УДК 621.397:004.891

Андреєв О.В., к.т.н., доцент
Гераймович О.М., ст. викл.
Шевченко М.В., студент, гр. ТР-21-1
Державний університет «Житомирська політехніка»
Дубина О.Ф., к.т.н., доцент
Житомирський військовий інститут ім. С.П. Корольова

АЛГОРИТМ ОБРОБКИ ЗОБРАЖЕНЬ ВІДЕОКАМЕРИ OV2640 ДЛЯ ДЕТЕКТУВАННЯ РУХУ

На сьогоднішній день велика кількість досліджень присвячена комп'ютерному зору, який останнім часом отримує все більше застосування. Реалізація функції детектування руху при відеоспостереженні здійснюється апаратним чи програмним способом. Апаратний спосіб має на увазі застосування спеціального датчика, найчастіше інфрачервоного, вбудованого в корпус відеокамери. При цьому рух виявляється за рахунок перетину об'єктом, що переміщується, сегментів, що становлять зону виявлення датчика. При цьому практично неможливо домогтися повного співпадіння областей контролю датчика руху і камери, що вельми критично при встановленні камери на вулиці. При програмному варіанті відеокамера виявляє зміну зображення, що формується безпосередньо на її матриці. Такими можливостями володіють IP-камери відеонагляду. Метою виявлення руху є розпізнавання зміни позиції предмету на двох послідовних зображеннях. Як метод порівняння зображень на практиці найчастіше використовується нормалізована кореляція двох послідовних зображень. Даний метод має ряд переваг, що полягають у його малій чутливості до зміни яскравості зображення, можливості швидкого обчислення та простоті реалізації. Для порівняння двох зображень здійснюється їхнє розбиття на блоки, між відповідними парами яких обчислюється нормалізована кореляція C_n. Якщо отримане значення нормалізованої кореляції C_n менше деякого порогу T, то вважається, що блоки мають різні текстури і зображення у відповідних точках відрізняються. На рисунку 1 світлими точками виділені блоки результату порівняння, де зображення відрізняються, тобто $C_n<T$, а темні області відповідають умові схожості зображення у блоках. Для реалізації даного алгоритму у модулі ESP32-CAM із камерою OV2640 було запропоновано використовувати стиснене зображення у форматі PIXFORMAT_GRAYSCALE_JPEG. Цей формат являє 8-бітне зображення відтінків сірого та ідеально підходить для детектування руху при комп'ютерному зорі. Для розв'язання завдання виявлення руху рекомендована оптимальна для обробки зображення роздільна здатність камери 320x240. При цьому буфер зображення буде найменшим у порівнянні з іншими форматами, що дозволить зменшити навантаження під час математичних розрахунків. Виходячи з обраної роздільної здатності зображення, була побудована матриця детектування розміром 16x16 блоків. При цьому розмір кожного блоку склав 20x15 пікселів. Реалізація алгоритму нормалізованої кореляції показала, що для кожного блоку матриці детектування, при відсутності руху, середнє значення C_n коливається в діапазоні 0,97-0,99. Місцями присутні поодинокі значення в діапазоні 0,75-0,9, що пояснюється шумом зображення. При наявності руху значення нормалізованої кореляції Cn варіюється в діапазоні 0,1-0,5. Отже для коректного детектування слід обирати поріг $T < 0,7$, а для виключення можливості хибного спрацювання доцільно активувати сигнал тривоги тільки при виявленні руху на двох поспіль знімках. Під час програмної реалізації алгоритму на базі ESP32-CAM математичні операцій із числами з плаваючою комою виконуються повільніше, ніж операції із цілими числами. Середній час ітерації порівняння двох кадрів склав приблизно 400мс, при використанні типу даних «double» та 160мс - при використанні типу даних «float».

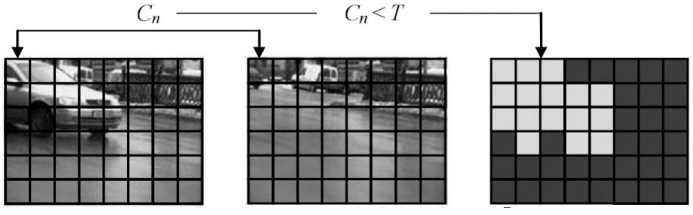

Рис. 1. Результат порівняння блоків двох сусідніх кадрів

З урахуванням того, що інформація про яскравість кожного пікселю подається цілочисельними значеннями фіксованого розміру від 0 до 255, було запропоновано спростити алгоритм детектування нормалізованої кореляції шляхом порівняння середньої яскравості пікселів окремих блоків. Завдяки цьому вдалося суттєво підвищити оперативність рішення задачі виявлення руху. Практично доведена можливість проведення обробки до 90 кадрів за секунду, що у 15 разів швидше порівняно із типовим алгоритмом нормалізованої кореляції при використанні типу даних «float».

УДК 004

Мельничук Є., студент
Граф М., Ph.D., доцент
Державний університет «Житомирська політехніка»

ВИКОРИСТАННЯ ШТУЧНОГО ІНТЕЛЕКТУ ПРИ СТВОРЕННІ СКРИПТІВ ДЛЯ ІГОР

Створення інді-ігор - це складний процес розробки відоігор невеликою командою, або однією людиною, часто без фінансової підтримки. Основними етапами розробки є препродакшн, продакшн (виробництво), пост-продакшн. Пост-продакшн включає визначення жанру, цільової аудиторії, платформи, створення концепт-документа і дизайн-документ, створення концепт-артів, вибір стилістики. Продакш - реалізація механік, створення моделей та спрайтів, саунд-дизайн, анімації, левел-дизайн та впровадження користувацького інтерфейсу [1]. У пост-продакшн входить альфа та бета тестування, відкриті тестування та випуск.

Метою є провести аналіз можливостей штучного інтелекту при розробці інді-ігор на прикладі додавання скрипту миготіння світла до існуючого ігрового проєкту.

Розглянемо використання штучного інтелекту (ШІ) ChatGPT [2] для генерації програмного скрипту миготіння світла.

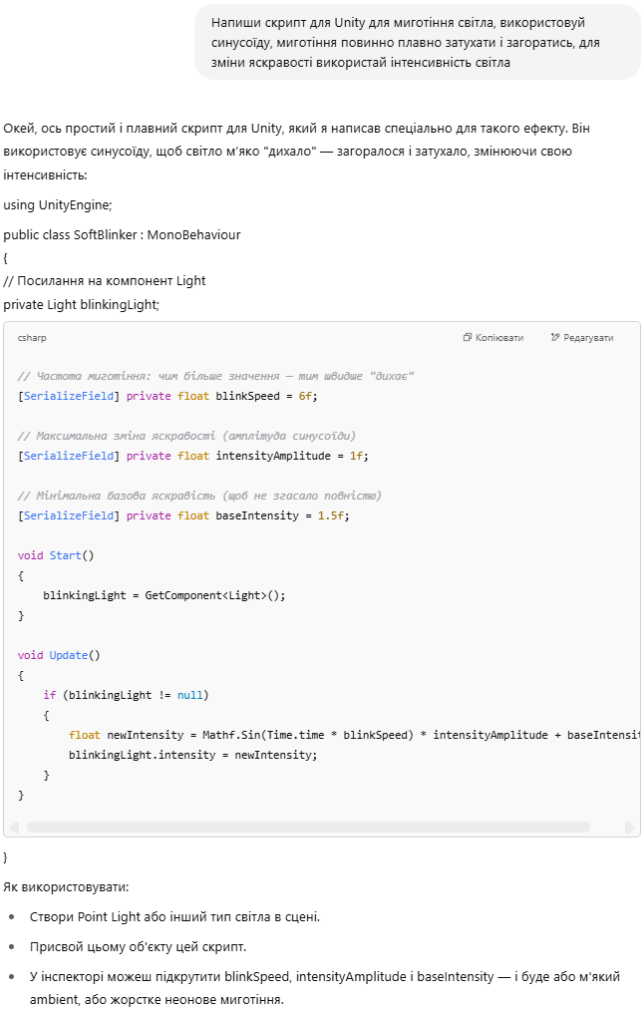

Рис. 1. Діалог з ШІ ChatGPT

Запропонований код було протестовано в рушії Unity, в ігровому 3D-проєкті “The Abyssal Truth” [3], що було представлено на GlobalGameJam-2025. Результат виконання показано на рисунку 2.

Рис. 2. Результат додавання скрипту світла, запропонованого ШІ

Як видно з рисунка 2 доданий скрипт коректно відображує світло. Також даний скрипт є зручним для редагування, адже всі змінні винесено окремо. Пропонується удосконалити запропоновану ШІ реалізацію, спростити за рахунок видалення зайвих змінних. Скрипт наводиться нижче.

```
using UnityEngine;
public class example : MonoBehaviour
{    Light myLight;
     void Start()
    {
       myLight = GetComponent<Light>();
    }
    void Update()
    {
       myLight.intensity = Mathf.Sin(Time.time*10)+1.5f;
    }
}
```

Запропонована реалізація дозволить зробити миготіння для всіх об’єктів сцени однаковим, що повністю відповідає поставленій задачі.

Отже штучний інтелект є дуже зручним інструментом для розробників, він дозволяє пришвидшити розробку, знизити вартість. Наявність ШІ в сучасній розробці є зручним інструментом для створення ігор. Протете, ШІ - це лише інструмент, він не має відчуття гармонії, атмосфери та загального бачення, що є надважливим при створенні ігор. Крім того для отримання задовільного результату часто потрібна велика кількість вхідних даних, що ШІ не завжди може корктно опрацювувати. Тому, елементи створені ШІ не завжди відповідають очікуванням, бувають неоптимальними та складними.

Список використаних джерел:

1. Бреславець В.С. Технології розробки комп’ютерних ігор: довідник модуля. / В.С. Бреславець. URL: https://repository.kpi.kharkov.ua/server/api/core/bitstreams/35b93624-72ac-405e-a5d1-7e4d1618afb8/content (дата звернення: 07.05.2025)

2. ChatGPT. Модель штучного інтелекту / OpenAI. URL: https://chat.openai.com/ (дата звернення: 07.05.2025).

3. Abyssal Truth [Електронний ресурс] // Global Game Jam. URL: https://globalgamejam.org/games/2025/abyssal-truth-7 (дата звернення: 07.05.2025).

UDC 004:004.5

Yanchuk V., Ph.D., Assoc. Prof.
Zhytomyr Polytechnic State University

PRACTICAL IMPLICATIONS OF SECURITY MEASURES AND DATA SECURITY IN E-COMMERCE SOLUTIONS

Ongoing invasion in Ukraine created a lot of complications for exchange visits and to economies of many countries, and at the same time, it created an opportunity to investigate the processes ongoing in the crisis period from inside Ukraine.

E-commerce security is not just a technical concern but a business imperative that directly impacts customer trust and business continuity.

Recent studies show that over 60% of consumers would stop doing business with a company after a security breach [1]. This work covers practical approaches to implementing security measures in e-commerce environments, from regulatory compliance to technical safeguards. Our goal is to provide actionable insights that can be implemented regardless of your organization's size or technical maturity.

Market share dimensions make E-Commerce a very attractive investment for many stakeholders. It is outstanding to realize that the same industry that exists in physical stores may develop into a separate market that can operate even without a warehouse per se. The new age of Dropship deliveries made this possible.

Recurring users prefer to keep their payment method remembered, so they do not need to enter the credit card data again, thus payment is even easier than ever. Products from international vendors are becoming more accessible. E-commerce is a profitable market tool that opens a lot of opportunities.

There are even more options for how you can boost your customers' activity on the web shop. The market trends of 2022 bring more tools for Machine Learning and Artificial Intelligence and serve to ecommerce applications [2]. However, most of these activities are possible only when you have identified your customers, their preferences, you have their data and can map them to your sales plan strategy.

