№ 4(58)
2026
НАУКА і ТЕХНІКА СЬОГОДНІ
ISSN 2786-6025 Online
УДК 004.89:004.82

https://doi.org/10.52058/2786-6025-2026-4(58)-4132-4142

Перепелиця В'ячеслав Ігорович доктор філософії, асистент кафедри комп'ютерних наук, Вінницький національний технічний університет, м. Вінниця, https://orcid.org/0009-0003-5246-750X

Мунтян Дмитро Віталійович бакалавр 4-го курсу за спеціальністю 122 «Комп'ютерні науки», Вінницький національний технічний університет, м. Вінниця, https://orcid.org/0009-0004-2288-5644

Петришин Сергій Іванович кандидат технічних наук, старший викладач кафедри комп'ютерних наук, Вінницький національний технічний університет, м. Вінниця, https://orcid.org/0009-0001-3465-1499

СЕМАНТИЧНА МОДЕЛЬ СИСТЕМИ ОБЛІКУ РОБОЧОГО ЧАСУ ПРАЦІВНИКІВ НА ОСНОВІ ГЕНЕРАТИВНОГО ШТУЧНОГО ІНТЕЛЕКТУ

Анотація. У статті розглядається актуальна проблема корпоративного менеджменту — неефективність традиційних систем обліку робочого часу працівників. Класичні трекери базуються виключно на кількісних метриках та ігнорують якісний контекст завдань. Це призводить до інформаційного перевантаження керівників, хибного зниження показників продуктивності та неможливості об'єктивно оцінити причини затримок. Для вирішення цих недоліків запропоновано та архітектурно обґрунтовано інноваційну модель інтелектуальної системи обліку робочого часу на основі генеративного штучного інтелекту (GenAI). Головною відмінністю програмного рішення є концептуальний перехід від механічного підрахунку відпрацьованих годин до глибокого семантичного аудиту продуктивності. Архітектура системи спирається на передовий технологічний стек. Клієнтська частина спроєктована за допомогою React, де застосування Redux Toolkit вирішує проблему розсинхронізації таймерів у неактивних вкладках браузера завдяки управлінню станом через абсолютні часові мітки. Масштабована серверна інфраструктура реалізована на базі фреймворку NestJS із застосуванням реляційної бази PostgreSQL (TypeORM), що гарантує високу відмовостійкість, надійну ізоляцію бізнес-логіки та цілісність даних при конкурентних запитах. Ключовим елементом наукової новизни є розроблений підхід до інтеграції великих мовних моделей (LLM). Представлено алгоритм агрегації атомарних логів користувача

ISSN 2786-6025 Online

в оптимізований текстовий потік. Інтегрований ШІ-модуль функціонує як неупереджений віртуальний аналітик: він досліджує текстові нотатки, виявляє приховані системні перешкоди (bottlenecks) в інфраструктурі проєктів та проактивно сигналізує про ранні ризики професійного вигорання. Запропонована модель трансформує моніторинг часу з інструменту нагляду на засіб захисту працівників та стратегічної оптимізації процесів.

Ключові слова: облік робочого часу, штучний інтелект, великі мовні моделі, React, Redux Toolkit, NestJS, TypeScript.

Perepelytsia Viacheslav PhD, Assistant of the Computer Science Department, Vinnytsia National Technical University, Vinnytsia, https://orcid.org/0009-0003-5246-750X

Muntian Dmytro 4th-year bachelor student majoring in 122 "Computer Science", Vinnytsia National Technical University, Vinnytsia, https://orcid.org/0009-0004-2288-5644

Petrishyn Sergiy Cand.Sc.(Eng.), Senior Lecturer, Department of Computer Science, Vinnytsia National Technical University, Vinnytsia, https://orcid.org/0009-0001-3465-1499

SEMANTIC MODEL OF EMPLOYEE WORKING TIME ACCOUNTING SYSTEM BASED ON GENERATIVE ARTIFICIAL INTELLIGENCE

Abstract. The article considers a pressing problem of corporate management - the inefficiency of traditional systems for recording employee working hours. Classic trackers are based exclusively on quantitative metrics and ignore the qualitative context of tasks. This leads to information overload for managers, a false decrease in productivity indicators, and the inability to objectively assess the causes of delays. To address these shortcomings, an innovative model of an intelligent working time recording system based on generative artificial intelligence (GenAI) is proposed and architecturally substantiated. The main difference of the software solution is the conceptual transition from mechanical counting of hours worked to a deep semantic audit of productivity. The system architecture is based on an advanced technology stack. The client part is designed using React, where the use of Redux Toolkit solves the problem of timer desynchronization in inactive browser tabs due to state management through absolute time stamps. The scalable server infrastructure is implemented on the basis of the NestJS framework using the PostgreSQL relational database (TypeORM), which guarantees high fault tolerance, reliable isolation of business logic and data integrity during competitive queries. The key element of

