

Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»
Кафедра інформатики факультету інформатики

**ДОСЛІДЖЕННЯ МЕТОДІВ ВІЗУАЛІЗАЦІЇ ДАНИХ ПРО
ВІДСТЕЖЕНИЙ ЧАС ДЛЯ САМООРГАНІЗАЦІЇ
СТУДЕНТАМИ ЧАСУ НА НАВЧАННЯ**

**Текстова частина до курсової роботи
за спеціальністю „Інженерія програмного забезпечення” - 121**

Керівник курсової роботи
магістр комп'ютерних наук,
ст. в. Медвідь С. О.

_____ (підпис)
“ ____ ” _____ 2025 р.

Виконала студентка ІІЗ-3

Крижанівська М. С.
“ ____ ” _____ 2025 р.

Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»
Кафедра інформатики факультету інформатики

ЗАТВЕРДЖУЮ

Зав.кафедри інформатики,
доц., к.ф-м.н.

_____ С. С. Гороховський
(підпис)

“ ____ ” _____ 2024 р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ

на курсову роботу

студентці Крижанівській Маргариті Сергіївні факультету інформатики 3-го курсу

ТЕМА: Дослідження методів візуалізації даних про відстежений час для самоорганізації студентами часу на навчання

Зміст ТЧ до курсової роботи:

Зміст

Анотація

Вступ

1 Аналіз наявних рішень для відстеження навчального часу

2 Методи для візуалізації даних

3 Розробка системи Study Time Tracker

Висновки

Список літератури

Додатки (за необхідністю)

Дата видачі “ ____ ” _____ 2024 р. Керівник _____
(підпис)

Завдання отримала _____
(підпис)

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапу курсової роботи	Термін виконання етапу	Примітка
1.	Отримання завдання на курсову роботу.	05.10.2024	
2.	Огляд літератури за темою роботи.	20.10.2024	
3.	Огляд програмних рішень для відстеження навчального часу.	03.11.2024	
3.	Дослідження методів візуалізації даних.	24.11.2024	
4.	Розробка власної системи для відстеження навчального часу.	12.01.2025	
5.	Синтез оптимальних методів візуалізації даних про час та пошук відповідних інструментів.	25.01.2025	
6.	Застосування винайденого синтезу методів візуалізації для відображення аналітики.	09.02.2025	
7.	Реалізація інтеграції з Moodle та використання LLM для аналізу даних і створення звітів.	23.02.2025	
8.	Написання текстової частини роботи.	20.04.2025	
9.	Корегування роботи за результатами перевірки науковим керівником.	27.04.2025	
10.	Остаточне оформлення пояснювальної роботи та створення презентації.	04.05.2025	
11.	Захист курсової роботи.	12.05.2025	

Студент _____

Керівник _____

“ _____ ” _____

ЗМІСТ

Анотація	7
Вступ.....	8
1 Аналіз наявних рішень для відстеження навчального часу	11
1.1 Огляд популярних програмних рішень	11
1.2 Порівняльний аналіз аналогів.....	12
2 Методи для візуалізації даних.....	18
2.1 Основні аспекти при виборі методів візуалізації даних.....	18
2.2 Синтез оптимальних методів для візуалізації навчального процесу	20
2.2.1 Застосування графіків та діаграм	21
2.2.2 Застосування зведених таблиць	24
2.2.3 Застосування календаря.....	26
2.2.4 Застосування карток та списків	28
2.3 Застосування LLM у візуалізації даних	30
2.3.1 Обґрунтування та суть методу	31
2.3.2 Інженерія промпту для отримання якісного звіту	32
2.3.3 Візуалізація звіту	34
2.4 Новизна та цінність використаних підходів	35
3 Розробка системи Study Time Tracker.....	36
3.1 Функціональні можливості програми	37
3.2 Браузерне розширення для інтеграції з Moodle	39
3.2.1 Принцип роботи розширення.....	40
3.2.2 Комунікація розширення та програми. Вебсокети	41
3.2.3 Перспективи для подальших інтеграцій	42