The global e-commerce market reached $5.9 trillion in 2024, creating an increasingly attractive target for cybercriminals. As you see from the projection, by 2040 the 95% of purchases will be done via e-commerce and it will bring additional annual growth, gradually adding 14.7% annual increase in sales [3].

But why we still see so many abandoned carts on the web-shops, not converting them into an order? It brings us to an analysis of the loyalty programs and possible discount solutions to boost the sales, introduce the Free Shipping program etc.

Constant increase in value for E-Commerce becomes attractive to cybercriminals.

Major breaches in 2023-2024 have shown sophisticated attack methods targeting payment systems, customer databases, and supply chain vulnerabilities. The average cost of a data breach in the retail sector now exceeds $3.2 million, with significant additional costs in lost customer trust and regulatory penalties [4]. E-commerce platforms face unique challenges due to their exposure to the public internet, complex integrations, and the high value of the data they process:

- Global e-commerce market size and growth
- Recent major security incidents (2023-2024)
- Common attack vectors in e-commerce
- Average cost of data breaches in the retail sector

Storage of your customer's data is not a heavy burden; this is a huge responsibility that goes on par with the customers' trust. Many reviewers in the network always explain that many of these threats are less probable than others and thus create false security, which may break at any second. The better approach is to consider how to build your solution and embed the procedures and approaches into it. A comprehensive e-commerce security strategy must address multiple layers, starting with network security to protect the infrastructure and application security to ensure software integrity.

Database security is critical for protecting customer information, while transaction security safeguards payment processes from interception or manipulation.

Physical security protects hardware and infrastructure, but ultimately, human factors remain one of the most significant vulnerabilities in any security program. Each component requires specific controls and continuous monitoring to create defense-in-depth by means of:

- Network security
- Application security
- Database security
- Transaction security
- Physical security measures
- Human factors

Being in Europe you are aware of GDPR, which has set a global standard for data protection with significant penalties of up to 4% of annual global turnover, while the CCPA gives California residents specific rights regarding their personal information. E-commerce operations must implement mechanisms to handle different requirements across regions, potentially requiring geo-specific data handling processes [5]. In the case of Enterprise with a turnover of 10 billion EURO the penalty may be comparable to the budget for an average department. A privacy impact assessment should be conducted to understand which regulations apply to your specific operations.

The financial and operational anxieties of maintaining data compliance in an ever-evolving global landscape can keep even the most seasoned revenue and legal leaders up at night.

Balancing the costs of compliance with the risks of non-compliance demands not only a cation, but a proactive approach to understanding and implementing robust data privacy measures. As you see, starting an electronic business, it is essential to be aware of all the variations in regulation as well as data transformations according to international regulations depending on the country you work for or send to.

Global e-commerce operations face a multi-dimensional security challenge where regulatory requirements create a complex matrix of sometimes contradictory obligations—from GDPR's right to erasure conflicting with financial record retention laws to the varying definitions of personal data across jurisdictions:

- Regional regulatory variations
- Multi-jurisdiction compliance
- Localization security challenges
- International payment security
- CDN and edge security
- Data sovereignty requirements
- Cross-border incident response

Your international security architecture must incorporate regional infrastructure with data localization capabilities, jurisdiction-specific encryption standards, localized payment security controls, and a globally coordinated but locally executed incident response capability while maintaining consistent customer experience and operational efficiency that enables rather than impedes your business expansion strategy:

- Data minimization principles
- Privacy by design approach
- Consent management
- Data subject rights handling
- Cross-border data transfer considerations

Diving deeper, we find, that Merchant levels determine validation requirements, with Level 1 merchants (processing over 6 million transactions annually) facing the most rigorous assessment procedures. Common compliance challenges include maintaining proper network segmentation, managing vendor access, and ensuring security of legacy systems. Non-compliance can result in fines ranging from $5,000 to $100,000 per month, increased transaction fees, and potential suspension of card processing privileges.

- Merchant levels and validation requirements
- Compliance maintenance challenges
- Common audit findings
- Cost implications of compliance

Regular security assessments should be conducted at least quarterly to identify new vulnerabilities as the threat landscape evolves. Vulnerability scanning provides automated identification of common security issues, while penetration testing involves human expertise to discover complex vulnerabilities through simulated attacks.

An Information Security Management System (ISMS) provides a systematic approach to managing sensitive information through risk assessment, security design, and continuous improvement. The risk assessment methodology identifies threats, vulnerabilities, and potential impacts to determine appropriate security controls. The Statement of Applicability documents which controls from the ISO 27001 Annex A will be implemented based on risk assessment results.

References:

1. Smith, J. et al. (2024). "Evolution of E-Commerce Security Threats in the Post-Pandemic Era." Journal of Cybersecurity Research, 15(2), 45-63.
2. Chen, H. & Patel, R. (2023). "Implementing Zero Trust Architecture in E-Commerce Environments." International Journal of Information Security, 22(3), 289-305.
3. Williams, T. (2024). "The Economics of E-Commerce Security: Cost-Benefit Analysis of Security Controls." Journal of Information System Security, 18(1), 78-92.
4. Forrester Research. (2024). The State of Application Security.
5. Johnson, M. (2023). "E-Commerce Security: Practical Approaches for Business Leaders.“

UDC 004

Verbovskyi O.Yu., Master's student, group IPZm-24-1, FICT
Loktikova T.M., Senior Lecturer at the Department of Software Engineering, FICT
Zhytomyr Polytechnic State University

USING DATA PREDICTION METHODS AND ALGORITHMS TO IMPROVE THE INTERACTIVE PLATFORM USER EXPERIENCE

Humanity has become very busy in the era of high-speed Internet, real-time communication capabilities, and the ability to solve problems of various formats instantly. As a result, people are very annoyed when they waste time on something that does not bring the expected result. Many people do not want to use various online services that could significantly simplify their lives because they are not sure that the service will be received without problems and will not have to spend personal time on it. A clear example is the refund mechanism – users do not always feel sufficiently protected because, in a critical situation, they will spend time and nerves communicating with the opponent and platform administrators, which still does not guarantee to receive compensation at the level they expected.

There are already solutions to improve the user experience in such cases – for example, a rating system that allows users to rate each other, based on which other users can compare previous ratings and determine what suits them better. However, this solution is not ideal because, for analysis, it is necessary to go to the user profiles every time, process them, and make decisions, still feeling uncertain whether everything was correctly defined and perceived during these actions.

In the modern world, data prediction algorithms, neural networks, and other technologies make it possible to forecast the success or failure of various operations. This is very useful because people trust computers much more than people. For good reason: a person, regardless of his characteristics, can make a mistake at any time, but a computer cannot. It performs the calculations given to it in the same version as always. So, nothing new and unpredictable almost never happens.

As mentioned earlier, today, there are various data forecasting methods. Some are designed for dependencies between numerical values, others for classification, and some can analyze text and contain patterns. Thus, there are many approaches and tools for implementing one prediction feature, so at this stage, there is a solution and an answer to the question - which method is best suited for predicting the probability of successful task completion?

For example, the linear regression method is most suitable for predicting a numerical outcome based on numerical input data. Such a decision would have come to pass if the trading platforms were formed based solely on the price and the purchase result. However, such an implementation is extremely low-versus, and today's platforms implement many useful functions, such as chat functionality between service providers and clients, rating system, activity history, and more. Therefore, it is advisable to use a different, more flexible approach for such analyses.

This could be the nearest neighbor method. This approach is based on the assumption that similar situations have similar results. That is, if a user behaves similarly to others who have completed a similar task, then with a high probability, he will also complete the order. However, despite the simplicity of implementation, the nearest neighbor method has significant drawbacks, particularly low scalability, and sensitivity to "noise" in the data.

Therefore, the decision tree algorithm is proposed. This is one of the most common algorithms for classification and regression. A general tree is built to implement the model, where steps are formed depending on the data based on a particular feature [1]. The leaves in these trees are the final predictions. For each new case, a new branch is built along with the leaf, which makes the method quite flexible and able to work with new cases in the future. Such a model is easy to test because it is enough to get an image of the tree to view the progress of the prediction.

However, such trees are also prone to overfitting. The tree can become too deep and will adapt to the training data even if a small change causes a large part to be reshaped. This can lead to erroneous answers later on.

However, it is worth noting that this algorithm can be improved, so it was applied in the project under development. In the ensemble method of the random forest, many decision trees are combined. This leads to a more stable and accurate forecast since each tree is created based on randomly selected subsets of the data. The final result can be obtained not only in the form of an answer of 0 or 1 but also in percentages because the final result can be obtained by summing up or averaging the predictions of all trees [2].

The project uses a separate server developed using the high-level Python language to develop the key feature, namely predicting the probability of successful task completion [3]. High-quality models for data prediction can be implemented with this language's considerable features. The sklearn package is one of them; it offers a variety of data prediction techniques, such as the random forest algorithm. The availability of strong tools for working with HTTP protocols is another benefit of Python. Thanks to Flask and FastAPI libraries, you can

easily create secure APIs that will provide convenient access to neural networks and other machine learning models for third-party servers, such as those written for the demonstration platform, using Node.js and Next.js.

It is also worth mentioning the psycopg2 library in Python, as it is one of the most popular libraries for working with PostgreSQL. This is especially useful when collecting data for training the model, as you can get real data directly from the workspace. Thus, errors related to data corruption are avoided in advance. For example, a typo may occur when transferring data to a file manually. In addition, a direct connection to the database simplifies communication between servers and the database, avoiding unnecessary use of the central backend server for transferring data for training.

As mentioned, PostgreSQL was decided upon for data storage and high search efficiency. This database management system supports complex data types, namely JSON, which allows you to store semi-structured data with the ability to retrieve and index it quickly. In addition, most backend technologies have libraries for integrating with PostgreSQL to protect against attacks with the introduction of SQL code, dramatically simplifying development and reducing the need to focus on security aspects [4].

The PyCharm environment was used to create the part of the project developed using Python, as it provides many tools that contribute to the effective writing, testing, and debugging of code. In particular, the following capabilities were actively used during the development process: built-in tools for working with version control systems, which made it possible to easily track changes and easily switch and develop different features in parallel without the risk of stumbling upon untested developments, tools for debugging code, which helped to effectively detect and eliminate logical errors, as well as tools for automatic code formatting, which helped to keep the code in the same style throughout the development.

The application's client side, which uses the API to predict the task's success, is built using Next.js, as this framework renders pages faster, thanks to the server-side rendering capability. This modern technology allows HTML code to be generated before loading the page on the application's client side. Thanks to this, the application is SEO-friendly. In addition, Next.js is built on React, which is known for its high performance due to the use of a virtual DOM and an efficient update algorithm, making it particularly fast and effective for single-page applications. Moreover, the framework is very convenient for integration with the server part written in Node.js or Python. This ensures harmonious interaction between the client and server on both parts of the project.

As noted, the application part that provides interaction with the API server to predict the success of task execution is implemented using the Node.js platform. This technology allows you to quickly create lightweight and reliable servers for effective interaction with the client part of the application. It also has convenient libraries for working with HTTP requests and data processing. Particularly useful for long-running Python server calls is asynchrony, which allows you to perform many operations simultaneously without waiting for each one to complete. Without this feature, the system would experience significant application latency.

Visual Studio Code (VS Code) was used to develop the Node.js and Next.js parts of the application, as it is known for its ease of use when developing with JavaScript technologies. This development environment also provides an intelligent editor with code analysis capabilities, automated error detection and correction tools, and a wide selection of extensions to support various languages and technologies, which were also used in developing the project.

For quick interaction with the database, the DBeaver program was used, as it is a convenient and fast tool for working with PostgreSQL and other databases. This application also has a graphical interface for viewing tables, filtering the selection, sorting it, executing SQL queries, and administering the database. Thanks to this and the presence of script autocompletion, the process of developing and testing the data warehouse structure and queries themselves is significantly accelerated.

The application is currently undergoing functional and non-functional testing, focusing on the model results' accuracy, stability, and interpretability. This will ensure the smooth operation of the application, increase user satisfaction, and support its competitiveness in the market.

