UDC 004. 7:378 **doi: 10.26906/SUNZ.2023.2.101**

M. Demydenko, M. Kalnytskyi

National University "Yuri Kondratyuk Poltava Polytechnic", Poltava, Ukraine

ONLINE PLATFORM PROTOTYPE USING MICROSERVICE ARCHITECTURE AND CONTAINERIZATION FOR DIGITALIZATION OF THE EDUCATIONAL PROCESS

Abstract. The COVID pandemic and full-scale military operations in our territory affected the implementation of digital and information technologies in the educational process. Digitalization has covered all spheres of education, and has especially affected distance learning technologies. But the problem of automating the process of managing the educational process remains unexplained. Ensuring quality management of the educational process requires a review of approaches to architecture, development processes, and implementation of appropriate software. Most of the modern information systems for the management of the educational process (ISMEP) are built according to the "monolithic" architecture. This limits the possibility of modernization without rebuilding the core of the system. The consequence of this is the problem of scaling and efficient use of cloud technologies. The purpose of the work is to develop a prototype of an online ISMEP platform using microservice architecture and containerization.

Keywords: digitalization of the educational process, microservice architecture, containerization.

Introduction

The COVID pandemic and full-scale military operations in our territory affected the implementation of digital and information technologies in the educational process. Digitalization has covered all spheres of education, and has especially affected distance learning technologies. But the problem of automating the process of managing the educational process remains unexplained.

Actuality of theme. Ensuring quality management of the educational process requires a review of approaches to architecture, development processes, and implementation of appropriate software. Most of the modern information systems for the management of the educational process (ISMEP) are built according to the "monolithic" architecture:

1. MKR [1].
2. Politek-soft [2].
3. ASU VNZ [3].

The listed *ISMEP*s are built on client/server technology and consist of several modules that automate the activities of the main structural divisions. This limits the possibility of modernization without rebuilding the core of the system. The consequence of this is the problem of scaling and efficient use of cloud technologies.

Complex solutions of educational tools from Google and Microsoft are mostly focused on e-learning. That is, there are no ISMEP on the market that are built on a microservice architecture.

Analysis of the latest research. Most of the publications devoted to microservice architecture in the educational process consider its use in e-learning. Thus, in the work [4], the main principles of the organization of microservice architectures (MSA), their parameters, functioning, and the possibility of application in electronic educational systems (EES) were considered. A comparative analysis of construction options for the MSA, their components and methods was carried out. A comprehensive method of building electronic educational systems based on MSA is proposed. In work [5], a study was conducted on increasing the productivity of EES when switching to MSA.

The objective of the work – to develop a prototype of an online ISMEP platform using microservice architecture and containerization.

Main part

Features of application of monolithic and MAS architecture in ISMEP. The parameters of ISMEP are determined not only by functionality and software, but also by architecture. Most ISMEPs with a developed core use monolithic structure. This gives them such advantages as good dynamics of development and support. An alternative to monolithic systems is the division of ISMEP into separate microservices. MSA provides the following advantages:

1. Modularity. Separate parts of the application are responsible for specific operations.
2. Reliability. When one of the microservices fails, only those functions for which it is responsible are disrupted. It will not be able to damage the server completely.
3. Changeability. You can change the microservice independently of other system services.
4. Scalability. You can add or remove a module at any time. If the microservice reaches its maximum load, new instances of the service can be deployed in a neighboring cluster.
5. Testability. New features can be tested immediately in production. It is enough to implement them for some segments of users to check how they will work [5].

Disadvantages using microservice architecture:

1. Organization of interaction of microservices in the entire system. This depends on the number of processes involved and requires additional efforts in planning, control, monitoring, testing and deployment.
2. Expenses. Microservices require the use of a larger total amount of memory than monolithic architectures, this is due to the need to organize isolated environments, as well as the costs of organizing communications between them (interfaces, protocols, communication channels, etc.).