Висновки	43
Список використаної літератури	44

ПЕРЕЛІК ВИКОРИСТАНИХ СКОРОЧЕНЬ І ТЕРМІНОЛОГІЇ

API – Application programming interface: інтерфейс програмування застосунків;

CRUD – create read update delete: чотири базові операції персистентного сховища;

CSS – Cascading Style Sheets: каскадні таблиці стилів;

DistEdu – система організації навчального процесу НаУКМА, побудована на основі платформи Moodle

GPT – Generative Pre-trained Transformer: генеративний попередньо тренований трансформер;

HTML – HyperText Markup Language: мова розмітки гіпертексту;

HTTP – Hypertext Transfer Protocol: протокол передачі гіпертекстових документів;

JSON – JavaScript Object Notation: текстовий формат для запису об'єктів JavaScript;

JWT – JSON Web Token: стандарт токена доступу на основі JSON;

LLM – Large Language Model: велика мовна модель;

Markdown – полегшена мова розмітки даних;

Study Time Tracker – розроблена в рамках дослідження система;

Вебсокет – WebSocket;

Дашборд – dashboard;

Кросплатформність – властивість програми працювати більше ніж на одній платформі або операційній системі;

НаУКМА – Національний Університет «Києво-Могилянська Академія»

Промпт – (англ. prompt) запит чи команда для LLM, яка надає моделі вказівки про те, що користувач хоче отримати у відповідь;

Таймлайн – timeline;

Теплова карта – heatmap;

АНОТАЦІЯ

У даній роботі досліджуються методи візуалізації даних про навчальний час та їх оптимальне поєднання для самоорганізації студентами навчального процесу та аналізу своєї продуктивності. У процесі роботи було проаналізовано наявні програмні рішення, досліджено різні підходи до візуалізації даних, а також розглянуто можливості інтеграції з іншими системами та застосування LLM для генерації звітів про навчальні тенденції користувачів. Результатом дослідження є вебзастосунок Study Time Tracker, який дозволяє студентам фіксувати витрачений на навчання час та аналізувати своє навантаження, навчальні звички та тенденції за допомогою наочної аналітики та статистики.

Ключові слова: візуалізація даних, вебзастосунок, відстеження часу, тайм-трекінг, аналітика, освітні технології, LLM.

ВСТУП

Після початку пандемії COVID-19 великої популярності набула дистанційна форма навчання. Такий формат навчання сприяє розвитку самодисципліни та самостійності, адже відповідальність за організацію свого навчального процесу та його успішність лягає на студентів [1]. Велику роль в цьому процесі відіграють системи та платформи для онлайн навчання, такі як Zoom, Moodle, Office 365, Google Classroom та інші. Ці програми допомагають організовувати синхронні та асинхронні заняття, проте для ефективного навчання та організації свого дня також були б корисними інші інструменти, які дозволятимуть відстежувати витрати часу на навчання та допомагатимуть використовувати його ефективніше.

Тож в рамках цієї роботи було розроблено платформу Study Time Tracker, яка допомагатиме студентам відстежувати свій час та краще організовувати свій навчальний процес. Такий інструмент може бути особливо корисним для студентів, які схильні до прокрастинації та мають труднощі з самодисципліною, адже він допомагатиме вести облік свого часу і надаватиме широку аналітику та поради щодо періодів найбільшої ефективності, взаємозв'язків з оцінками тощо.

Особлива увага була приділена методам візуалізації даних, отриманих в процесі відстежування часу, оскільки вичерпна та наочна аналітика допоможе користувачам краще зрозуміти свої навчальні звички, порівняти продуктивність у різні періоди та виявляти фактори, що впливають на ефективність навчання.

Мета дослідження — пошук та практичне застосування оптимальних методів візуалізації даних про витрачений на навчання час. Для її досягнення були поставлені такі завдання:

1. Проаналізувати інші програмні рішення для відстежування часу

2. Дослідити різні методи для візуалізації даних
3. Визначити найбільш ефективні методи для візуалізації даних про час та його кореляції з іншими показниками
4. Розробити систему для отримання та зберігання даних
5. Впровадити синтез різних методів візуалізації даних для відображення повної та змістовної аналітики

Об'єктом дослідження є процеси самоорганізації студентом свого часу та навчання та методи візуалізації даних, що описують цей процес.