ISSN 2786-6025 Online
scientific novelty is the developed approach to the integration of large language models (LLM). An algorithm for aggregating atomic user logs into an optimized text stream is presented. The integrated AI module functions as an unbiased virtual analyst: it examines text notes, detects hidden system bottlenecks (bottlenecks) in the project infrastructure and proactively signals early risks of professional burnout. The proposed model transforms time monitoring from a surveillance tool to a means of employee protection and strategic process optimization.

Keywords: time tracking, artificial intelligence, large language models, React, Redux Toolkit, NestJS, TypeScript.

Постановка проблеми. Традиційні підходи [1, 2] до контролю трудової дисципліни спираються виключно на метрики часу: тривалість сесій або загальну кількість годин на робочому місці. Сучасні системи обліку успішно автоматизували збір цих даних, проте створили нову проблему — інформаційне перевантаження менеджменту. Керівники отримують тисячі рядків логів, з яких неможливо зробити швидкий висновок про реальну ефективність.

Головним недоліком існуючих рішень є їхня «сліпота» до контексту: для класичного трекера 5 годин, витрачених на рутинну правку коду, і 5 годин складного архітектурного дослідження виглядають однаково. Актуальним завданням є створення системи, яка діє як віртуальний аналітик: не просто агрегує секунди, а розуміє семантику витраченого часу. Така система має вирішити наступні проблеми:

– відсутність інструментів для виявлення системних перешкод у робочому процесі.

– високе навантаження на менеджмент при аналізі великих обсягів текстових нотаток розробників.

– потреба у проактивній оцінці стану працівника (залученість, втома, втрата фокусу) на основі його щоденної активності.

Аналіз останніх досліджень і публікацій. Питання автоматизації обліку часу та проектування складних інформаційно-аналітичних систем активно вивчається у вітчизняному та зарубіжному науковому просторі. Алгоритмічні засоби моніторингу активності працівників за комп'ютером, що базуються на фіксації подій введення-виведення, детально описано у дослідженні [1]. Фундаментальні основи моделювання інформаційно-аналітичних систем обліку робочого часу співробітників представлені у праці [2], де акцент зроблено на створенні гнучких архітектур для багатопараметричної звітності.

Сучасна парадигма розробки наголошує, що контейнеризація та використання хмарних платформ є ключовими факторами ефективної реалізації таких систем. Практичні аспекти впровадження автоматизованих систем аудиту та умови їх доцільності наведено у роботі [3]. Дослідження [4, 5] фокусуються

ISSN 2786-6025 Online

на специфіці розгортання клієнт-серверних рішень безпосередньо в інфраструктурі підприємств.

Водночас, ефективність управління станом клієнтської частини у сучасних веб-застосунках за допомогою екосистеми React (зокрема Redux Toolkit) підтверджується останніми порівняльними дослідженнями продуктивності [6]. Що стосується впливу штучного інтелекту на продуктивність, звіти провідних дослідницьких центрів доводять значне підвищення ефективності при використанні генеративного ШІ та великих мовних моделей (LLM) у реальних робочих процесах [7].

Останні розробки у сфері аналізу логів (наприклад, фреймворк LogLLM) демонструють високу ефективність архітектур на базі LLM у виявленні аномалій шляхом семантичного аналізу записів [8].

Проте алгоритми машинного навчання у таких системах досі використовувалися переважно для біометричної ідентифікації працівників. Інтеграція генеративного штучного інтелекту (GenAI) для семантичного розуміння текстових нотаток розробників та автоматичного зіставлення планових годин із фактичними є новим вектором досліджень, що робить розробку таких моделей інноваційною.

Мета статті. Розробка та архітектурне обґрунтування моделі інтелектуальної системи обліку робочого часу, яка ефективно поєднує можливості клієнтської оптимізації (React, Redux Toolkit) зі стабільним бекендом (NestJS), де ключовою інноваційною перевагою є впровадження модуля ШІ для семантичного аудиту.

Виклад основного матеріалу. Сучасна парадигма розробки корпоративного програмного забезпечення вимагає поступового відходу від жорстко детермінованих алгоритмів на користь адаптивних систем.