3. Security issues. Microservices require security solutions due to the presence of internal service communication.

4. Decrease in productivity compared to monolithic systems. The reason is the occurrence of additional delays in data exchange between microservices. Therefore, it is necessary to solve problems with the correct design of queues, asynchronous processing and balancing.

5. Problems of inconsistency. Unlike monolithic systems, in MSA, errors when updating individual microservices can lead to out of sync violations [4].

For the realization of ISMEP, there are chosen the following functions (Fig. 1):

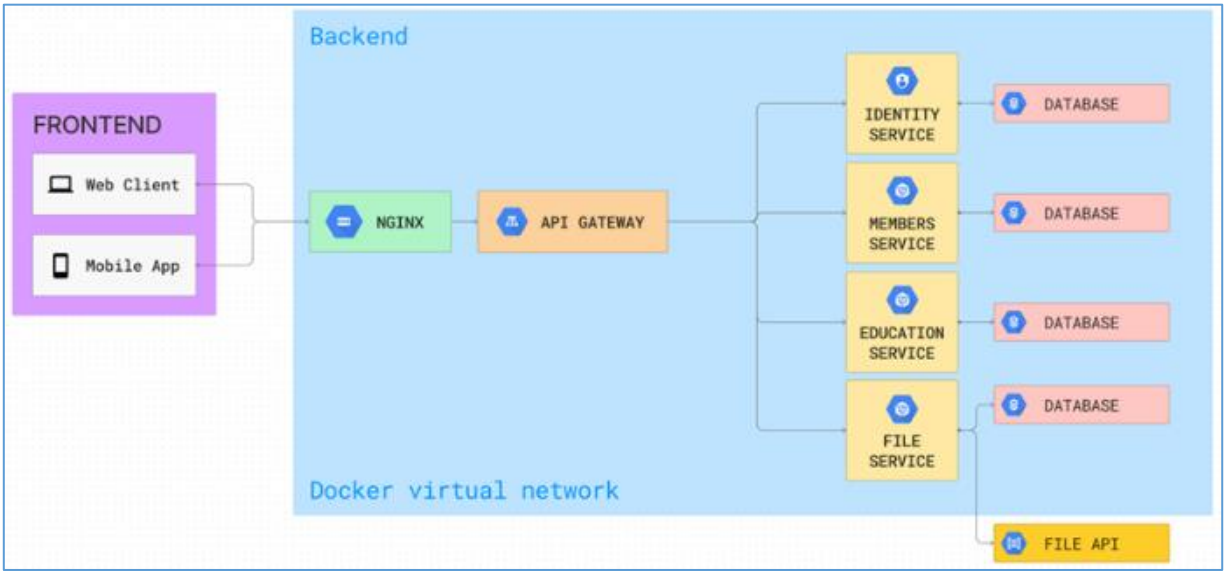

Fig. 1. ISMEP prototype architecture

The implementation of the platform prototype was divided into two parts - the creation of the server (backend) and client (frontend) parts. The server part is a REST API system built on a microservice architecture and deployed using containerization technologies, the task of which is to receive HTTP requests from the client (web application), process them and return data in JSON format.

1. API Gateway is a request distributor and aggregator, provides only permitted functionality for users with different access levels, and collects data from multiple microservices as needed. The only point of interaction between the client and the server is the API GATEWAY. This microservice does not contain any database, and performs two functions: a role limiter and a request aggregator.

The first role of Api Gateway is to provide a certain function only to authenticated users with the required role (student/teacher/administrator). Authorization starts through guest access points that are directed to the identity microservice.

The second role of Api Gateway is aggregation. Some requests require data and processing in multiple microservices, so gateway's job is to send requests to all required microservices, collect all responses separately, filter, and return the response to the client. Specific microservices are closed from the user, so there is no role checking. One big problem in the aggregation process is the lack of atomicity of operations. Ideally, when sending requests to 2-3 microservices, if one of them does not perform the operation and gives an error, then all changes in other microservices should be canceled.