У процесі виконання поставлених завдань було використано декілька **методів дослідження**. Під час дослідження програмних аналогів був застосований метод порівняльного аналізу для знаходження сильних та слабких сторін продуктів. Під час дослідження методів візуалізації даних було проведено аналіз літератури. Після отримання цих даних був застосований метод синтезу, для поєднання найкращих сторін аналогів та методів візуалізації даних у цій роботі. Насамкінець в процесі моделювання було розроблено вебзастосунок, у якому були використані досліджувані інструменти та реалізовані функціональні можливості.

Джерела дослідження включають наукові статті та книжки на тему візуалізації даних та дотичних тем, офіційні документації використаних інструментів та технологій, сайти аналогічних програмних продуктів.

Наукова новизна роботи полягає у новому поєднанні методів візуалізації даних у застосунках для відстеження часу та в застосуванні великих мовних моделей для аналізу цих даних. За допомогою LLM широке поєднання візуальної аналітики у вигляді графіків, діаграм, карток, списків та зведених таблиць доповнюється коротким текстовими викладами про навчальні тенденції користувача, найпродуктивніші години навчання та зміни у його продуктивності, а також висновками з рекомендаціями щодо ефективнішого використання часу. У програмі також передбачена інтеграція з побудованою

на базі Moodle навчальною платформою DistEdu, що використовується в НаУКМА, що робить використання програми більш зручним для студентів НаУКМА [2].

Результатом цієї роботи є програмний продукт, який може бути використаний студентами Києво-Могилянської академії, а також інших вишів, що використовують Moodle в навчальному процесі, для покращення самодисципліни та ефективності шляхом відстеження свого часу та навчальних тенденцій.

1 АНАЛІЗ НАЯВНИХ РІШЕНЬ ДЛЯ ВІДСТЕЖЕННЯ НАВЧАЛЬНОГО ЧАСУ

1.1 Огляд популярних програмних рішень

В мережі Інтернет можна знайти багато програмних рішень для відстеження часу. Серед них переважають системи для обліку робочого часу, такі як Yaware TimeTracker (<https://yaware.com.ua/>), Worksection (<https://worksection.com/>), Toggl Track (<https://toggl.com/>), Memtime (<https://www.memtime.com/>), TrackingTime (<https://trackingtime.co/>), Early (<https://early.app/>), Clockify (<https://clockify.me/>) та інші.

Усі з перелічених систем пропонують інструменти для відстеження часу: автоматичне фонове відстеження активності на комп'ютері користувача, додавання активності через календар, запуск секундоміра або ж поєднання декількох з цих способів. Також у них присутня певна аналітика по витраченому часу. Хоча функціонально вищеперераховані та інші подібні системи можуть бути використані для відстеження часу на навчання, вони відрізняються за своєю метою та предметом аналітики, адже розроблені для організації робочого процесу у компаніях та аналізу продуктивності співробітників, тож детально розглядатись не будуть.

Застосунків націлених на відстеження навчального часу менше, але серед них можна виділити Forest (<https://www.forestapp.cc/>), Study Bunny (<https://superbyte.site/studybunny>), Athenify (<https://athenify.io/>), Pomofocus (<https://pomofocus.io/app>) та Study Tracker: Your AI Coach (<https://play.google.com/store/apps/details?id=ataberkw.derstakip>). Вони націлені на індивідуальне використання з метою відстеження навчального прогресу, покращення концентрації та мотивації та обліку часу, проте кожен з них має свої концептуальні особливості.

Ідея застосунку Forest полягає у вирощуванні віртуального лісу у процесі навчання, а Study Bunny надає можливість розвивати віртуального кролика. Концепт Pomofocus базується на використанні методу Pomodoro, який полягає в чергуванні двадцятип'ятихвилинних сесій концентрації та коротких перерв [3]. Ці застосунки є доволі універсальними та можуть бути використані для організації свого часу поза навчанням, тоді як Athenify націлений саме на студентів та учнів і вирізняється комплексним підходом до організації навчального процесу на основі персональних часових цілей на семестр. Study Tracker теж розроблений для навчання, проте має вузьку нішу: він націлений на підготовку до екзаменів та надає можливість додавати уроки, теми та питання, винесені на іспит.