На відміну від аналогів, де аналітика будується на статичних математичних формулах та тригерах, розроблений застосунок використовує архітектуру AI-First.

Це означає, що система не просто збирає дані для подальшого відображення, а формує інформаційний фундамент для безперервного машинного аналізу.

Весь технологічний стек спроєктовано з метою максимально чистого, надійного та швидкого збору контексту для його подальшої передачі у велику мовну модель (LLM).

Вибір інструментарію базувався на критеріях відмовостійкості, строгої типізації та можливості горизонтального масштабування. Схема взаємодії клієнт-серверної архітектури інтелектуальної системи обліку робочого часу працівників показана на рисунку 1.

ISSN 2786-6025 Online

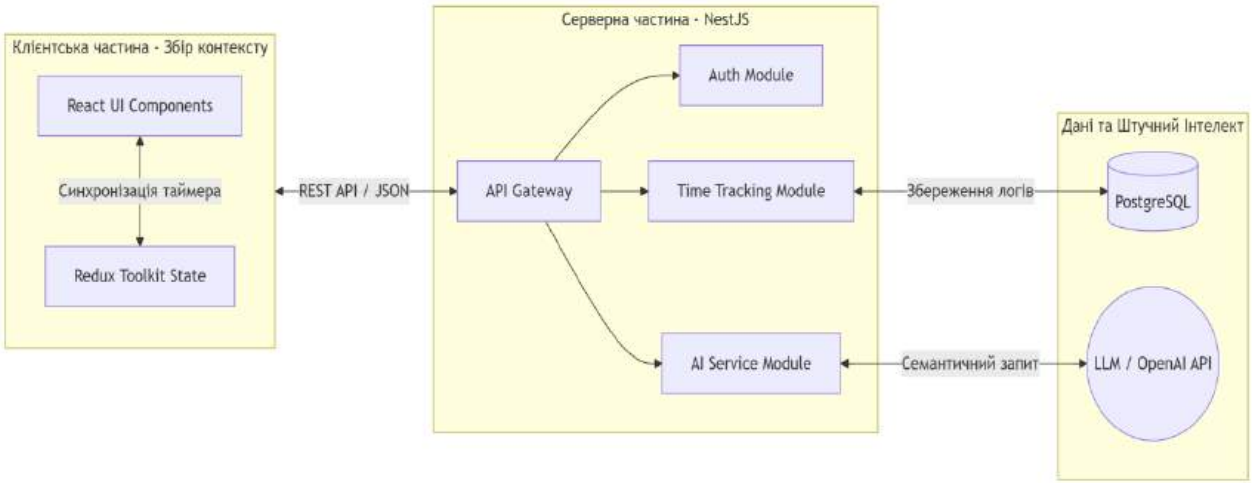


Рисунок 1 – Схема взаємодії клієнт-серверної архітектури інтелектуальної системи обліку робочого часу працівників

Фронтенд-частина системи побудована на декларативному підході до створення інтерфейсів із застосуванням React та Redux Toolkit. Віртуальний DOM (Virtual DOM) бібліотеки React забезпечує миттєвий відгук інтерфейсу навіть при складних візуалізаціях даних. Своєю чергою, Redux Toolkit гарантує точну синхронізацію часу без паразитного ререндерингу всього дерева компонентів, що критично важливо для тайм-трекерів, які працюють у фонових вкладках браузера.

Для серверної частини доцільно обрати фреймворк NestJS, архітектура якого похідна від Angular. Використання патерну впровадження залежностей (Dependency Injection) дозволяє чітко розмежувати бізнес-логіку збору логів, автентифікації та обробки асинхронних ШІ-запитів, мінімізуючи зв'язність коду (coupling).

Реляційна структура бази даних PostgreSQL та TypeORM – відповідає стандартам ACID, надійно зберігає складні ієрархічні зв’язки між працівниками, завданнями та згенерованими ШІ-звітами, гарантуючи цілісність даних при конкурентних запитах.

Для наочного розуміння фундаментальних переваг розробленої системи наведено порівняльний аналіз традиційних та інтелектуальних підходів (таб. 1).

