further development of artificial intelligence functionality today is promising for the work and professional development of developers.

REFERENCES

В настоящее время технологии веб-программирования продолжает развиваться быстрыми темпами и важно знать о самых популярных и эффективных фреймворках, которые доступны сегодня. Фреймворки стали неотъемлемой частью процесса веб-разработки из-за растущих стандартов веб-приложений и сложности используемых технологий. При выборе правильного фреймворка есть много факторов, которые нужно учесть: требования к проекту, опыт команды и постоянно меняющиеся тенденции в индустрии. Фреймворк, как правило, содержит только базовые программные модули, а все специфичные для проекта компоненты уже реализуются на их основе разработчиком. Тем самым достигается не только высокая скорость разработки, но и большая производительность и надёжность решений. Если сравнивать фреймворки с другими классами платформ - SaaS, CMS или CMF - то фреймворки значительно эффективнее использовать в проектах со сложной бизнес-логикой и с высокими требованиями к скорости работы, надёжности и безопасности.

В статье рассматриваются типы популярных фреймворков, технологий (MVC) и инструментов, предназначенных для разработки веб-приложений с учетом современных стандартов. Приведены характеристики архитектуры и тип фреймворков. Рассматриваются популярные серверные и клиентские фреймворки и языки программирования, с которыми они работают. Отмечены преимущества и недостатки каждого из рассмотренных инструментов.
негізінде жузеге асырады. Осылайша, дамудың жоғары жылдамдығына ғана емес, сонымен қатар
шешімдердің ұлға өнімділігі мен сенімділігіне кол жеткізіледі. Егер біз фреймворктарды басқа
платформалық сыртқа алып, SaaS, CMS немесе CMF - пен салыстыратыңыз болсақ, онда
фреймворктарды күрделі бизнес логикасы бар және жұмыс жылдамдатына, сенімділікке және
қауіпсіздікке жоғары талаптары бар жобаларда колдану әлдекайда тиімді.
Макалада заманауи стандарттарды ескере отырып, веб-қосымшаларды жасауға арналған
tанымал құрылымдардың технологиялардың (MVC) және құралдардың түрлері карастырылады.
Архитектуралық сипаттамалары және жақтау түрі берілген. Танымал серверлік және клиенттік
құрылымдар мәндәр әлар жұмыс құжаттарына бағдарламалау тілдері карастырылады. Карастырылған
құралдардың еркейісісінің артықшылықтары мен кемшіліктері атап өтілді.