1.2 Порівняльний аналіз аналогів

Перед розробкою системи Study Time Tracker було проведено аналіз вищезгаданих аналогів: Forest, Study Bunny, Athenify, Pomofocus та Study Tracker. Метою цього аналізу є визначення сильних та слабких сторін кожного з цих інструментів, а також їх відповідність потребам студентів у процесі самоорганізації навчального часу.

У цьому аналізі порівнюються функціональні можливості систем, повнота даних, які аналізуються, та доступність. Результати аналізу наведено у таблицях 1.1 – 1.3.

У таблиці 1.1 порівняно функціональні можливості для керування часом, та дисципліною у застосунках Forest, Study Bunny, Athenify, Pomofocus та Study Tracker. Усі додатки мають різний набір можливостей, проте у більшості з них реалізована принаймні половина розглянутого функціоналу, окрім Pomofocus, в якому реалізовано лише таймер.

	Forest	Study Bunny	Athenify	Pomofocus	Study Tracker
Наявність секундоміра	+	+	+	-	+
Наявність таймера	+	+	-	+	+
Наявність календаря	-	+	-	-	+
Режим глибокого фокусування	+	-	-	-	-
Наявність системи нагород та досягнень	+	+	+	-	-

Таблиця 1.1 – Результати порівняння аналогів за функціоналом керування часом і дисципліною

Найбільш поширеним інструментом для відстеження часу у цих застосунках є таймер та/або секундомір, цей функціонал присутній майже у кожній із зазначених програм. Study Bunny та Study Tracker також надають можливість переглядати активності у вигляді календаря. Водночас Study Tracker дозволяє також редагувати активність на календарі. Forest додатково пропонує режим глибокого фокусування або ж блокування відволікаючих програм, який не дозволяє користуватись певними застосунками під час навчальної сесії.

Більшість розглянутих застосунків використовують елементи гейміфікації, такі як нагороди, дошки лідерів, внутрішню валюту та можливість індивідуалізувати своє середовище та навчальний процес у застосунку [4]. Наприклад, у Forest за ігрові монети можна купувати нові віртуальні дерева та фонові звуки, у Study Bunny ж можна розвивати віртуального кролика. Ці два

застосунки та Athenify також використовують систему досягнень, які видаються користувачам за дотримання їх навчального плану, подолання певної часової відмітки тощо. Такий підхід підвищує мотивацію та залучення користувачів у процес навчання, і змушує їх більше часу проводити за навчанням за участі застосунку [5].

У таблиці 1.2 наведено результати порівняння за повнотою даних, які зберігаються, та методами візуалізації статистики. За усіма показниками лідирують Athenify та Study Tracker, водночас найгірші результати має Study Bunny.

	Forest	Study Bunny	Athenify	Pomofocus	Study Tracker
Наявність статистики	+	+	+	+	+
Використані методи візуалізації статистики	Лінійні графіки, стовпчикові та кругові діаграми	Текстові дані	Стовпчикові, кругові та пелюсткові діаграми, таблиці, списки, дашборди	Стовпчикові діаграми, списки	Стовпчикові діаграми, лінійні графіки
Можливість додавати навчальні предмети	+	+	+	+	+
Можливість додавати завдання до предметів	-	-	+	-	+
Можливість додавати оцінки	-	-	-	-	-

Можливість налаштувати період для статистики	+	-	+	+	+
Можливість ставити цілі	-	-	+	-	+

Таблиця 2.2 – Результати порівняння аналогів за повнотою даних та їх візуалізацією

Усі застосунки надають статистику по витраченому часу. Більшість з них відображають її у вигляді різних графіків та діаграм, проте поширеними також є списки та картки. Крім того, Athenify та Study Tracker надають можливість ставити цілі, що розширює їх панель статистики інформацією про прогрес відносно мети. Варто відзначити, що Athenify аналізує найширший обсяг статистичних даних та використовує найбільше різних методів візуалізації, порівняно з іншими аналогами.