Identity Service is created for authorization, registration and storing of personal data of users. Application authorization and authentication is built on a modern approach that uses JWT tokens.

JWT (JSON Web Token) is a way of securely transferring information between a client and a server in the form of a special string token. The token contains information required for authorization and authentication (such as an ID and role) and can be used to access certain resources or functions in the application. The JWT is transmitted with each request, accordingly, the validity of the token and the possibility of access of the given user to the requested sections of the functionality at the moment will be checked each time.

When entering the correct login and password on the site, the Identity microservice generates a token that stores the unique user ID and role (student/teacher/administrator) and returns it to the client for storage. The token itself is signed with a secret key, making counterfeiting by a third party impossible.

The lifetime of the token, which determines its validity, is recorded in the body of the token and is determined by the server that issues it. The recommended time is up to 30 minutes.

In order for the user not to re-enter his login and password every ten minutes to log in to the account, a second key, called Refresh Token, is issued during authorization. This is a special code that can be used by the client to update the JWT without passing the login and password.

The Refresh Token is stored in the browser's Http-only cookie, which guarantees the impossibility of receiving the token by third-party scripts and programs.

The single access point to the server functionality (API Gateway) contains four routes through which requests can pass. Each of the three roles (student/teacher/administrator) has its own route, which checks the existence of the required role in the transferred JWT token before executing the commands. The fourth route is made for non-authorized users, it does not require any tokens and is used for user registration or login.

The microservice data scheme is shown in Fig. 2.

2. Members Service is designed to store and process data about students, teachers, groups and reviews. The data scheme is shown in Fig. 3.