ISSN 2786-6025 Online

Таблиця 1

Порівняльний аналіз традиційної системи обліку часу із інтелектуальною системою з AI-модулем

Критерій оцінки	***Традиційні системи (Без ШІ)***	***Інтелектуальна система з ШІ-модулем***
Характер даних	Кількісний (години, хвилини)	Кількісний + Семантичний (розуміння суті роботи)
Роль менеджера	Ручна вичитка кожного логу та зведення таблиць	Отримання готових висновків та інсайтів (High-level)
Виявлення вигорання	Неможливе на ранніх етапах	Проактивне (виявлення частого перемикання контексту)
Оцінка естимацій	Математичне відхилення (План/Факт)	Аналіз причин відхилення на основі нотаток

У системах обліку часу критично важливо забезпечити безперебійну роботу таймера на стороні клієнта. Головна складність полягає в тому, що сучасні браузери часто призупиняють або обмежують виконання JavaScript-коду (throttling) у неактивних вкладках для економії ресурсів процесора.

Redux Toolkit вирішує цю проблему шляхом створення ізольованих «зрізів» (slices) стану та роботи з абсолютними часовими мітками замість відносних лічильників. Це дозволяє уникнути розсинхронізації. Нижче наведено фрагмент програмної реалізації управління станом сесії на стороні клієнта за допомогою TypeScript.

```
import { createSlice, PayloadAction } from '@reduxjs/toolkit';
interface TrackingState {
 isActive: boolean;
 currentTaskId: string | null;
 startTime: number | null;
}
const initialState: TrackingState = {
 isActive: false,
 currentTaskId: null,
 startTime: null,
};
const trackingSlice = createSlice({
 name: 'tracking',
 initialState,
 reducers: {
 startTracking: (state, action: PayloadAction<string>) => {
```

ISSN 2786-6025 Online

```
state.isActive = true;
state.currentTaskId = action.payload;
state.startTime = Date.now();
},
stopTracking: (state) => {
state.isActive = false;
state.currentTaskId = null;
state.startTime = null;
}
}
});
export const { startTracking, stopTracking } = trackingSlice.actions;
export default trackingSlice.reducer;
```

Цей механізм формує єдине джерело істини для всього застосунку. Він гарантує, що при навігації між сторінками застосунку (наприклад, при переході від дашборда до налаштувань профілю) таймер не втрачає свій стан, а дані надійно утримуються в оперативній пам'яті клієнта до моменту успішної транзакції на сервер. Структура ключових UI-компонентів наведена в таблиці 2.

Таблиця 2

Функціональне призначення UI-компонентів клієнтської частини

Назва компонента	***Технологія / Патерн***	***Функціональне призначення***
ActiveTimerWidget	React Hooks (useEffect, useState)	Відображення реального часу поточної сесії без ініціювання повного ререндерингу сторінки.
StateStoreManager	Redux Toolkit	Глобальне управління даними профілю працівника, токенами авторизації та статусом виконання завдань.
AI_InsightsPanel	React.memo / Material UI	Відображення структурованого семантичного звіту від модуля ШІ. Оптимізовано для уникнення зайвих перемальовувань.

Основна логіка системи зосереджена в інфраструктурі NestJS, що дозволяє обробляти тисячі конкурентних запитів від клієнтів. Проте саме модуль штучного інтелекту є центральним елементом наукової та практичної новизни системи. Його перевага наочно демонструється при виникненні операційних аномалій під час розробки програмного забезпечення. Наприклад, якщо розробник витрачає 8 годин на завдання, попередньо оцінене в 2 години:

ISSN 2786-6025 Online

– класичний трекер механічно зафіксує перевитрату часу, що автоматично знизить рейтинг ефективності (KPI) і може призвести до хибних висновків керівництва щодо некомпетентності працівника;

– інтелектуальний AI-модуль глибоко проаналізує текстові коментарі до мікросесій (наприклад, "очікування відповіді від DevOps", "довгий білд через конфлікт залежностей", "недоступність тестового сервера") і у своєму звіті вкаже керівнику на "наявність системної проблеми в інфраструктурі проєкту", а не на особисту провину співробітника.

Така модель кардинально змінює роль системи моніторингу, адже вона стає інструментом захисту працівника та оптимізації процесів (рис. 2).