Проте жоден із застосунків не зберігає дані про оцінки за завдання і, відповідно, не враховує ці дані в аналітиці. Крім того, тільки в Athenify та Pomofocus є можливість додавати завдання до предметів або проєктів, інші ж застосунки надають лише можливість створювати теги.

У таблиці 1.3 порівняно такі аспекти як цінова політика та кросплатформність.

	Forest	Study Bunny	Athenify	Pomofocus	Study Tracker
Операційні системи, на яких працює застосунок	Android	Android, IOS	Android, IOS, є вебверсія	Windows, Linux, є вебверсія	Android

Можливість інтеграції з іншими системами	-	-	-	-	-
Ціна	Від 75 грн на місяць	Безплатно	Від 5,99 євро на місяць	Від 1.5 долара на місяць	Від 51.66 грн на місяць
Наявність реклами	+	+	-	-	+
Наявність безплатного пробного періоду	7 днів	-	30 днів	7 днів	-
Наявність безплатної версії	+	+	-	+	+

Таблиця 3.3 – Результати порівняння аналогів за доступністю

Досліджувані програми розроблені здебільшого під мобільні пристрої, крім Athenify та Pomofocus, які доступні як на мобільних пристроях, так і на комп'ютерах.

Кожен із застосунків має безплатну версію з обмеженим функціоналом чи рекламою або безплатний пробний період, який дозволяє користувачу визначитись, чи підходить йому цей продукт. Найбільша ціна та тривалість пробного періоду в Athenify, тоді як Study Bunny взагалі не має платної версії.

Також жоден з застосунків не надає можливості інтеграції з іншими системами та сервісами, що означає, що всі дані мають заповнюватись користувачем вручну.

Отже, усі аналізовані аналоги поділяють мінімальний набір функцій для відстеження часу й перегляду статистики та мають свої особливості, переваги та недоліки у засобах відстеження часу, охопленні даних для аналітики,

візуалізації цих даних та способах використання. Проте, жоден з аналізованих додатків не надає можливості інтеграції з іншими системами та сервісами, а також не враховує дані про оцінки у своїй аналітиці.

Комбінуючи сильні сторони цих застосунків, можна створити більш ефективну систему для організації навчального часу, яка буде враховувати оцінки в аналізі продуктивності та дозволить студентам спростити використання системи, шляхом автоматизованого додавання курсів та завдань з університетської платформи.

2 МЕТОДИ ДЛЯ ВІЗУАЛІЗАЦІЇ ДАНИХ

2.1 Основні аспекти при виборі методів візуалізації даних

Візуалізація завжди відігравала велику роль у когнітивних процесах людини, оскільки найбільше інформації про світ отримується за допомогою зору [6]. З розвитком технологій необхідність у візуалізації даних зростає, оскільки збільшується обсяг та складність даних, які треба аналізувати. Сучасні методи комп'ютерної візуалізації дають можливість ефективніше обробляти великі обсяги даних та допомагають у прийнятті рішень [6], [7]. Завдяки цьому підхід до візуалізації даних набуває особливої важливості у контексті роботи з великими обсягами числової, текстової та часової інформації, що особливо актуально для аналізу навчального процесу.

Існує багато методів візуалізації даних та їх класифікацій. Для вибору оптимального методу та інструменту треба враховувати такі аспекти, як проблема або завдання, яке має бути вирішене, характер та розмірність даних, які аналізуються, їх інформаційна структура та очікуваний тип взаємодії з ними [8].

Шнайдерман [9] виділив 7 типів даних та 7 завдань, які вони допомагають вирішити. Ці завдання включають:

- огляд
- масштабування
- фільтрація
- запит деталей за потребою
- відображення взаємозв'язків
- збереження історії
- витяг підмножин даних

а типи даних включають:

- одновимірні
- двовимірні
- тривимірні
- багатовимірні
- часові
- ієрархічні
- мережеві.