References:

1. Yang F.-J. An Extended Idea about Decision Trees. 2019 International Conference on Computational Science and Computational Intelligence (CSCI). 2019. P. 349–354. DOI: 10.1109/CSCI49370.2019.00068.

2. Identification of Brain Stroke using Boosted Random Forest / Sapra V. et al. 2022 International Conference on Advances in Computing, Communication and Materials (ICACCM). 2022. P. 1–5. DOI: 10.1109/ICACCM56405.2022.10009527.

3. S. Di Meglio. Starting a New REST API project? A Performance Benchmark of Frameworks and Execution Environments. CEUR Workshop Proceedings. 2023. Vol. 3543. P. 109–127.

4. J. A. Recio-Garcia. Case-based Explanation of Classification Models for the Detection of SQL Injection Attacks. CEUR Workshop Proceedings. 2023. Vol. 3438. P. 200–215.

УДК 004

Озорнін Я.В., студент, гр. ІПЗ-21-1, ФІКТ
Локтікова Т.М., ст. викл.
Державний університет «Житомирська політехніка»

РЕАЛІЗАЦІЯ АВТЕНТИФІКАЦІЇ ТА АВТОРИЗАЦІЇ У ВЕБЗАСТОСУНКУ ГОЛОСУВАННЯ З ВИКОРИСТАННЯМ JWT І SPRING SECURITY

У сучасному цифровому середовищі важливість забезпечення безпеки веб-застосунків зростає пропорційно до кількості персональних і чутливих даних, що передаються та зберігаються в електронному вигляді. Особливо актуальним це є для систем онлайн-голосування, які потребують надійних механізмів автентифікації та авторизації користувачів. Адже ключовими вимогами до таких систем є забезпечення цілісності голосування, запобігання повторному голосуванню, захист від підробки результатів і забезпечення доступу до системи лише авторизованим користувачам згідно з їхніми ролями. Пропонується реалізація надійного механізму безпеки на основі JSON Web Token (JWT) у поєднанні з фреймворком Spring Security – стандартним інструментом для побудови безпечних Java-застосунків.

Поняття автентифікації та авторизації хоч і тісно пов'язані, проте мають різну суть. Автентифікація – це процес перевірки особистості користувача, тобто перевірка його облікових даних (логіна й пароля або іншого способу входу), тоді як авторизація – це процес визначення рівня доступу, тобто які дії має право виконувати користувач після входу в систему. Для системи онлайн-голосування ці процеси є критично важливими: вони гарантують, що кожен користувач може отримати лише доступ, який відповідає його ролі, і не більше. Наприклад, звичайний користувач має право лише голосувати або створювати опитування, тоді як модератор – переглядати скарги або блокувати некоректні опитування, а адміністратор – мати повний доступ до всіх функцій системи.

Одним із сучасних підходів до реалізації безпечної автентифікації є використання JSON Web Token. Цей формат токенів дозволяє здійснювати автентифікацію користувача без необхідності зберігання сесійного стану на сервері. Тобто, система стає безстатевою (stateless), що значно покращує її масштабованість та продуктивність. Структура JWT (рис. 1) є простою та складається з трьох частин: заголовка (header), корисного навантаження (payload) та цифрового підпису (signature). У заголовку визначається тип токена (JWT) та алгоритм підпису (наприклад, HMAC або RSA), у payload містяться основні дані користувача – унікальний ідентифікатор, ім'я, e-mail, роль тощо, а підпис забезпечує цілісність і захист токена від модифікації. На сервері підпис перевіряється за допомогою секретного ключа або публічного сертифіката.

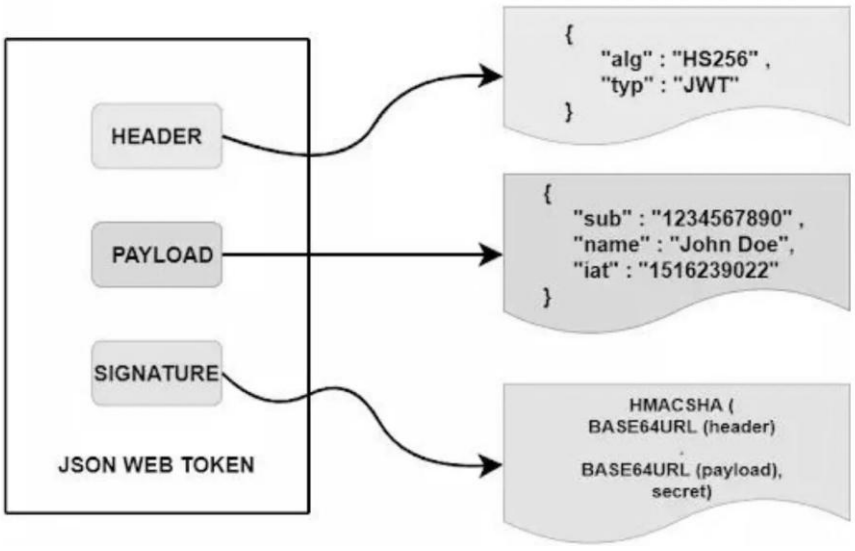

Рис. 1. Структура JWT токена

У реалізації проєкту ключову роль відіграє Spring Security – потужний фреймворк, що забезпечує гнучку та розширювану інфраструктуру безпеки в Java-застосунках. З його допомогою реалізовано фільтрацію HTTP-запитів, перевірку токенів, обмеження доступу до маршрутів за ролями, обробку помилок автентифікації та інші елементи безпеки. Конфігурація системи безпеки здійснюється за

допомогою компоненту SecurityFilterChain, де встановлюються правила, які маршрути є загальнодоступними, а які вимагають автентифікації. Також у конфігурації задається фільтр, що перехоплює всі запити, витягує з них JWT, перевіряє його дійсність, і в разі успіху передає дані користувача до контексту безпеки SecurityContextHolder. Послідовність виконання аутентифікації зображено на рис. 2.

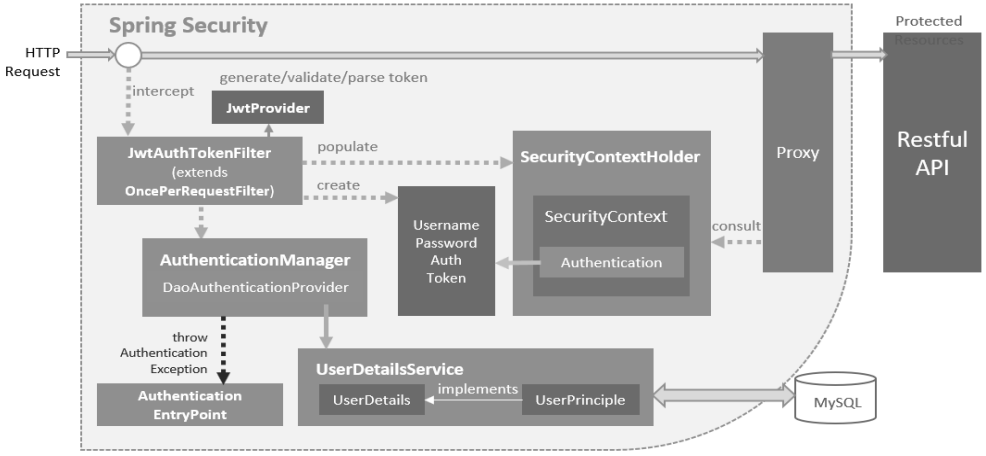

Рис. 2. Послідовність валідації JWT токена при доступі до ресурсів

Для розмежування доступу до ресурсів системи реалізовано рольову модель доступу (RBAC – Role-Based Access Control). У системі передбачено кілька ролей: "USER", "MODERATOR", "ADMIN", кожна з яких має певний набір дозволених дій. Наприклад, роль "USER" дозволяє створювати опитування та голосувати, "MODERATOR" – переглядати та редагувати публічні опитування, а "ADMIN" – керувати всіма користувачами, правами доступу, статистикою тощо. Для реалізації такого підходу використовується анотація @PreAuthorize, яка накладається безпосередньо на методи сервісів або контролерів і виконує перевірку прав доступу на рівні бізнес-логіки.

Особливої уваги потребує питання зберігання JWT на стороні клієнта. У більшості випадків токени зберігаються в localStorage або sessionStorage браузера. Цей підхід є простим у реалізації, але потребує додаткових заходів захисту. Зокрема, для запобігання атак типу Cross-Site Scripting (XSS) необхідно фільтрувати вхідні дані, уникати виконання довіреного JavaScript-коду і використовувати політику Content Security Policy (CSP). Також вся передача токенів і будь-яких конфіденційних даних повинна здійснюватися виключно через HTTPS-з'єднання, що гарантує шифрування даних і захист від атак типу Man-in-the-Middle (MITM).

Також JWT має певні обмеження, зокрема неможливість відкликання токена до завершення його терміну дії без додаткової інфраструктури. Це вирішується за допомогою реалізації blacklist-механізму або шляхом зменшення часу життя токена й введення механізму "refresh token". Такий підхід дозволяє підтримувати короткотривалі токени доступу й окремо зберігати refresh-токени для продовження сесії без повторного логіну користувача. Проте refresh-токени повинні зберігатися більш безпечно, наприклад, у HTTP-only куках, які не доступні JavaScript-коду.

Таким чином, реалізація автентифікації та авторизації з використанням JWT і Spring Security дозволило створити надійне, масштабоване й сучасне рішення для системи онлайн-голосування. Поєднання безстатевої моделі автентифікації з гнучкою системою обмежень доступу за ролями гарантує, що кожен користувач може виконувати лише ті дії, які дозволені йому згідно з його статусом у системі. А використання перевірених рішень, а саме Spring Security, забезпечує не лише безпеку, а й простоту розширення та підтримки проєкту в майбутньому. Результатом такої реалізації є система, здатна ефективно протистояти зовнішнім загрозам, гарантувати конфіденційність, цілісність і доступність даних, а також відповідати сучасним вимогам до побудови безпечних інформаційних систем.

Список використаних джерел:

1. Spring Boot – фреймворк Java з відкритим вихідним кодом [Електронний ресурс] – Режим доступу до ресурсу: https://spring.io/projects/spring-boot

2. Spring Security – модуль Spring для забезпечення автентифікації та авторизації [Електронний ресурс] – Режим доступу до ресурсу: https://spring.io/projects/spring-security

3. JWT – JSON Web Token. Офіційна документація [Електронний ресурс] – Режим доступу до ресурсу: https://jwt.io/introduction

УДК 004.4

Гаманюк І.А., студент, гр. ІПЗ-21-1, ФІКТ
Локтікова Т.М., ст. викл.
Державний університет «Житомирська політехніка»

РОЗРОБКА ІНТЕРАКТИВНОЇ ОСВІТНЬО-ТЕСТУВАЛЬНОЇ ПЛАТФОРМИ З ІГРОВОЮ МОТИВАЦІЄЮ

Пропонується розробка інтерактивної освітньо-тестувальної платформи з ігровою мотивацією. Пропонована платформа поєднує в собі кілька взаємопов'язаних модулів, об'єднаних єдиною архітектурою на базі Next.js [1], MongoDB [2] і деплойменту на Vercel [3]. Ідея платформи полягає в тому, щоб учасники могли проходити навчальні модулі та тестові сесії, заробляти внутрішню валюту, відкривати віртуальні кейси з предметами різної цінності та рідкісності, нарощувати ігровий прогрес і аналізувати власні результати за допомогою детальної статистики. Усе це поєднується з бойовим форматом тестів, який дозволяє візуалізувати рівень майстерності користувача в ігровій формі. Бойова система не є обов'язковою для проходження навчання – учасник може ігнорувати ігрові елементи, залишаючись на початковому рівні з мінімальними анімаціями та базовою взаємодією.

