УДК 004.4’2

П. Б. Супрун

PECULIARITIES OF DEVELOPING WEB APPLICATIONS ON ASP.NET CORE

Vinnytsia National Technical University

Анотація

У статті досліджено актуальність створення веб застосунків. Проаналізовано особливість створення веб застосунків та використання можливостей платформи ASP.NET Core.

Ключові слова: веб застосунок, веб розробка, веб архітектура, .NET.

Abstract

The article examines the relevance of creating web applications. The peculiarity of creating web applications and using the capabilities of the ASP.NET Core platform is analyzed..

Keywords: web application, web developing, web architecture, .NET.

Intro

Users place high demands on modern web applications. Regardless of the screen size, such a web application should work stably on almost any device anywhere in the world. Web applications need to be secure, flexible and scalable to deal effectively with sudden load spikes. Full-featured user interfaces implement increasingly complex JavaScript-based scenarios and provide efficient interoperability through web APIs. The ASP.NET Core platform is optimized for modern web applications and cloud deployment scenarios. Due to their modular structure, programs use only those components that they really need, which improves their security and performance while reducing deployment resource requirements [1].

How web applications work

Web applications are a special type of software built on a client-server architecture. Their peculiarity lies in the fact that the web application itself is located and executed on the server - the client receives only the results of the work as shown on the picture 1. The operation of the application is based on receiving requests from the user (client), processing them and issuing a result. The transfer of requests and the results of their processing occurs via the Internet [2].

Picture 1 – Nowadays web application architecture

On the server side, the web application is executed by special software (web server), which receives client requests, processes them, generates a response in the form of a page described in HTML, and sends it to the client. One such web server is Microsoft's Internet Information Services (IIS), which is capable of running web applications built using ASP.NET technology.

In the process of processing a user request, a web application composes a response based on the execution of server-side program code. As a result, an HTML page is formed, which is sent to the client. It turns out that the result of the work of the web application is identical to the result of a request to a traditional web site. However,

in contrast to it, the web application generates HTML code depending on the user's request, and not just passes it to the client in the form in which this the code is stored in a file on the server side. That is, the web application dynamically generates a response using executable code, also called executable part [3].

Due to the presence of an executable part, web applications are able to perform almost the same operations as regular Windows applications, but the code is executed on the server, the browser acts as the system interface, and the environment to exchange data is the Internet [4].

The most common operations performed by web applications include:

- receiving data from the user and storing it on the server;
- performing various actions at the request of the user: extracting data from the database, adding, deleting, changing data in the database, performing complex calculations;
- user authentication and display of the system interface corresponding to the given user;
- display of constantly changing operational information, etc.

Purpose of .NET environment

ASP.NET is a platform for building web applications and web services that run on IIS. Moreover, the underlying programming languages that make it possible to develop web applications today are completely object-oriented, making development of an executable part, as well as its modification, maintenance, debugging, and reuse, much easier than in other technologies.

The history of ASP.NET actually began with the release of the first version of .NET in early 2002, and since then ASP.NET and .NET have evolved in parallel. However, 2014 marked a big change, in fact, a revolution in the development of the platform: Microsoft set a course for the development of ASP.NET as a cross-platform technology that is being developed as an open source project. This development of the platform was later called ASP.NET Core. The first version of the updated platform was released in June 2016. Now it began to work not only on Windows, but also on MacOS and Linux. It has become more lightweight, modular, it has become easier to configure, in general, it has become more in line with the requirements of the current time [5].

Developing models

ASP.NET Core allows you to build web applications using a variety of development models.

First of all, this is the basic ASP.NET Core, which supports all the main points necessary for the operation of a modern web application: routing, configuration, logging, the ability to work with various database systems, etc.

ASP.NET Core MVC represents in a general way building an application around three main components - Model (models), View (views) and Controller (controllers), where models are responsible for working with data, controllers represent the request processing logic, and views define the visual component.

Razor Pages introduces a model in which special entities, so-called Razor Pages, are responsible for handling the request. Each individual entity can be associated with a separate web page.

The ASP.NET Core Web API provides an implementation of the REST pattern where each type of http request (GET, POST, PUT, DELETE) is assigned a separate resource. Such resources are defined as Web API controller methods. This model is especially suitable for single-page applications.

Blazor introduces a framework that allows you to create interactive applications on both the server and client side and allows you to use low-level WebAssembly code at the browser level.

Platform key features:

- ASP.NET Core runs on top of the .NET platform and allows you to use all of its functionality;
- The programming languages supported by the .NET platform are used as development languages. C# and F# have officially built-in support for ASP.NET Core projects;
- ASP.NET Core is a cross-platform framework, which can be deployed on all major popular operating systems: Windows, Mac OS, Linux. So, using ASP.NET Core, we can both create cross-platform applications on Windows, Linux and Mac OS, and run them on these OSes;
- Thanks to the modularity of the framework, all the necessary components of a web application can be loaded as separate modules through the Nuget package manager;
- Support for most common database systems: MS SQL Server, MySQL, Postgres, MongoDB;

- ASP.NET Core is extensible. The framework is built from a set of relatively independent components. And we can either use the built-in implementation of these components, or extend them using the inheritance mechanism, or even create and use our own components with our own functionality;
- Rich toolkit for application development. As a development tool, we can use a development environment with rich functionality such as Visual Studio from Microsoft. In addition, the .NET CLI allows you to create and run ASP.NET projects in the console. That means that to write code you can use a conventional text editor, such as Visual Studio Code [6].

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Characteristics of Modern Web Applications [Електронний ресурс] // Microsoft. – 2021. – Режим доступу до ресурсу: https://docs.microsoft.com/en-us/dotnet/architecture/modern-web-apps-azure/modern-web-applications-characteristics.
2. Web Application [Електронний ресурс] // RingCentral. – 2021. – Режим доступу до ресурсу: https://www.ringcentral.co.uk/gb/en/blog/definitions/web-application/.
3. Web Application Architecture in 2021: Moving in the Right Direction [Електронний ресурс] // MobiDev. – 2021. – Режим доступу до ресурсу: https://mobidev.biz/blog/web-application-architecture-types.
4. Shklar L. Web Application Architecture: Principles, Protocols and Practices / L. Shklar, R. Rosen., 2009. – 440 с.
5. Enterprise Application Development with C# 9 and .NET 5 / Ravindra Akella, Arun Kumar Tamirisa, Suneel Kumar Kunani, Bhupesh Guptha Muthiyalu., 2021. – 610 с.
6. Metzgar D. .NET Core in Action / Dustin Metzgar., 2018. – 288 с.

Супрун Павло Борисович *— студент групи 1КІ-21м, факультет інформаційних технологій та комп'ютерної інженерії, Вінницький національний технічний університет, Вінниця, e-mail: suprunS123@gmail.com*

Науковий керівник: ***Мельник Олеся Дмитрівна*** *— кандидат філологічних наук, доцент кафедри іноземних мов, Вінницький національний технічний університет, Вінниця.*

Suprun Pavlo Borysovych *— student of the group 1CE-21m, Department of Information Technologies and Computer Engineering, Vinnytsia National Technical University, Vinnytsia, e-mail: suprunS123@gmail.com*

Supervisor: ***Melnyk Olesya Dmytrivna*** *— candidate of Philological Sciences, Associate Professor at the Department of Foreign Languages, Vinnytsia National Technical University, Vinnytsia.*