Одновимірні дані містять тексти, впорядковані списки тощо, двовимірні – площини, такі як карти або схеми поверхів, тоді як тривимірні репрезентують об’єкти з реального світу [10]. Багатовимірні дані представляють об’єкти, які мають багато атрибутів, та відображаються на тривимірних графіках, як точки у просторі на координатах, які відповідають значенням атрибутів [9]. Варто зазначити, що ці чотири типи також можна віднести до розмірності даних.

Часові дані найчастіше представлені у вигляді часових шкал, які використовуються для відображення хронологічної послідовності подій та співвідношень їх тривалості, часу початку або кінця, перетину тощо [11].

Ієрархічні структури використовуються для відображення вкладених відносин, де об’єкти розташовані на різних рівнях залежності, а мережі – для структур зі складнішими зв’язками [9].

Щодо завдань, огляд має на меті дати користувачу загальне уявлення про дані, їх структуру і розподіл з обмеженої можливостю деталізації. Масштабування використовується для аналізу даних на різних рівнях деталізації задля виявлення загальних та локальних особливостей та закономірностей [9]. Наприклад, цифрові карти дозволяють масштабувати місцевість багато разів та отримувати більше деталей про певне місце.

Фільтрація допомагає відокремити релевантні дані від нерелевантних [9]. Прикладом методу, який виконує це завдання, може бути повзунки, що

регулюють відображення певного контенту та дозволяють швидко перемикатись між даними.

Стратегія «запит деталей за потребою» забезпечує доступ до додаткової інформації лише у випадку необхідності, що допомагає уникнути перевантаження користувача зайвими деталями [12]. Поширеною реалізацією є спливні вікна з детальною інформацією про об'єкт або елементи, що згортаються та розгортаються.

Відображення взаємозв'язків спрямоване на знаходження та візуалізацію зв'язків між окремими елементами даних [9]. Прикладом може бути рекомендація схожих або пов'язаних об'єктів в описі іншого об'єкту або у пошуку.

Збереження історії потрібне для обліку дій та можливості їх повторювати та скасовувати, а стратегія витягу підмножин даних сприяє виділенню та подальшому аналізу окремих сегментів великого обсягу інформації [9].

За характером дані можуть бути категоричними, тобто неупорядкованими, порядковими, тобто такими, які можна відсортувати за певними характеристиками, та кількісними [8].

За типом взаємодії представлення може бути статичним, маніпулятивним та трансформованим. Цей аспект визначає ступінь інтерактивності візуалізації та впливатиме на вибір інструменту для візуалізації [8].

2.2 Синтез оптимальних методів для візуалізації навчального процесу

Відповідно до вищезазначених аспектів було обрано методи для візуалізації даних про навчальний процес студента, а саме про кількість часу, витраченого на конкретні предмети й завдання та його ефективність.

Серед найбільш поширених сучасних методів візуалізації зазначають умовне форматування, графіки та діаграми, лінії тренду, дашборди, інфографіки, таймлайни, картограми, ментальні карти тощо [10]. У цій роботі за основний метод було обрано дашборд, який є комбінацією декількох інших методів.

2.2.1 Застосування графіків та діаграм

Система має показувати користувачу дані про предмети та завдання, над якими він працював найдовше, найбільш навантажені години, дні та місяці, прогрес порівняно з попереднім звітним періодом тощо. Отже, маємо дані порядкового та кількісного характеру, одновимірного та часового типу. Більшість таких даних можна візуалізувати за допомогою графіків та діаграм. Графіки найкраще підходять для відображення тенденцій, тоді як діаграми – для порівняння кількісних значень [13], тож доцільно буде відобразити хронологічні тенденції фокусування користувача на лінійних графіках. Таким методом було візуалізовано тенденції фокусування протягом доби у порівнянні з попереднім звітним періодом (рисунок 2.1).