Концептуально платформа складається з кількох ключових компонентів. Першим є модуль тестів і навчальних вправ, який підтримує різноманітні формати запитань: множинний вибір, заповнення пропусків, відповідність тощо. Після проходження блоків тестів учасник отримує бали внутрішньої валюти, яку надалі може обмінювати на кейси з предметами. Другий модуль – кейсів – передбачає механіку випадкового винагородження: учасник відкриває кейс, отримує екіпірування, аватарки, нікнейми з різними властивостями та візуальними ефектами. Ці предмети можуть підвищувати атаку або захист у бойовому режимі, який є центральним ігровим елементом платформи. Бойовий режим побудовано на схемі «коректна відповідь = удар по ворогу», де графічні анімації реалізуються за допомогою CSS [4] і Framer Motion [5].

Система ранжування ґрунтується на статистиці останніх сеансів користувача; нові звання автоматично присвоюються залежно від досягнень і середнього відсотка правильних відповідей. Якщо статистичних даних недостатньо, то звання не присвоюється, і учасник залишається на стартовому рівні. Одночасно реалізована система персональних досягнень, яка фіксує виконання певних умов, наприклад, «серія з десяти правильних відповідей поспіль» або «відкриття п'ятдесятого кейсу». Додатково існують щоденні виклики з особливими нагородами, які стимулюють регулярну взаємодію з платформою.

Аналітичний модуль формує після кожного тесту сторінку з докладним розбиттям результатів – кількість правильних і неправильних відповідей, відсоток успішності за категоріями, час на одне питання і загальний час проходження. Ці дані зберігаються в базі та відображаються в розділі «Прогрес», де можна побачити графіки динаміки рівня знань, сильні і слабкі теми, а також зміну показників з часом. Усі дані взаємодії з платформою керуються через zustand [6], який є менеджером стану, що дозволяє швидко оновлювати інтерфейс без зайвих перезавантажень.

Інтерфейс платформи спроєктовано за принципами мінімалізму та адаптивності. Основні кольори й теми можна налаштовувати у профілі, а для тих, хто цінує простоту, передбачено класичну світлу й темну схеми. Аватарки, нікнейми, комплекти екіпірування доступні у внутрішньому магазині за зароблені бали. Модерація контенту не потрібна, оскільки всю візуальну складову розробляє адміністрація через окрему панель керування. Користувачі не можуть додавати власні зображення чи повідомлення, а значить платформа позбавлена ризику небажаного контенту.

Реєстрація не є обов'язковою: новачок може почати проходження тестів одразу ж, без введення електронної пошти чи пароля, в гостьовому режимі.

Усе спілкування між клієнтом і сервером забезпечується клієнт-серверною архітектурою, де клієнт відправляє запити за потребою, а сервер відповідає лише необхідними даними. Це зменшує навантаження на мережу, підвищує швидкодію та дозволяє обробляти кілька паралельних запитів. За рахунок розділення бізнес-логіки й інтерфейсу досягається висока гнучкість у підтримці та масштабуванні, зокрема можливість додавати нові клієнтські застосунки або мобільні версії.

Для зберігання даних вибрано базу MongoDB через її гнучкість у роботі з документаціями, відсутність жорсткої схеми і можливість масштабування. Зміни структури даних відбуваються без складних міграцій, що важливо для стартап-проєктів із частими оновленнями функціоналу.

Таким чином, платформа поєднує навчальний контент із гейміфікованою системою, що підвищує мотивацію до навчання за рахунок елементів рольової гри та колекціонування екіпірування. Використання сучасного стека Next.js, MongoDB, Vercel і zustand забезпечує гнучкість в оновленнях, високу швидкодію та адаптивність під зростання навантаження. Такий підхід дозволяє створити привабливий, безпечний і масштабований продукт, який сприятиме підвищенню рівня знань і розвитку критичного мислення користувачів.

Для реалізації інтерактивної освітньо-тестувальна платформи було спроєктовано базу даних. Діаграма створеної бази даних представлена на рисунку 1.

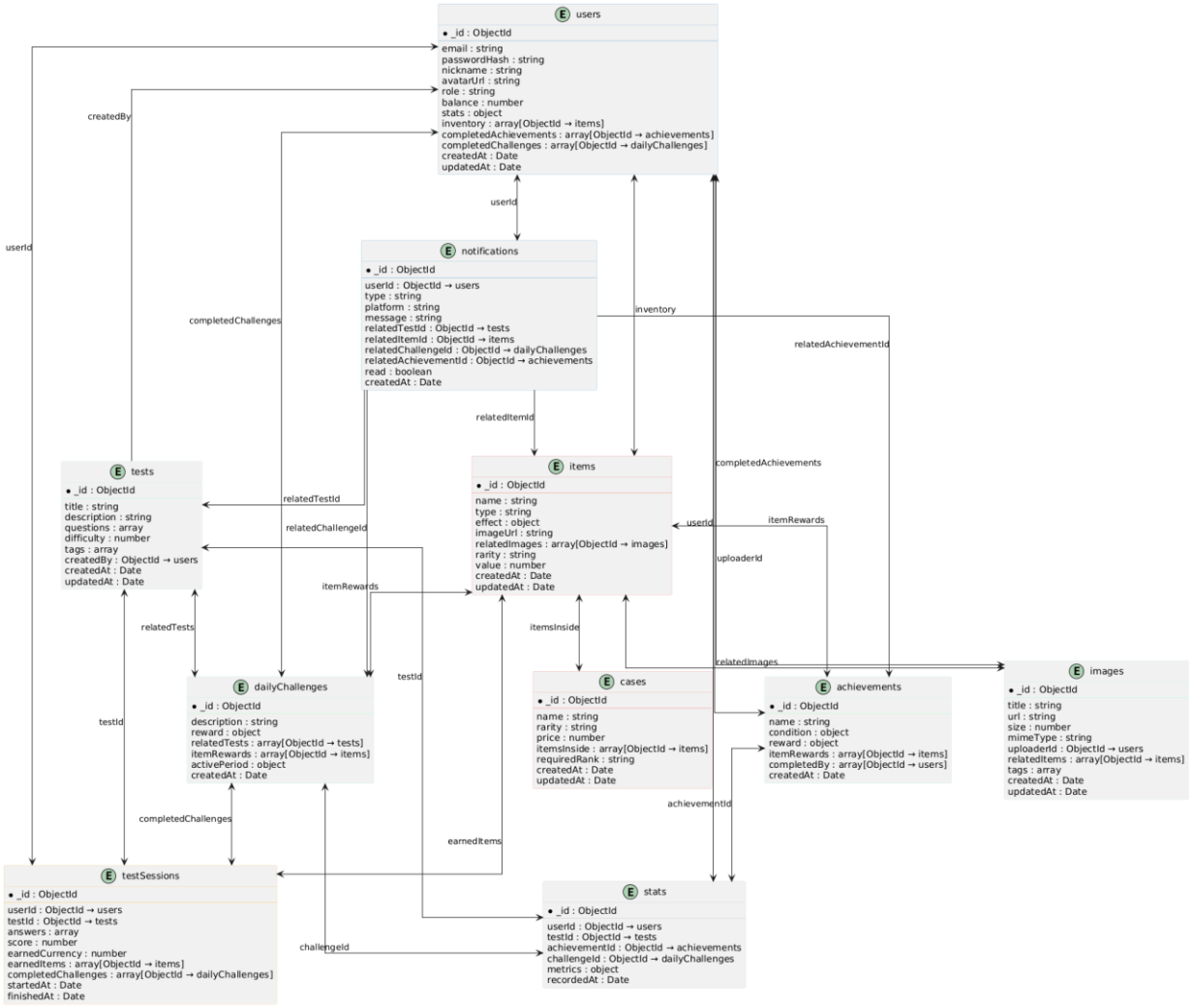

Рис. 1. Діаграма бази даних

Розроблений застосунок проходить як функціональне, так і нефункціональне тестування, що гарантує відповідність програмного забезпечення встановленим вимогам і стандартам – важливому аспекту в процесі розробки.

Список використаних джерел:

1. Адмін. Introduction | Next.js. Next.js by Vercel - The React Framework. URL: https://nextjs.org/docs (дата звернення: 01.05.2025).

2. Team M. D. MongoDB Documentation. MongoDB: The World’s Leading Modern Database | MongoDB. URL: https://www.mongodb.com/docs/ (дата звернення: 01.05.2025).

3. Vercel Team. Vercel Documentation. Vercel: Build and deploy the best web experiences with the Frontend Cloud. URL: https://vercel.com/docs (дата звернення: 02.05.2025).

4. MDN Team. CSS: Cascading Style Sheets | MDN. MDN Web Docs. URL: https://developer.mozilla.org/en-US/docs/Web/CSS (дата звернення: 04.05.2025).

5. Motion Team. Docs - Motion. Motion - A modern animation library for JavaScript, React and Vue. URL: https://motion.dev/docs (дата звернення: 05.05.2025).

6. Адмін. Introduction - Zustand. Poimandres documentation. URL: https://zustand.docs.pmnd.rs/getting-started/introduction (дата звернення: 06.05.2025).

УДК 004

Гнатюк Д.С., студент, гр. ІПЗ-21-3, ФІКТ
Локтікова Т.М., ст. викл.
Державний університет «Житомирська політехніка»

СИСТЕМА ПЕРЕКЛАДУ З ЕЛЕМЕНТАМИ НАВЧАННЯ

В умовах глобалізації значна кількість користувачів щоденно зіштовхується з необхідністю опрацювання іншомовного контенту в Інтернеті. Знання іноземних мов при цьому виступає одним із ключових чинників особистої та професійної успішності. Існуючі інструменти для перекладу зазвичай забезпечують лише базовий фрагментарний функціонал, не підтримуючи процес навчання, запам'ятовування й осмислення нового мовного матеріалу. Крім того, необхідність окремих засобів для перекладу, збереження та повторення інформації ускладнює навчальний процес і знижує мотивацію до систематичного вивчення мови.

У відповідь на вказані виклики запропоновано створити мультифункціональну систему, яка інтегрується у повсякденну роботу користувача та об'єднує інструменти перекладу й освоєння мови. Такий підхід сприяє полегшенню взаємодії з іншомовними ресурсами та безперервному розвитку мовних навичок у природному середовищі. Розроблена система дозволяє перекладати тексти безпосередньо під час перегляду веб-ресурсів, зберігати переклади в особисту базу даних, редагувати їх, повторювати вивчене, а також отримувати навчальні матеріали, сформовані на основі особистих даних користувача. До складу функціоналу також входить інтегрований чат зі штучним інтелектом, розширена персоналізація та інші додаткові можливості, що виділяють продукт на фоні аналогів.

Система реалізована за принципами розподіленої клієнт-серверної архітектури. Вона складається з віддаленого сервера з глобальною базою даних та клієнтської частини, що включає десктопний застосунок, браузерне розширення, локальний сервер і локальну базу даних. Такий підхід дозволяє забезпечити централізований доступ до даних через хмару, підтримку офлайн-режиму, ефективну синхронізацію між пристроями користувача, масштабованість та зручність у розгортанні.

Віддалений сервер реалізований на основі платформи ASP.NET Web API, яка забезпечує зручну реалізацію RESTful-сервісів, підтримує асинхронну обробку запитів і легко масштабується. Логіка сервера інкапсулює взаємодію з базою даних та зовнішніми сервісами, надаючи лише публічне API, що сприяє безпеці та ізоляції даних.

У якості СУБД обрано SQL Server. Для локального зберігання даних використовується LocalDB — легковаговий варіант зберігання у вигляді файлу, а для хмарної частини – Azure SQL Database, що забезпечує резервне копіювання, масштабованість та високий рівень захисту. Єдина структура обох баз даних дозволила створити спільну бібліотеку з моделями та контекстом, яку використовують як клієнтська, так і серверна частини.