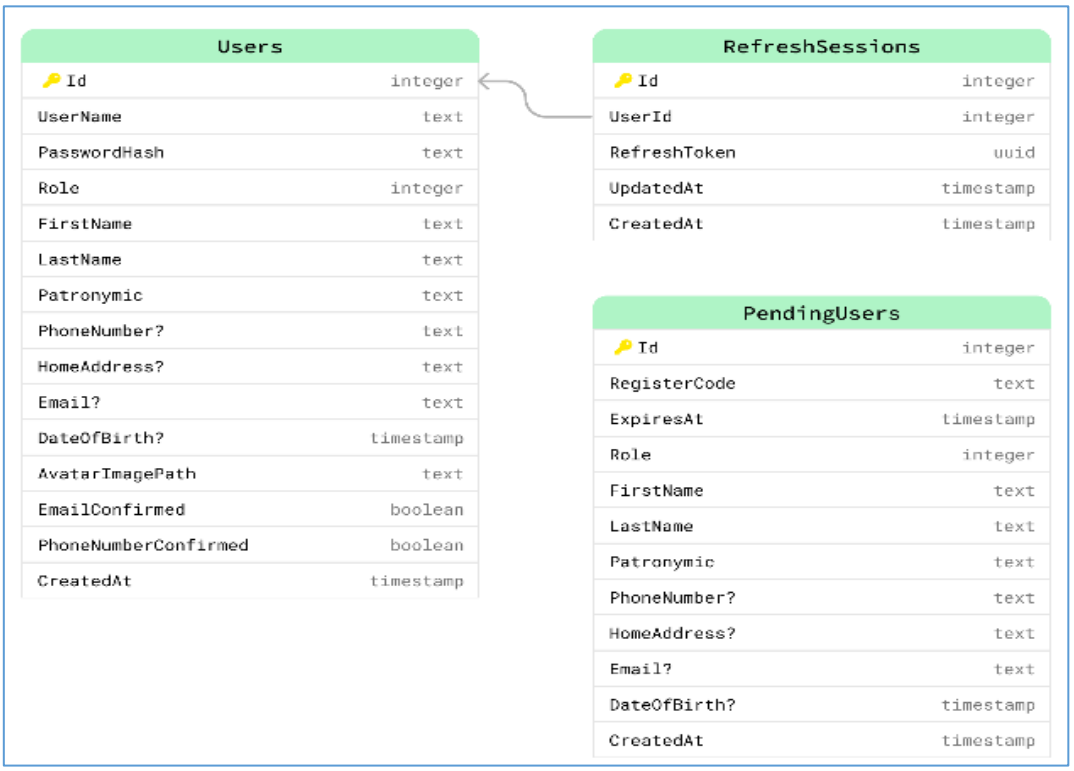

Fig. 2. Identity microservice data scheme

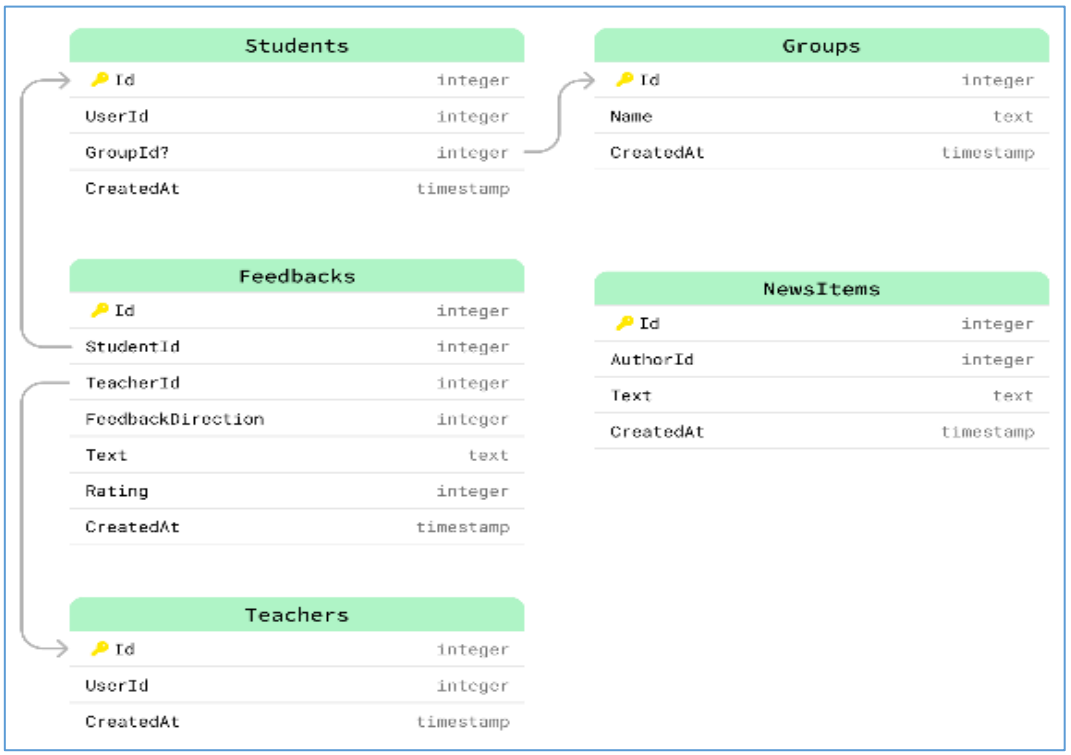

Fig. 3. Members microservice data scheme

3. Education service is designed to store and process everything related to the educational process: assignments, schedules, grades, assessments, exams. The data scheme is shown in Fig. 4.

4. File Service is designed to work with external applications for working with files and images. Amazon S3 is used to store and download files. The data scheme is shown in Fig. 5.

Core functions of the prototype of the online platform ISMEP.

There are three user groups defined:

- student;
- teacher;
- administrator.

Diagram of the possible actions is shown in Fig. 6. To ensure containerization, Docker containers are used, which are a kind of virtual machines that allow developers to package and run programs in an isolated environment.

They contain everything the application needs to run, including dependencies and settings.