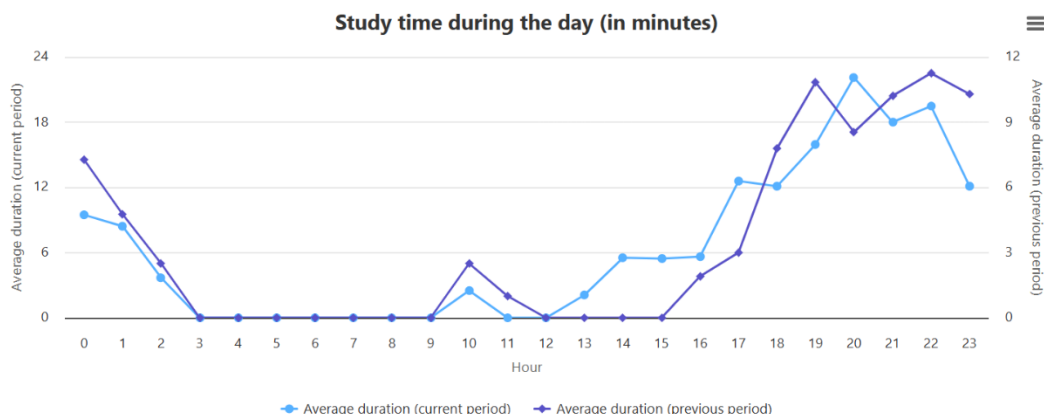


Рисунок 2.1 – Приклад використання лінійного графіку

Діаграми найкраще підходять для відображення кількісних та відсоткових співвідношень або впорядкованих списків, тож вони були використані для візуалізації списків предметів та завдань, відсортованих за кількістю

витраченого на них часу. Зокрема було обрано кругові діаграми для кращої ілюстрації частки кожного предмету або завдання у навчальному часі студента (рисунок 2.2).

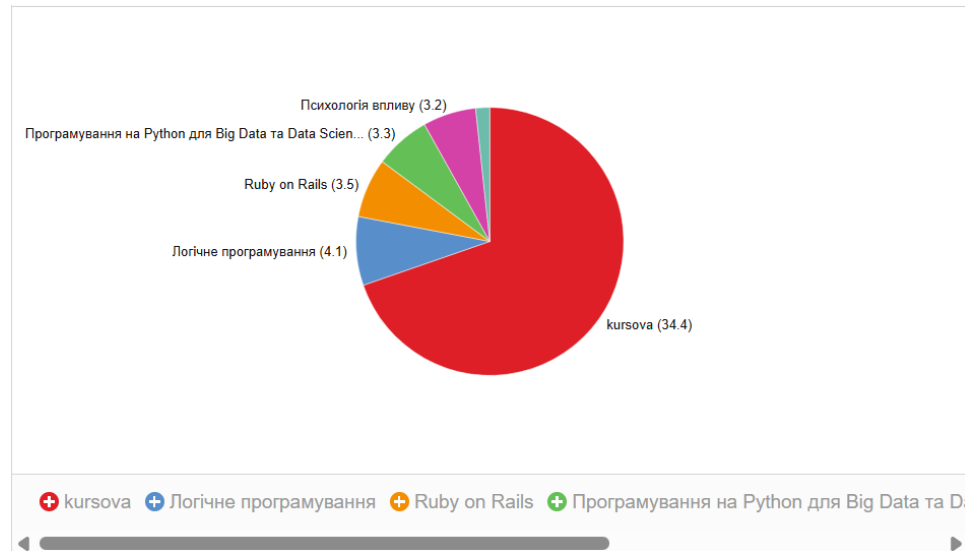


Рисунок 2.2 – Приклад використання кругової діаграми

Крім того, кожен предмет охоплює підмножину завдань, тому оптимальна візуалізація передбачає можливість запити деталей та маніпуляції виглядом діаграми. Отже, ця діаграма має бути інтерактивною та дозволяти «розгортати» сегменти предметів для перегляду частки часу, витраченого на його завдання (рисунок 2.3).