Окремий локальний сервер також реалізований на ASP.NET Web API, що дало змогу уніфікувати технологічний стек і повторно використовувати значну частину коду. Цей сервер виступає як точка взаємодії для браузерного розширення та десктопного застосунку. Така архітектура надає можливість обходити обмеження безпеки, накладені на сучасні браузери, працювати з кешованими сторінками без доступу до інтернету та уникати дублювання логіки між компонентами.

Десктопний застосунок розроблений на основі технології WPF, що забезпечує можливість створення зручного та функціонального інтерфейсу, а також дозволяє повторно використовувати спільні компоненти, моделі даних та інтерфейси, які застосовуються в усій системі. Основна роль застосунку полягає у забезпеченні повноцінної роботи з локальним сервером: він відповідає за авторизацію користувача, керування перекладами, доступ до навчальних матеріалів, організацію повторення вивченого та інші пов'язані функції, які виходять за межі базового перекладу.

Браузерне розширення використовує бібліотеку React для створення сучасного, адаптивного та зрозумілого інтерфейсу, а мова TypeScript застосовується для більш безпечної роботи з даними. Основна функція розширення — надання зручного інструменту для перекладу безпосередньо під час перегляду веб-сторінок. Для підвищення продуктивності розробки використовується компілятор Vite, а також бібліотеки, що дозволяють миттєво застосовувати зміни під час розробки без необхідності ручного оновлення розширення у браузері, що значно прискорює тестування.

Список використаних джерел:

1. Документація ASP.NET [Електронний ресурс] – Режим доступу до ресурсу: https://learn.microsoft.com/en-us/aspnet/core.
2. Документація React [Електронний ресурс] – Режим доступу до ресурсу: https://react.dev/learn.

УДК 004

Керест Н.І., студент, гр. ІПЗ-21-1, ФІКТ
Локтікова Т.М., ст. викл.
Лисогор Ю.І., ст. викл.
Державний університет «Житомирська політехніка»

PWA ВЕБМЕСЕНДЖЕР З ПОКРАЩЕНОЮ АВТЕНТИФІКАЦІЄЮ ТА ФУНКЦІЄЮ ОБМІНУ КОНТЕНТОМ

Останнім часом, з розвитком цифрових послуг, люди все більше потребують простих, безпечних та універсальних інструментів для спілкування, які можна легко налаштувати під свої потреби. Саме тому пропонований вебмесенджер, розроблений за технологією Progressive Web App (PWA), є актуальним та ефективним рішенням. Він поєднує в собі зручність, надійність та широкий спектр функцій, забезпечуючи комфортне спілкування на будь-якому пристрої, особливо на мобільних телефонах, що є критично важливим у сучасному активному світі.

Завдяки клієнт-серверній архітектурі та об'єктно-орієнтованому дизайну, проєкт має чітку і зрозумілу структуру. Це значно спростило його розробку, а в подальшому – обслуговування та масштабування. Використання діаграми класів (рис. 1) допомогло розподілити відповідальність між компонентами, що робить код більш організованим і зрозумілим, сприяючи швидкій адаптації до нових вимог.

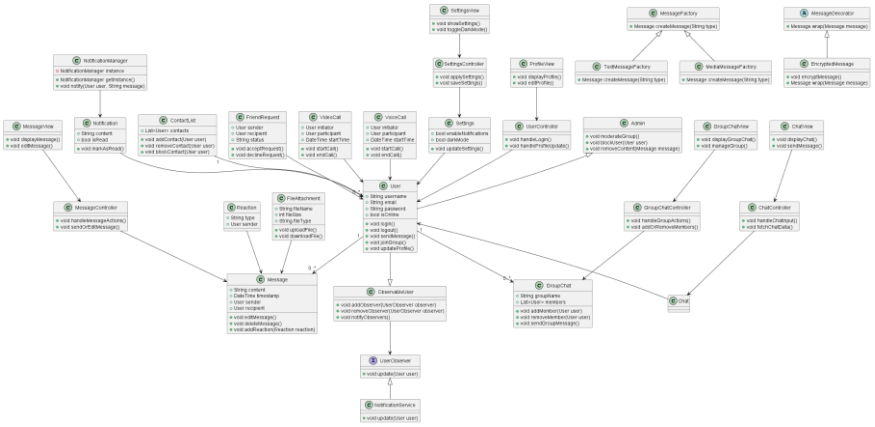

Рис. 1. Діаграма класів вебзастосунку месенджера

Головні функції вебзастосунку охоплюють основні аспекти сучасного цифрового спілкування, зосереджуючи увагу на зручності, функціональності та безпеці. Зокрема, забезпечено захищений доступ до системи – користувачі можуть реєструватися та входити за допомогою номерів телефонів або електронних адрес. Для додаткового захисту реалізовано підтвердження особи через SMS або e-mail, можливість входу за допомогою QR-коду та двофакторну автентифікацію, що суттєво зменшує ризики несанкціонованого доступу. Спілкування відбувається в режимі реального часу – користувачі можуть швидко обмінюватися текстовими повідомленнями, редагувати або видаляти їх, а також відстежувати статус доставки і прочитання. Для ефективної командної роботи передбачена можливість створення групових чатів з гнучкими налаштуваннями – можна додавати чи видаляти учасників, а також визначати їхні ролі та дозволи. Окрім текстових повідомлень, система дозволяє надсилати та отримувати мультимедійні файли, а саме картинки та відео, а також обмінюватися документами та архівами різних форматів, що значно розширює можливості використання застосунку як у повсякденному, так і в професійному житті. Ключові класи, такі як “User”, “Message” та “GroupChat”, формують основу для комплексної системи обміну повідомленнями. Дотримання принципів SOLID та застосування патернів проєктування Observer, Factory та Singleton забезпечують високу стабільність, гнучкість та масштабованість системи. Інтеграція з PWA дозволяє використовувати месенджер на різних платформах без необхідності встановлення, а також частково в режимі офлайн. Це забезпечує зручність використання як для особистих, так і для корпоративних потреб, де критично важливими є надійність, функціональність та безпека передачі даних.

Список використаних джерел:

1. Документація Progressive Web Apps [Електронний ресурс] – Режим доступу до ресурсу: https://developer.mozilla.org/en-US/docs/Web/Progressive_web_apps.

2. Документація OWASP Authentication Cheat Sheet [Електронний ресурс] – Режим доступу до ресурсу: https://cheatsheetseries.owasp.org/cheatsheets/Authentication_Cheat_Sheet.

УДК 004

Зіневич Р.Р., студент, гр. ІПЗ-21-1, ФІКТ
Локтікова Т.М., ст. викл.
Державний університет «Житомирська політехніка»

КАЛЬКУЛЯТОР ДЛЯ РОЗРАХУНКІВ КОНФІГУРАЦІЇ ПК

Головна проблема під час складання ПК — це сумісність деталей. Наприклад, процесор може не підійти до материнської плати, відеокарта не поміститься в корпус або блок живлення не витягне все навантаження. Також потрібно враховувати охолодження, щоб комп'ютер не перегрівався. Користувачам без досвіду або технічної освіти доводиться витрачати значну кількість часу на вивчення характеристик, порівняння цін і перевірку сумісності. В результаті можна купити не те, що треба, або переплатити.

Метою запропонованого проєкту є створення вебзастосунку, який допомагатиме користувачам підбирати комплектуючі для персонального комп'ютера. Застосунок має забезпечувати простий та інтуїтивно зрозумілий інтерфейс, перевірку сумісності в режимі реального часу та врахування фінансових обмежень користувача.

Під час дослідження були оглянуті й проаналізовані існуючі аналогічні сервіси.

PCPartPicker.com [1] — популярний сервіс для підбору комплектуючих. Має розширену базу товарів, автоматичну перевірку сумісності, можливість збереження конфігурацій та моніторинг цін. Основним недоліком є орієнтація на ринок США, що знижує його зручність для користувачів з України.

Telemart.ua [2] — українська платформа, з актуальними цінами та можливістю підбору комплектуючих. Зручна для локального ринку, однак обмежена в гнучкому налаштуванні конфігурацій, не завжди коректно працює з менш популярними моделями деталей.

BuildMyPC.com [3] — має базовий функціонал перевірки сумісності, дозволяє створювати бюджетні збірки. Простий у використанні, але прив'язаний лише до Amazon, працює повільно і має обмежений набір налаштувань.

На основі проведених огляду й аналізу було створено власний вебзастосунок, адаптивний до різних пристроїв, зі зручним інтерфейсом і швидкою перевіркою сумісності компонентів. Для розробки використано сучасні інструментальні засоби: React з TypeScript — для клієнтської частини, Tailwind CSS — для стилізації, Node.js — для серверної логіки та MongoDB — як базу даних.

У застосунку реалізовано можливість збереження створених конфігурацій, що дозволяє повертатись до них пізніше або ділитися з іншими. Система також підтримує попередній перегляд збірки з короткою аналітикою: виводиться орієнтовна вартість, основні характеристики, сумарне енергоспоживання та розміри. Такий функціонал дозволяє користувачу швидко оцінити, наскільки обрана конфігурація відповідає заданим умовам і вимогам. Це особливо актуально для тих, хто планує проконсультуватися перед покупкою або зібрати ПК поступово.

Реалізовано фільтрацію за ціною, брендом, категоріями та додатковими характеристиками, а саме розмірами корпусу, енергоспоживанням, рівнями шуму. Це дозволяє не тільки перевірити сумісність, а й налаштувати систему під власні потреби. Крім того, впроваджено можливість обміну готовими збірками між користувачами. Інформація про компоненти братиметься з відкритих джерел або додаватиметься вручну, що підвищує гнучкість платформи та достовірність даних.

У застосунку реалізовано функціонал користувацьких акаунтів. Користувачі мають можливість зберігати історію вибраних конфігурацій, редагувати їх, а також ділитися з іншими користувачами. Це сприяє створенню активної спільноти навколо сервісу та дозволяє отримувати корисну інформацію й поради від інших відвідувачів платформи.

У ході розробки було враховано недоліки існуючих рішень, що дало змогу створити систему, краще адаптовану до потреб українських користувачів. Запропонований вебзастосунок забезпечує не лише перевірку сумісності компонентів, а й надає зручні та гнучкі інструменти для аналізу, підбору та збереження конфігурацій відповідно до особистих вимог і фінансових можливостей.

Таким чином, розроблена вебплатформа поєднує в собі простоту використання, необхідний функціонал і гнучкість, створюючи ефективний інструмент для сучасного користувача, який прагне самостійно зібрати персональний комп'ютер без зайвих труднощів.

Список використаних джерел:

1. PCPartPicker.com [Електронний ресурс] – Режим доступу до ресурсу: https://pcpartpicker.com/.
2. Telemart.ua [Електронний ресурс] – Режим доступу до ресурсу: https://telemart.ua/ua/.
3. BuildMyPC.com [Електронний ресурс] – Режим доступу до ресурсу: https://buildmypc.net/.
4. React – Офіційний сайт [Електронний ресурс] – Режим доступу: https://reactjs.org/.
5. Tailwind CSS – Офіційний сайт [Електронний ресурс] – Режим доступу: https://tailwindcss.com/.
6. Node.js – Офіційний сайт [Електронний ресурс] – Режим доступу: https://nodejs.org/.

УДК 004

Кириченко О.С., студент, гр. ІПЗ-21-1, ФІКТ
Локтікова Т.М., ст. викл.
Державний університет «Житомирська політехніка»

ОРГАНАЙЗЕР З КЕРУВАННЯ ВИТРАТАМИ

Пропонується розробка спеціалізованого застосунку для легкого та простого керування власними фінансами, що надає користувачам засоби для їхнього детального обліку, перегляду, планування та інвестування. Програмний продукт забезпечує зручний доступ до фінансових даних як з комп'ютерів, так і з мобільних пристроїв.

Основний функціонал охоплює облік доходів і витрат на рахунках із різними валютами, детальне групування фінансових операцій, створення фінансових цілей та моніторинг бюджету. Застосунок також надає можливість обліку інвестицій для відстеження активів й їхньої ефективності, налаштування повторюваних операцій, використання тегів для гнучкого групування та створення структурованих фінансових звітів.