Conclusions

The analysis of recent researches showed that the majority of publications devoted to microservice architecture in the educational process consider its use in e-learning.

Most of the modern information systems for the management of the educational process (ISMEP) are built according to the "monolithic" architecture. Based on the results of the research, a prototype of the online platform was developed.

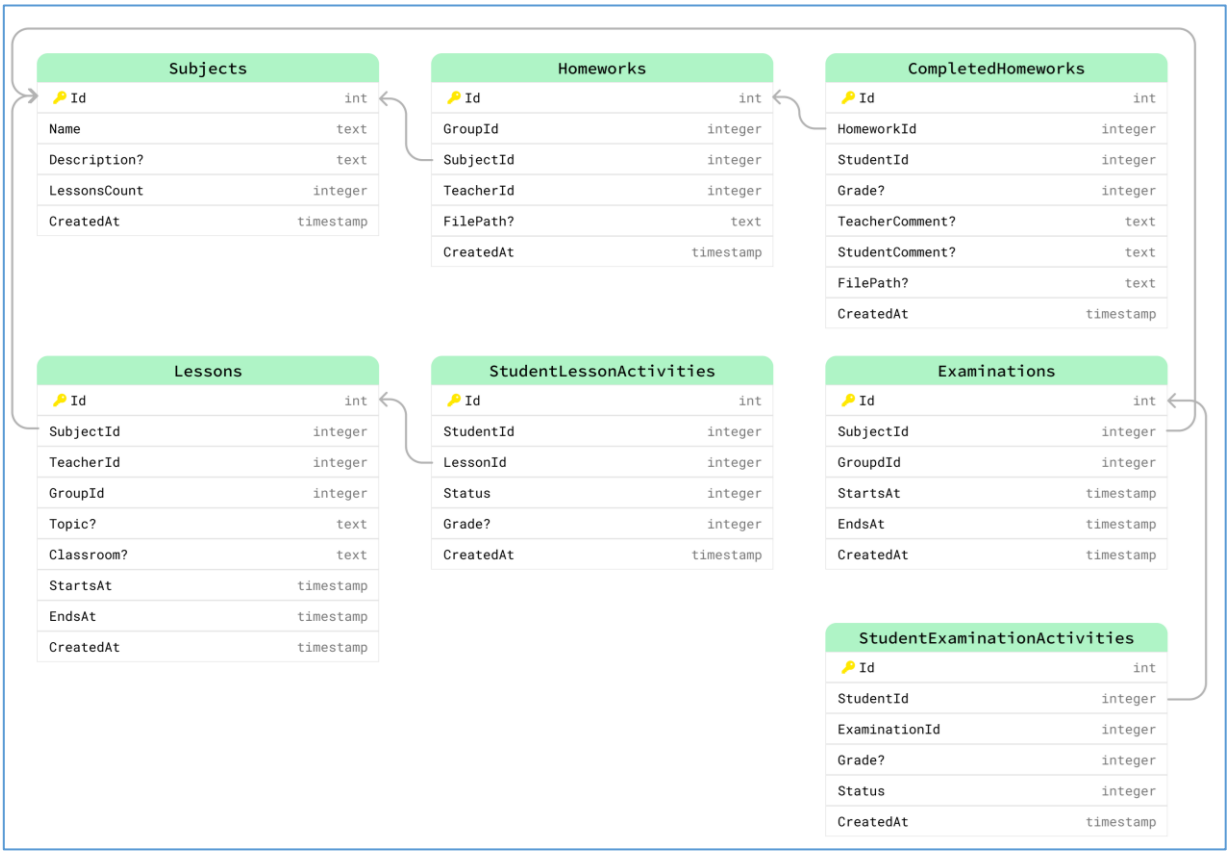

Fig. 4. Education microservice data scheme

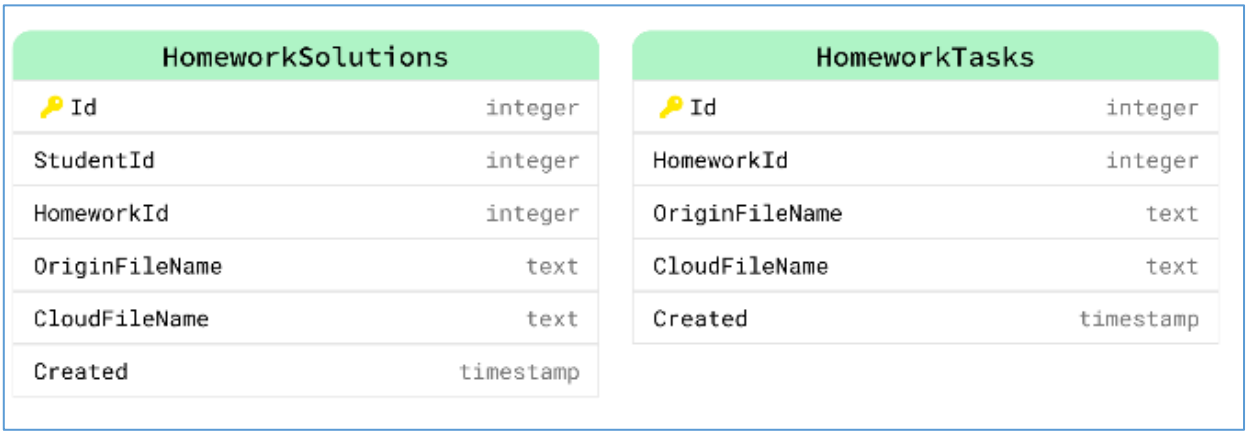

Fig. 5. File microservice data scheme

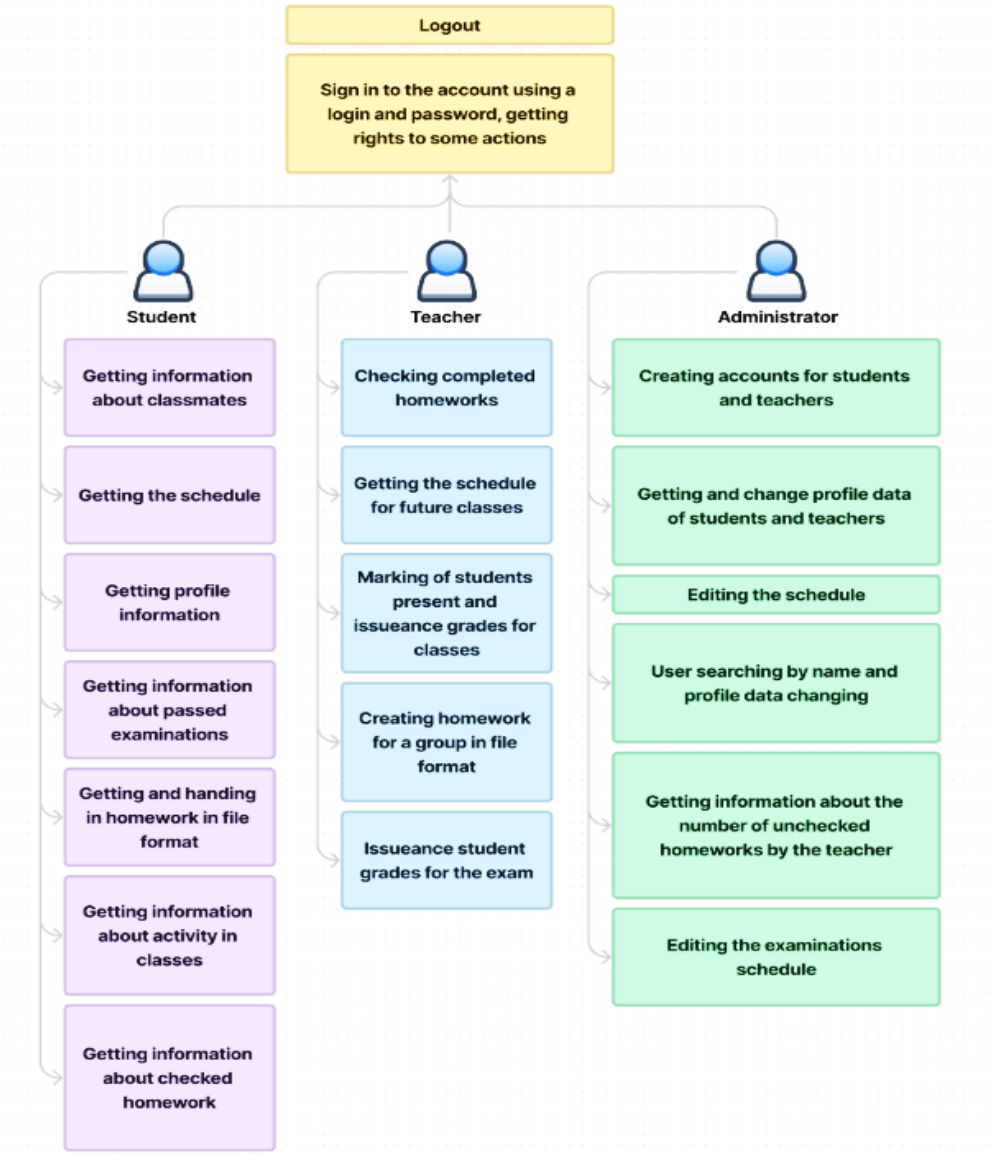

Fig. 6. ISMEP core functions

REFERENCES

1. MKR [Electronic resource] –Access mode:https://mkr.org.ua/
2. Politek-soft [Electronic resource] –Access mode: http://www.politek-soft.kiev.ua/ru/