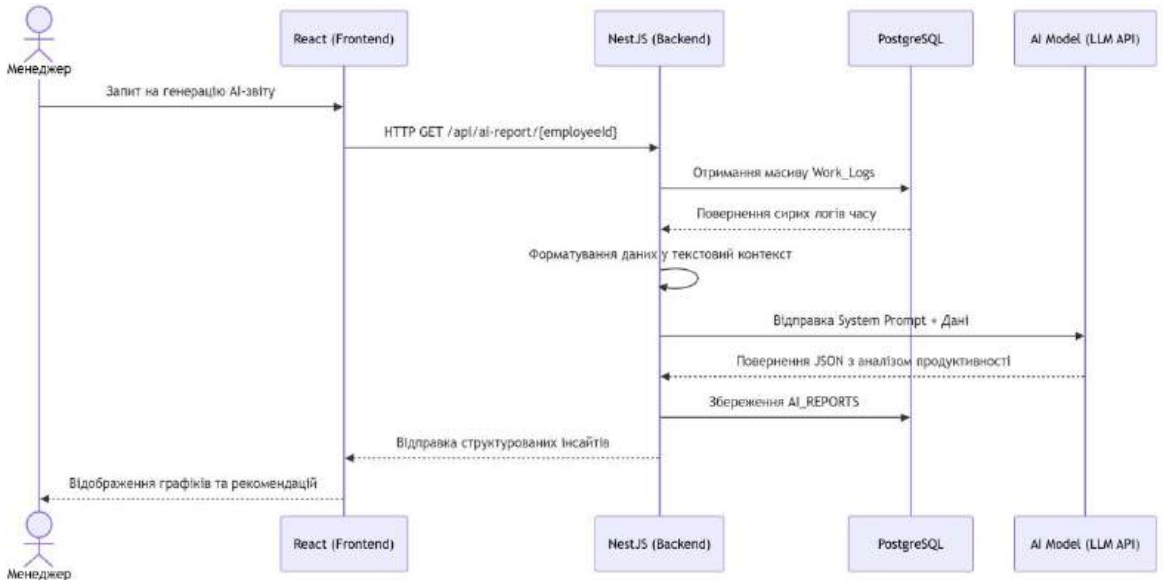


Рисунок 2 – Модель інформаційної взаємодії компонентів системи під час генерації ШІ-звітів

Нижче наведено фрагмент програмної реалізації сервісу мовою TypeScript, який форматує дані, оптимізує використання токенів та звертається до API великої мовної моделі.

```
import { Injectable, Logger } from '@nestjs/common';
import { OpenAI } from 'openai';
import { LogEntryDto } from './dto/log-entry.dto';
@Injectable()
export class AiService {
 private readonly logger = new Logger(AiService.name);
 private openai: OpenAI;
 constructor() {
 this.openai = new OpenAI({ apiKey: process.env.AI_API_KEY });
 }
```

ISSN 2786-6025 Online

```
async analyzeWorkPeriod(employeeId: string, logs: LogEntryDto[]): Promise<any> {
const context = this.prepareContext(logs);
try {
const response = await this.openai.chat.completions.create({
model: 'gpt-4o',
messages: [
{
role: 'system',
content: 'You are an expert HR and Project Management AI. Your goal is to evaluate true productivity, find hidden bottlenecks in the workflow, and assess burnout risks based on the provided time logs.'
},
{
role: 'user',
content: `Analyze the semantic value of these work logs for employee ${employeeId}: ${context}`
}
],
response_format: { type: 'json_object' }
});
return JSON.parse(response.choices[0].message.content || '{}');
} catch (error) {
this.logger.error('AI Analysis failed', error);
throw new Error('Analysis Service Unavailable');
}
}
private prepareContext(logs: LogEntryDto[]): string {
return logs.map(l =>
`Task: ${l.title}; Time spent: ${l.duration}min; Developer Note: ${l.description}`
).join(' | ');
}
}
```

Метод prepareContext виконує важливу функцію попередньої обробки даних – він агрегує розрізнені атомарні записи активності в єдиний, оптимізований текстовий потік, дозволяючи мовній моделі бачити цілісну картину робочого дня в межах обмеженого контекстного вікна. Системний промпт (інструкція для моделі) жорстко задає ШІ роль неупередженого аналітика з пошуку «вузьких місць» та оцінки ймовірності професійного вигорання.

Формат json_object гарантує отримання суворо структурованої відповіді від мовної моделі, яка без додаткового складного парсингу зберігається у базі даних та відображається у вигляді графіків на клієнтському дашборді.

Висновки. Розроблена модель інтелектуальної системи обліку робочого часу працівників демонструє синергію сучасних фреймворків розробки та

ISSN 2786-6025 Online

технологій штучного інтелекту. Використання React у поєднанні з Redux Toolkit забезпечило створення високопродуктивного та реактивного інтерфейсу, здатного витримувати інтенсивні оновлення стану таймера. Архітектура NestJS дозволила побудувати надійний та легко масштабований бекенд. Інтеграція підсистеми ШІ переводить процес обліку часу на рівень стратегічного аналізу, що дозволяє компаніям не лише фіксувати відпрацьовані години, а й проактивно впливати на продуктивність та запобігати професійному вигоранню співробітників.