Пропонований застосунок побудовано за клієнт-серверною архітектурою, де серверна частина забезпечує виконання бізнес-логіки та керування даними, а клієнтські застосунки – їх представлення. Це дозволяє передавати або отримувати дані лише за потреби, зменшуючи навантаження на мережу та підвищуючи ефективність обміну інформацією. Крім того, можливість асинхронної роботи клієнта та сервера дозволяє обробляти запити без блокування інтерфейсу користувача, покращуючи загальну продуктивність і відгук системи. Також обрана архітектура сприяє гнучкості у розгортанні системи і дозволяє легко під’єднувати нові типи клієнтських програм (наприклад, для різних платформ), які будуть взаємодіяти з тим самим API.

Для реалізації серверної частини проєкту обрано PHP-фреймворк Laravel. Хоча існують інші потужні PHP-фреймворки, наприклад, Symfony, який надає більшу гнучкість та набір компонентів для корпоративних рішень, Laravel був обраний як такий, що сприяє підвищенню швидкості розробки для одного розробника або маленьких груп. Цей фреймворк має багато інструментів, наприклад, Eloquent ORM для полегшення роботи з базою даних, потужну маршрутизацію, вбудовані механізми для побудови API та засоби безпеки. Це дозволяє швидше впроваджувати бізнес-логіку на відміну від методів, які потребують більше ручних налаштувань або написання коду на PHP, де немає готової структури та захисту.

Для зберігання даних обрано PostgreSQL – потужну об'єктно-реляційну систему для управління базами даних. Цей вибір зумовлений високою надійністю PostgreSQL, дотриманням ACID-властивостей, що є важливими для забезпечення цілісності фінансових угод, і можливістю виконувати складні запити. В порівнянні з MySQL, PostgreSQL частіше обирають за кращу відповідність стандартам SQL та більшу стабільність при складних операціях. Для фінансового застосунку, для якого точність та узгодженість даних є головними пріоритетами, переваги реляційної системи, а саме PostgreSQL над NoSQL-рішеннями, які можуть надати більшої гнучкості структури, але за рахунок гарантій транзакції, є визначальними.

Автентифікація користувачів та захист точок доступу до API реалізовані за допомогою JSON Web Tokens (JWT). На відміну від звичайних сесій, що зберігають стан на сервері та можуть ускладнити масштабування, JWT не зберігають стану на сервері. Це робить їх кращим вибором для роботи між SPA-клієнтами, мобільними додатками та сервером, оскільки кожен запит містить всю потрібну інформацію для автентифікації. Це стандартний безпечний і гнучкий підхід для сучасних API.

Для створення клієнтської частини проєкту, з якою безпосередньо взаємодіють користувачі, обрана JavaScript-бібліотека React. У порівнянні з іншими фреймворками, а саме Angular або Vue.js, React відзначається більшою гнучкістю завдяки своїй компонентній структурі і орієнтованості на UI. Використання віртуального DOM гарантує високу швидкість оновлення інтерфейсу, що робить його більш швидким та чутливим до дій користувача, особливо важливими для SPA.

Для створення мобільних застосунків для iOS та Android використовується платформа Expo.js, яка базується на React Native. Це надає можливість застосовувати одну кодову базу JavaScript/React для обох платформ, що є значно ефективнішим, ніж окрема розробка на Swift/Objective-C для iOS і Kotlin/Java для Android. У порівнянні з React Native, Expo спрощує формування мобільних застосунків, доступ до нативних API, а також має багато готових модулів, що також є його перевагою.

Для оформлення інтерфейсів та забезпечення сучасного пристосованого дизайну на всіх платформах використовується фреймворк Tailwind CSS. На відміну від бібліотек компонентів типу Bootstrap або Material UI, що надають готові компоненти, які зазвичай важко налаштувати, Tailwind CSS пропонує підхід на основі утилітарних класів. Це забезпечує повний контроль над виглядом кожного елементу за допомогою набору низькорівневих класів-утиліт CSS, дозволяючи швидко створити унікальний дизайн без написання великої кількості власного коду.

Для структурування та зберігання всієї фінансової інформації спроєктовано реляційну базу даних. Детальна структура цієї бази даних, що включає всі таблиці та їхні взаємозв'язки, представлена на ER-діаграмі (рис. 1).

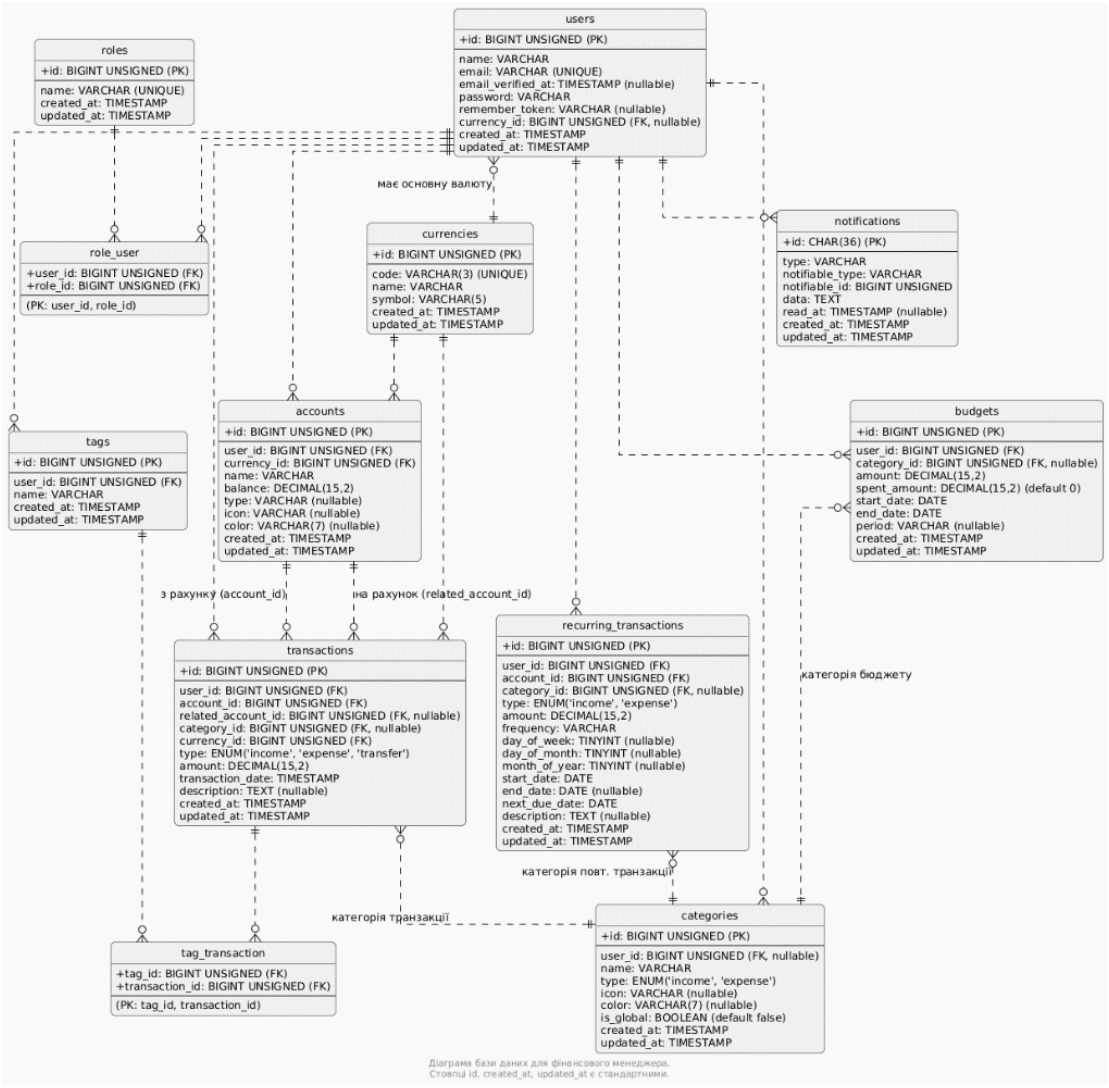

Рис. 1. ER-діаграма створеної бази даних

Обраний стек технологій складається з програмного забезпечення з відкритим вихідним кодом, такого як PHP, Laravel, PostgreSQL, React, Expo.js та Tailwind CSS. Це означає, що розробка та використання пропонованого продукту не потребують витрат на комерційне ліцензування для цих ключових компонентів.

Наразі розроблений органайзер з керування витратами проходить як функціональне, так і нефункціональне тестування. Це допоможе отримати застосунок, який відповідатиме всім визначеним вимогам.

Список використаних джерел:

1. Документація PHP [Електронний ресурс] – Режим доступу до ресурсу: https://www.php.net/manual/uk/

2. Документація Laravel [Електронний ресурс] – Режим доступу до ресурсу: https://laravel.com/docs/

3. Документація pgSQL [Електронний ресурс] – Режим доступу до ресурсу: https://www.postgresql.org/docs/

4. Документація React [Електронний ресурс] – Режим доступу до ресурсу: https://react.dev/

5. Документація Tailwind [Електронний ресурс] – Режим доступу до ресурсу: https://tailwindcss.com/docs/

УДК 004

Рикун А.О., студент, гр. ІПЗ-21-1, ФІКТ
Кушнір Н.О., ст. викл.
Державний університет «Житомирська політехніка»

РОЗРОБКА ПЛАТФОРМИ ДЛЯ ОЦІНЮВАННЯ ТА ОБГОВОРЕННЯ ФІЛЬМІВ

У сучасному цифровому середовищі інтерес до обговорення кінофільмів у мережі зростає з кожним роком. Користувачі прагнуть не лише переглядати фільми, а й ділитися враженнями, оцінювати, залишати відгуки та брати участь в обговореннях. У зв'язку з цим виникла ідея створення веб-платформи, що дозволить користувачам взаємодіяти навколо кіноконтенту, залишаючи власні рецензії, оцінки, коментарі та скарги на недоречні публікації. Для реалізації проєкту було обрано стек технологій, орієнтований на продуктивність і простоту розробки. Серверна частина побудована за допомогою фреймворку Laravel, який забезпечує чітке розділення логіки за архітектурою MVC (Model-View-Controller), дозволяє організувати зручне маршрутизування. Для представлення інтерфейсу використовується шаблонізатор Blade, а для стилізації - утилітарний CSS-фреймворк Tailwind CSS, який забезпечує швидке створення адаптивного сучасного дизайну.

Структурно платформа поділяється на кілька основних модулів. Модуль аутентифікації та авторизації керує реєстрацією, входом і розмежуванням прав доступу для користувачів та адміністраторів. Модуль управління фільмами дозволяє адміністраторам додавати, редагувати, переглядати та видаляти фільми. Модуль оцінювання дає змогу користувачам виставляти фільмам оцінки та формує середній рейтинг. Модуль відгуків та коментарів реалізує публікацію рецензій користувачами та коментування цих рецензій іншими учасниками платформи. Модуль скарг забезпечує надсилання скарг на порушення правил у відгуках і коментарях з можливістю подальшої модерації з боку адміністратора. Ключовою складовою етапу проєктування платформи стало створення ER-діаграми, яка відображає структуру реляційної бази даних та взаємозв'язки між основними сутностями. Основні сутності включають користувачів з вказаними ролями, фільми, відгуки, коментарі до відгуків, оцінки, акторів, а також скарги на відгуки. Наприклад, один користувач може залишати багато відгуків, кожен фільм має кілька оцінок і коментарів, а також пов'язаний із багатьма акторами. Передбачено зберігання скарг на недоречні рецензії. Така модель забезпечує гнучкість і масштабованість системи.

ER-діаграма бази даних представлена на рис. 1. Як уже відзначалось, ER-діаграма відіграє важливу роль у візуалізації логіки побудови бази даних і допомагає наочно зрозуміти зв'язки між таблицями, що надалі спрощує реалізацію міграцій, написання запитів та забезпечення цілісності даних у системі.