3. ASU VNZ [Electronic resource] –Access mode: http://ndipit.com.ua/ua/rozrobky/asu-vnz
4. Є. Паламарчук, «МЕТОДИ ПОБУДОВИ МІКРОСЕРВІСНОЇ АРХІТЕКТУРИ СИСТЕМ ЕЛЕКТРОННОГО НАВЧАННЯ», ІТКІ, вип. 53, вип. 1, с. 43–54, Лют 2022. https://doi.org/10.31649/1999-9941-2022-53-1-43-54
5. A. Kravchuk Мікросервісна архітектура: плюси та мінуси. Access mode https://blog.iteducenter.ua/articles/microservices-architecture-advantages-and-disadvantages/

Надійшла (received) 22.03.2023
Прийнято до друку (accepted for publication) 23.05.2023

Прототип онлайн платформи з використанням мікросервісної архітектури та контейнеризації для діджиталізації освітнього процесу

М. І. Демиденко, М. А. Кальницький

Анотація. Пандемія COVID, та повномасштабні військові дії на нашій території вплинули на впровадження цифрових та інформаційних технологій в освітній процес. Діджиталізація охопила всі сфери освіти , а особливо вплинула на технології дистанційного навчання. Але проблема автоматизації процесу управління освітнім процесом залишається не висвітленою. Забезпечення якісного менеджменту освітнього процесу вимагає перегляду підходів до архітектури, процесів розроблення, та впровадження відповідного програмного забезпечення. Більшість сучасних інформаційних систем управління освітнім процесом (ІСУОП) побудовані по «монолітній» архітектурі. Це обмежує можливість модернізації без перебудови ядра системи. Наслідком цього є проблема масштабування та ефективного використання хмарних технологій. Мета роботи – розробити прототип онлайн платформи ІСУОС з використанням мікросервісної архітектури та контейнеризації.

Ключові слова: діджиталізація освітній процес, мікросервісна архітектура, контейнеризація.