ЛІТЕРАТУРА

1. Кіндій В. А., Парамуд Я. С. Алгоритмічно-програмні засоби моніторингу часу роботи працівника за комп'ютером. *Computer Systems and Networks*. 2022. Vol. 4, No. 1. С. 59-66. https://science.lpnu.ua/sites/default/files/journal-paper/2022/dec/29472/zhurnal-ksm41verstka1-61-68.pdf

2. Чубаєвський В. І., Рзаєва С. Л., Рзаєв Д. О. Моделювання інформаційно-аналітичної системи обліку робочого часу співробітників підприємства. *Технічні науки та технології*. 2019. № 2 (16). С. 79–85. https://ir.kneu.edu.ua/items/19fbdf4c-1d5a-4aae-9fc9-b962934d1602

3. Бєлий Д. С. Автоматизована система обліку та аудиту робочого часу працівників. Миколаїв : ЧНУ ім. П. Могили, 2022. 24 с. https://krs.chmnu.edu.ua/jspui/handle/123456789/332

4. Кулагін С. Автоматизована система обліку робочого часу працівників на підприємстві. *Інформаційні технології в науці, виробництві та підприємництві* : матеріали конф. Київ : КНУТД, 2019. С. 15-18. https://ela.kpi.ua/handle/123456789/57678

5. Вайман С. М. Аналіз та впровадження системи обліку робочого часу. Тернопіль : ТНТУ ім. І. Пулюя, 2019. 85 с. http://elartu.tntu.edu.ua/handle/lib/31377

6. Bello, S., Bello, S. React Query vs. SWR vs. Redux Toolkit Query: Choosing the Right Tool for Your Project. 2025. https://www.researchgate.net/publication/394407975

7. Jaffe S. et all. Generative AI in Real-World Workplaces. Microsoft Research. 2024. https://www.microsoft.com/en-us/research/wp-content/uploads/2024/07/Generative-AI-in-Real-World-Workplaces.pdf

8. Wei Guan et all. LogLLM: Log-based Anomaly Detection Using Large Language Models. 2024. https://arxiv.org/abs/2411.08561

REFERENCES

1. Kindii, V. A., & Paramud, Y. S. (2022). Algorithmic and software tools for monitoring working time of an employee on a computer. *Computer Systems and Networks*, 4(1), 59-66. https://science.lpnu.ua/sites/default/files/journal-paper/2022/dec/29472/zhurnal-ksm41verstka1-61-68.pdf

2. Chubaievskyi, V. I., Rzaieva, S. L., & Rzaiev, D. O. (2019). Modelling of information and analytical accounting system of employees' working time. *Technical Sciences and Technologies*, 2(16), 79-85. https://ir.kneu.edu.ua/items/19fbdf4c-1d5a-4aae-9fc9-b962934d1602

3. Bielyi, D. S. (2022). Automated system for accounting and auditing of employees' working time. Mykolaiv: Petro Mohyla Black Sea National University, 24 p. https://krs.chmnu.edu.ua/jspui/handle/123456789/332

4. Kulahin, S. (2019). Automated system of employee working time accounting at the enterprise. *Information technologies in science, production and entrepreneurship*. Kyiv: KNUTD, 15-18. https://ela.kpi.ua/handle/123456789/57678

ISSN 2786-6025 Online

5. Vaiman, S. M. (2019). Analysis and implementation of a working time accounting system. Ternopil: Ivan Puluj National Technical University, 85 p. http://elartu.tntu.edu.ua/handle/lib/31377

6. Bello, S., Bello, S. React Query vs. SWR vs. Redux Toolkit Query: Choosing the Right Tool for Your Project. 2025. https://www.researchgate.net/publication/394407975

7. Jaffe S. et all. Generative AI in Real-World Workplaces. Microsoft Research. 2024. https://www.microsoft.com/en-us/research/wp-content/uploads/2024/07/Generative-AI-in-Real-World-Workplaces.pdf

8. Wei Guan et all. LogLLM: Log-based Anomaly Detection Using Large Language Models. 2024. https://arxiv.org/abs/2411.08561

Дата першого надходження статті до видання: 12.04.2026
Дата прийняття статті до друку після рецензування: 26.04.2026