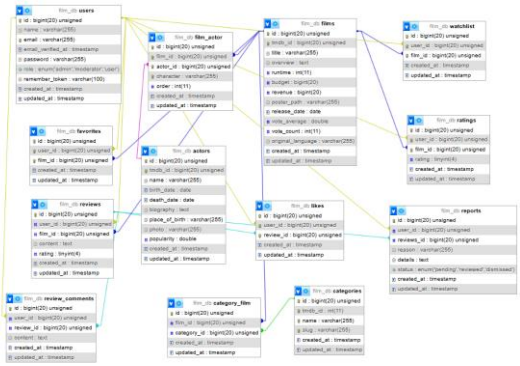

Рис. 1. ER-діаграма бази даних

Уся побудована структура дозволяє підтримувати чисту та розширювану кодову базу, де кожен компонент виконує чітко визначену роль. Це не лише спрощує підтримку та розвиток системи, а й дозволяє легко впроваджувати нові функціональні можливості в майбутньому, наприклад, реалізацію пошуку фільмів через зовнішній API, персоналізовані рекомендації або підтримку медіафайлів.

Список використаних джерел:

1. Laravel – The PHP Framework for Web Artisans [Електронний ресурс] – Режим доступу до ресурсу: https://laravel.com/

2. Tailwind CSS – Rapidly build modern websites [Електронний ресурс] – Режим доступу до ресурсу: https://tailwindcss.com/

3. MySQL :: MySQL Community Edition [Електронний ресурс] – Режим доступу до ресурсу: https://dev.mysql.com/

УДК 612.67

Сімчук А.Р., аспірант
Мацієвський В.А., аспірант
Нікітчук Т.М., к.т.н., доцент
Коломієць Р.О., к.т.н., доцент
Державний університет «Житомирська політехніка»

ДОСЛІДЖЕННЯ ПРОЦЕСІВ ПРИЙОМУ ТА ПЕРЕДАЧІ БІОМЕДИЧНИХ СИГНАЛІВ ЗА ДОПОМОГОЮ IOT-ТЕХНОЛОГІЙ

Біомедичні сигнали – це сигнали, які генеруються в організмі людини та можуть відображати інформацію про фізіологічні процеси, які у ньому проходять. Вони є важливим джерелом даних у медицині, оскільки надають можливість оцінити стан організму без прямого втручання. З розвитком технологій Інтернету Речей (IoT) виникають нові можливості для моніторингу і управління здоров'ям пацієнтів, що суттєво покращує якість медичного обслуговування. Біомедичні сигнали, такі як електрокардіограми (ЕКГ), електроенцефалограми (ЕЕГ), рівень кисню в крові та інші параметри, є ключовими для діагностики, лікування та профілактики захворювань. Інтеграція цих сигналів з IoT-системами відкриває нові горизонти для бездротового моніторингу, дистанційного керування та інтеграції даних у реальному часі.

За даними дослідницького агентства Transforma Insights, до кінця 2030 року частка сектору охорони здоров'я в розвитку інтернету речей становитиме $4,7 млрд, що відповідає 17 % усього ринку IoT. Internet of Things є мережею з фізичних об'єктів і пристроїв, які можуть обмінюватися між собою інформацією Введення технологій IoT у медицині дозволило спростити проведення операцій, покращити досвід лікаря під час моніторингу здоров'я пацієнтів, контролювати витрати, підвищити продуктивність [1].

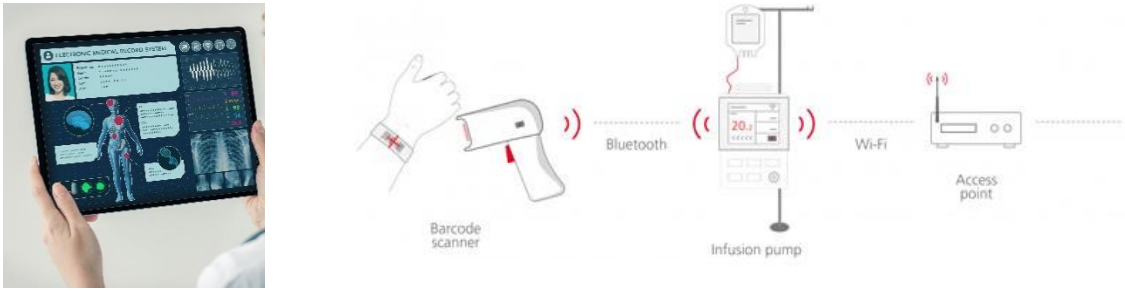

Рис. 1. Приклад використання IoT у медицині [1, 4]

Актуальність теми обумовлена потребою в розробці ефективних методів для прийому та передачі біомедичних сигналів через IoT-платформи. Зокрема, існує необхідність в підвищенні точності, надійності та безпеки передачі медичних даних, а також в оптимізації використання IoT-ресурсів для забезпечення швидкого і безперебійного обміну інформацією.

Мета даної роботи полягає в дослідженні процесів прийому та передачі біомедичних сигналів за допомогою IoT-технологій. Для досягнення цієї мети необхідно вирішити кілька завдань: проаналізувати основи IoT для медичних застосувань, оцінити методи збору і обробки біомедичних даних, розглянути протоколи і технології передачі даних в IoT-системах, а також вивчити вплив зовнішніх факторів на якість сигналів.

Об'єктом дослідження є процеси збору, передачі та обробки біомедичних сигналів в рамках IoT-інфраструктури, а предметом дослідження є методи і засоби, що забезпечують ефективну інтеграцію і управління медичними даними.

У роботі проведено детальний аналіз методів збору та обробки біомедичних даних для IoT [1-3], включаючи розгляд конкретних типів сенсорів та їхніх характеристик (точність, енергоспоживання, розміри), аналіз методів стиснення даних для ефективної передачі через IoT-мережі, наведено опис алгоритмів попередньої обробки сигналів (фільтрація шумів, усунення артефактів). Розглянуті методи збору біомедичних даних демонструють різний рівень ефективності щодо точності вимірювань, енергоспоживання та компактності сенсорів. Вибір конкретного сенсора визначається типом сигналу, вимогами до точності діагностики та обмеженнями щодо мобільності та тривалості роботи пристрою. Щодо методів обробки, алгоритми попередньої фільтрації є ефективними у зменшенні рівня шумів та артефактів, що підвищує якість подальшого аналізу. Однак, складніші алгоритми можуть вимагати значних обчислювальних ресурсів та енергії, що є критичним для малопотужних IoT-пристроїв. Методи стиснення даних є необхідними для оптимізації передачі великих обсягів біомедичної інформації через

мережі з обмеженою пропускною здатністю, але ступінь стиснення може впливати на точність відновлення сигналу.

Розгляд протоколів та технологій передачі даних в IoT-системах для медичних застосувань включав порівняльний аналіз різних протоколів (MQTT, CoAP, HTTP) з точки зору надійності, затримки, енергоефективності та безпеки, огляд технологій бездротового зв'язку (Bluetooth Low Energy (BLE), Wi-Fi, LoRaWAN, NB-IoT) та їхньої придатності для передачі різних типів біомедичних сигналів. Вибір оптимального протоколу та технології бездротового зв'язку для передачі біомедичних даних в IoT-системах залежить від специфічних вимог конкретного сценарію медичного моніторингу, включаючи тип даних, частоту передачі, вимоги до надійності, затримки, енергоефективності та радіус дії.

Для сценаріїв, що вимагають передачі невеликих обсягів даних з низькою частотою, таких як моніторинг температури тіла, рівня кисню в крові (SpO2) або частоти серцевих скорочень у стаціонарних умовах або вдома, BLE є енергоефективним та достатньо надійним варіантом для зв'язку з локальним хабом або смартфоном. Протокол MQTT є легким та добре підходить для передачі телеметричних даних до хмарних платформ з низьким енергоспоживанням.

У випадках, коли потрібна передача більших обсягів даних або потокових даних (наприклад, ЕКГ, ЕЕГ) у межах лікарняної мережі або в умовах наявності стабільного інтернет-з'єднання, Wi-Fi забезпечує високу пропускну здатність та низьку затримку. Протокол HTTP/HTTPS є загальноприйнятим для передачі веб-орієнтованих даних та інтеграції з існуючими IT-інфраструктурами.

Для сценаріїв дистанційного моніторингу на значних відстанях або в умовах слабкого покриття традиційних мереж, технології LoRaWAN та NB-IoT є перспективними завдяки їхній великій дальності зв'язку та низькому енергоспоживанню. Протокол CoAP, розроблений для ресурс-обмежених пристроїв, може бути ефективним вибором для передачі даних через ці мережі. Однак слід враховувати їхню відносно низьку пропускну здатність та можливі затримки.

Для критично важливих застосувань, де потрібна висока надійність та низька затримка (наприклад, системи дистанційного керування медичними пристроями або системи тривожного сповіщення), слід надавати перевагу протоколам та технологіям з гарантованою якістю обслуговування (QoS) та резервними каналами зв'язку, якщо це можливо.

Питання безпеки та конфіденційності передачі біомедичних даних в IoT-системах розглянуто з урахуванням аналізу потенційних загроз безпеці (перехоплення, модифікація даних, кібератаки на пристрої), вибору методів шифрування даних (на пристрої, під час передачі, на сервері), а також вибору механізмів аутентифікації та авторизації користувачів і пристроїв. Авторами піднімається питання відповідності нормативним вимогам щодо захисту персональних медичних даних (наприклад, GDPR, HIPAA).

Практичне значення роботи полягає у впровадженні отриманих результатів у розробку і вдосконалення IoT-систем для медичних цілей, що сприятиме поліпшенню якості телемедичного моніторингу та підвищенню ефективності управління здоров'ям пацієнтів. Практичні аспекти розробки та впровадження IoT-систем для телемедичного моніторингу можливі за описом архітектури типової IoT-системи для моніторингу біомедичних сигналів, оскільки розглянуто питання інтероперабельності з існуючими медичними інформаційними системами та здійснено аналіз вимог до енергоефективності та автономності пристроїв.

Питання зручності використання та прийнятності систем пацієнтами та медичним персоналом і економічні аспекти впровадження IoT-систем у медицині можливі при впровадженні в реальні умови використання таких систем, що буде здійснено в найближчий час.

Список використаних джерел:

1. IoT у медицині: від теорії до реальних кейсів – Режим доступу до ресурсу: https://hub.kyivstar.ua/articles/iot-u-medyczyni-vid-teoriyi-do-realnyh-kejsiv

2. IoT Privacy and Security: Challenges and Solutions – Режим доступу до ресурсу: https://www.mdpi.com/2076-3417/10/12/4102/pdf 7.

3. 7 design principles for IoT – Режим доступу до ресурсу: https://futurice.com/blog/7-design-principles-for-iot

4. Bluetooth low energy possibilities in healthcare [Електронний ресурс] – Режим доступу: https://www.electronicspecifier.com/news/blog/bluetooth-low-energy-possibilities-in-healthcare

УДК 623.48

Мілевський О.В., аспірант
Манойлов В.П., д.т.н., професор
Воротніков В.В., д.т.н., доцент
Державний університет «Житомирська політехніка»

ОБМЕЖЕННЯ ТА ПЕРСПЕКТИВИ ВПРОВАДЖЕННЯ PDM НА БАЗІ МІКРОКОНТРОЛЕРІВ

Впровадження складних систем діагностики обладнання вимагає від компаній чіткого визначення сфери їх застосування, оскільки висока вартість і технічна складність значно обмежують потенційні галузі використання. Це твердження цілком справедливе й для методу прогнозування стану (Predictive Maintenance, PdM). Наприклад, для ефективної роботи PdM часто потрібна надзвичайно висока частота збору даних, порядку сотень кілогерц. Лише за один оберт такого промислового обладнання, як вентилятор або редуктор, обсяг зібраних даних може сягати сотень тисяч одиниць. Збереження та обробка таких значних масивів інформації висуває серйозні вимоги до обчислювальних ресурсів і швидкодії процесорів. Звісно, для сучасних потужних настільних комп'ютерів обробка таких обсягів даних не становить значної проблеми. Однак, інтеграція методів PdM безпосередньо в промислове обладнання часто передбачає використання мікроконтролерів, які мають суттєво обмежені обчислювальні можливості та обсяги пам'яті. Використання ж потужних програмованих логічних контролерів (ПЛК) і розробка спеціалізованого програмного забезпечення для PdM під конкретну машину вимагає від програмістів глибоких знань у цій вузькій галузі, що також сповільнює поширення PdM. Саме тому, з метою здешевлення систем прогнозування стану, інженери пішли шляхом розробки інтелектуальних датчиків на основі MEMS (мікроелектромеханічних систем), які включають в себе вбудований мікроконтролер. Ідея полягає в тому, щоб первинна цифрова обробка сигналу виконувалася безпосередньо на рівні датчика. Для реалізації складних обчислень в умовах обмежених ресурсів датчика виникла нагальна потреба у вдосконаленні існуючих алгоритмів обробки сигналів. Найбільш поширеними алгоритмами, які традиційно застосовуються в методах прогнозування стану, є швидке перетворення Фур'є (Fast Fourier Transform, FFT), синхронне усереднення в часовій області (Time Synchronous зарекомендували себе в системах PdM, але їх обчислювальна інтенсивність вимагає використання потужних обчислювальних пристроїв. У процесі розробки датчиків з інтегрованими алгоритмами PdM робляться спроби оптимізувати вищезгадані алгоритми шляхом застосування технік пошуку в таблицях (look-up tables) та алгоритму Кленшоу (Clenshaw algorithm). Наприклад, застосування цих оптимізацій дозволило значно пришвидшити виконання реального FFT – до 14 разів порівняно зі стандартним алгоритмом FFT. Завдяки таким оптимізаціям стало можливим виконання обчислень алгоритму FFT безпосередньо на датчику, використовуючи мікроконтролер AVR32UC30512C. Щодо алгоритму BEA, результати його оптимізованої реалізації також показали значне прискорення – у 8 разів порівняно зі стандартним алгоритмом. Однак, цього рівня прискорення виявилося недостатньо для широкого застосування відносно недорогих мікроконтролерів у системах вібродіагностики. Як наслідок, датчики вібродіагностики й досі залишаються у високому ціновому сегменті, що стримує їхнє масове використання в системах прогнозування стану обладнання. Таким чином, поширення систем раннього виявлення несправностей дотепер не набуло очікуваного широкого розгортання. Основна причина полягає в тому, що обчислювальної потужності дешевих лінійок мікропроцесорів недостатньо для ефективної обробки складних алгоритмів PdM, а використання потужних процесорів значно збільшує загальну вартість системи.

Отже, для подолання цього бар'єру необхідно продовжувати дослідження у двох ключових напрямках: вдосконалення існуючих алгоритмів прогнозування стану з метою зниження їхньої обчислювальної складності та розробка недорогих інтелектуальних датчиків, здатних виконувати значну частину обробки даних на борту. Паралельно з цим, важливим є створення економічно ефективних систем оповіщення про виявлені несправності та зберігання діагностичних даних, що дозволить користувачам своєчасно реагувати на потенційні проблеми та запобігати серйозним поломкам обладнання. Подальший прогрес у напрямку зазначених досліджень відкриє шлях до ширшого впровадження систем прогнозованого стану, що призведе до значного підвищення надійності та ефективності роботи промислового обладнання.

Список використаних джерел:

1. Засоби організації системи технічного обслуговування [Електронний ресурс]. – Режим доступу: https://www.researchgate.net/publication/356887866_Tehnologii_ta_zasobi_organizacii_sistemi_tehnicnogo_obslugovuvanna

УДК 004

Федорович Д.М., студент, гр. ІПЗ-21-3, ФІКТ
Кушнір Н.О., ст. викл.
Лисогор Ю.І., ст. викл.
Державний університет «Житомирська політехніка»

ЗАСТОСУНОК ДЛЯ ПРОДАЖУ АВТОМОБІЛІВ

У звичайних ситуаціях купівля та продаж автомобілів вимагають ефективного інструменту, який поєднує зручність, безпеку та швидкість обміну інформацією. У процесі проєктування були оглянуті та проаналізовані існуючі аналоги програмного продукту. Найбільш впізнаваним сервісом в Україні є **AUTO.RIA [1]** – платформа, добре знайома користувачам, які прагнуть купити або продати авто. Її перевагами є наявність широкої бази оголошень, потужних фільтрів пошуку, перевірки VIN-коду, зручний інтерфейс та висока довіра з боку користувачів. Однак варто врахувати, що **AUTO.RIA є платним сервісом**, що обмежує можливості безкоштовного розміщення оголошень, особливо для пересічного користувача.

На відміну від **AUTO.RIA**, розроблений застосунок є **повністю безкоштовним** і пропонує простий та інтуїтивно зрозумілий інтерфейс, надійні механізми перевірки оголошень та інструменти для безпосереднього спілкування між покупцями та продавцями. Це робить платформу доступною для кожного, забезпечуючи прозорість та безпеку на всіх етапах купівлі-продажу авто. Регістрація та авторизація дозволяють отримати доступ до особистого кабінету. Реалізовано можливість перегляду каталогу автомобілів, використовуючи фільтрацію за маркою, роком випуску, станом та ціновим діапазоном. Потрібні пропозиції можна знайти за допомогою розширеного пошуку. Для продавців доступний функціонал розміщення оголошень через сервіс Cloudinary, який дозволяє безкоштовно завантажувати до трьох фотографій. Також реалізовано низку додаткових послуг, які пропонує розроблена система, наприклад пріоритетне просування оголошень за додаткову плату. Модератори проводять верифікацію автомобілів, щоби переконатися, що інформація є достовірною. Користувачі отримують сповіщення про статус верифікації через сервіс EmailJS. Крім того, за допомогою VIN-коду система автоматично визначає марку та модель транспортного засобу. Учасники платформи можуть спілкуватися через систему чату, яка дозволяє продавцям і покупцям обговорювати деталі угоди. У спеціальному розділі можливо віднайти новини, пов’язані з автомобілями. В особистому кабінеті користувача відображається історія розміщених оголошень, а також дані про їхній статус.

Архітектура застосунку базується на моделі клієнт-сервер. Серверна частина, розроблена на платформі Node.js, забезпечує високу продуктивність обробки запитів і асинхронне виконання операцій. Бібліотека React використовується для реалізації клієнтської частини, що дозволило спроєктувати динамічний інтерфейс з високою швидкодією. Інтерфейс можна адаптувати до різних пристроїв і платформ завдяки використанню сучасних інструментів розробки. MongoDB є основною системою зберігання даних через свою гнучкість у роботі з напівструктурованими даними, можливість горизонтального масштабування та високу продуктивність при операціях читання та запису. Це особливо важливо для систем, у яких структура даних про автомобілі змінюється. У базі даних основними є колекції User **та Cars,** які пов’язані між собою логічною структурою**.** Колекція User зберігає інформацію про користувачів: електронну пошту, ім’я, пароль та рівень доступу (наприклад, звичайний користувач або адміністратор). Головне призначення цієї колекції – забезпечення автентифікації, реєстрації користувачів та управління їхніми правами доступу. Колекція **Cars** описує автомобілі, які користувачі можуть додавати до системи. Тут зберігаються тип транспорту, назва, деталі, фото, VIN-код, бренд, модель, статус використання (наприклад, новий або вживаний), перевірений стан (верифіковане оголошення або ні), ціна, дата розміщення й закінчення оголошення.

Розроблений додаток проходить як функціональне, так і нефункціональне тестування. Це забезпечить відповідність програмного забезпечення всім вимогам та стандартам, що є ключовим аспектом у процесі розробки.

Список використаних джерел:

1. AUTO.RIA [Електронний ресурс] – Режим доступу до ресурсу: https://auto.ria.com/uk/
2. Документація Node.js [Електронний ресурс] – Режим доступу до ресурсу: https://nodejs.org/en/docs/
3. Документація React [Електронний ресурс] – Режим доступу до ресурсу: https://react.dev/learn
4. Документація MongoDB [Електронний ресурс] – Режим доступу до ресурсу: https://www.mongodb.com/docs/

УДК 004

Талько П.В., здобувач
Граф М.С., Ph.D., доцент
Державний університет «Житомирська політехніка»

АНАЛІЗ ГНУЧКИХ МЕТОДОЛОГІЙ УПРАВЛІННЯ ІТ-ПРОЄКТАМИ: AGILE, SCRUM, KANBAN

У сучасних умовах стрімкого розвитку інформаційних технологій та постійних змін у вимогах замовників особливої актуальності набувають гнучкі методології управління проєктами. Одними з найпоширеніших таких підходів є Agile, Scrum та Kanban.

В одній із своїх робіт Andre Caron Zanezi сказав: «Agile є реалістичним і адаптивним підходом до методів управління проектами широко зайняті в кількох секторах бізнесу. В епоху швидкої трансформації Agile розвиток відомий тим, що має справу зі складними проблемами та швидко адаптується до бізнесу» [1]. Primadhika Marnada говорив: «Kanban є гнучким, оскільки він зосереджується на важливих завданнях проекту та ставить їх у пріоритет» [2].

Метою даного дослідження є порівняльний аналіз основних принципів, переваг, обмежень та особливостей застосування Agile, Scrum і Kanban у сфері управління ІТ-проєктами.

Кожна з методологій має свої унікальні особливості:

- Agile — це парадигма, що базується на ітеративному підході, взаємодії з клієнтом і готовності до змін.
- Scrum — одна з реалізацій Agile, яка включає чітко структуровані ролі, церемонії та артефакти.
- Kanban — візуальна система управління завданнями, що сприяє безперервному покращенню процесу та контролю над робочим навантаженням.

Основні аспекти аналізу включають:

- Принципи та філософію кожної методології.
- Ролі та обов’язки в команді.
- Механізми планування, пріоритезації та звітності.
- Сценарії ефективного застосування в ІТ-проєктах.
- Інтерпретація та звітність.

Порівняльна характеристика гнучких методологій подана у таблиці 1.

Таблиця 1

Порівняльна таблиця

Критерії	**Методологія Agile**	**Методологія Scrum**	**Методологія 1Kanban**
Структура	Гнучкаметодолгія	Чітко визначена структура з ролями і спринтами	Візуалізація процесу, немає жорстких правил
Управління завданнями	Ітеративне	Сплановані ітерації (спринти)	Безперервний потік завдань
Пріоритетність	Динамічна	Визначається перед кожним спринтом	Завдання впорядковуються за важливістю
Гнучкість до змін	Висока	Помірна (зміни вносяться між спринтами)	Дуже висока — зміни можливі в будь-який момент
Сферазастосування	Універсальна	Найчастіше для командної роботи в ІТ	Підходить для підтримки та операційної діяльності

В роботі проведено аналіз гнучких методології управління ІТ-проєктами. Такі методології стали стандартом в ІТ-сфері завдяки своїй адаптивності, прозорості та орієнтації на результат. Вибір між проаналізованими Agile, Scrum і Kanban залежить від розміру команди, типу проєкту, швидкості змін вимог і потреб замовника. Правильне впровадження обраної методології дозволяє суттєво підвищити ефективність проєктної діяльності, зменшити ризики та забезпечити якісний кінцевий результат.

Список використаних джерел:

1. Zanezi, A.C. and Carvalho, M.M. (2022), “How project management principles affect Lean Six Sigma program and projects: a systematic literature review”
2. Marnada, P., Raharjo, T., Hardian, B. and Prasetyo, A. (2022), “Agile project management challenge in handling scope and change: A systematic literature review