Рисунок 2.3 – Приклад «розгорнутої» інтерактивної кругової діаграми

Також для відображення кількісних даних добре підходять стовпчикові діаграми. Вони можуть бути застосовані як для впорядкованих, так і неупорядкованих даних. Їх використання доцільніше за кругові діаграми, коли важливо зберегти послідовність або порядок чи показати складники. Для другого випадку існує окремий вид стовпчикових діаграм – з накопиченням. В контексті візуалізації навчального часу вона добре підходить для відображення навчального часу по датах з розбиттям на предмети (рисунок 2.4). Зокрема на цьому прикладі дати впорядковані за кількістю витраченого часу у спадному порядку.

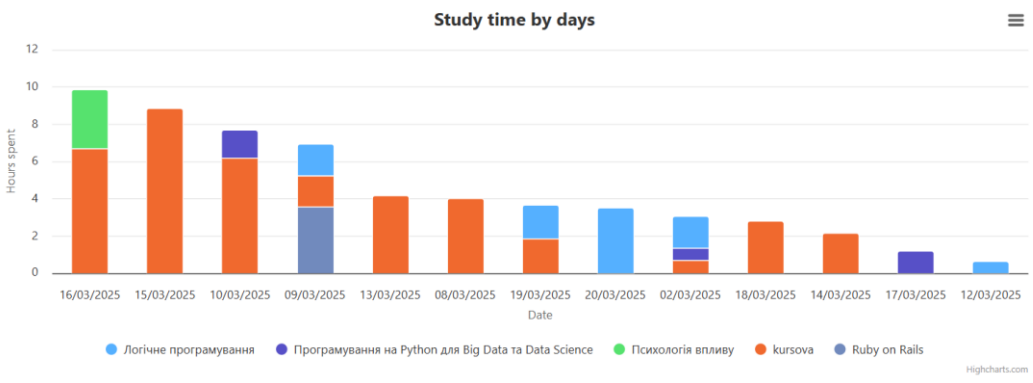


Рисунок 2.4 – Приклад використання стовпчикової діаграми з накопиченням

Проте ці ж дані можна подати в іншому вигляді, якщо хронологічна послідовність важливіша за кількісне значення. Для відображення розподілу кількісних даних протягом місяця може підійти такий метод візуалізації як теплова карта, або ж heatmap. Теплові карти використовуються для аналізу двовимірних даних та знаходження у них закономірностей на певних ділянках [14]. Візуальна особливість цього методу в тому, що кількісна характеристика даних виражається не розміром, а кольором, наприклад його непрозорістю, насиченістю, відтінком тощо. Оскільки місяць на календарі можна подати у вигляді рядків (тижнів) та колонок (днів), дані про витрачений протягом місяця час також доцільно подавати за допомогою теплової карти (рисунок 2.5).

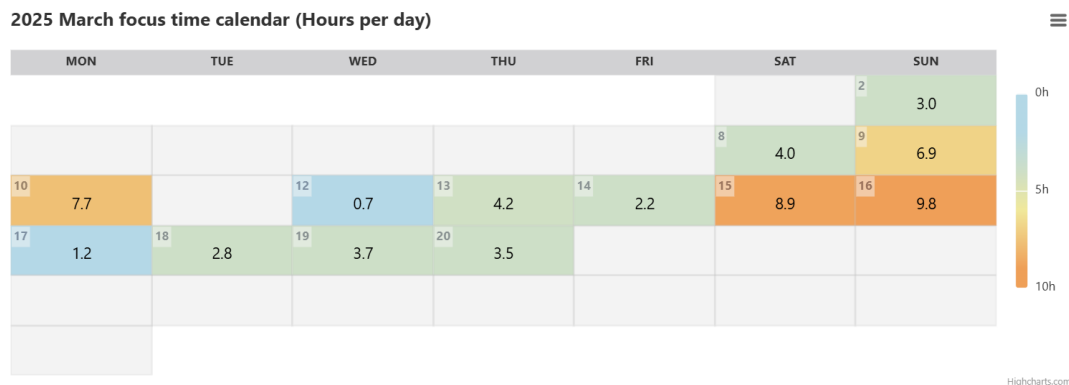


Рисунок 2.5 – Приклад використання теплової карти у вигляді календаря

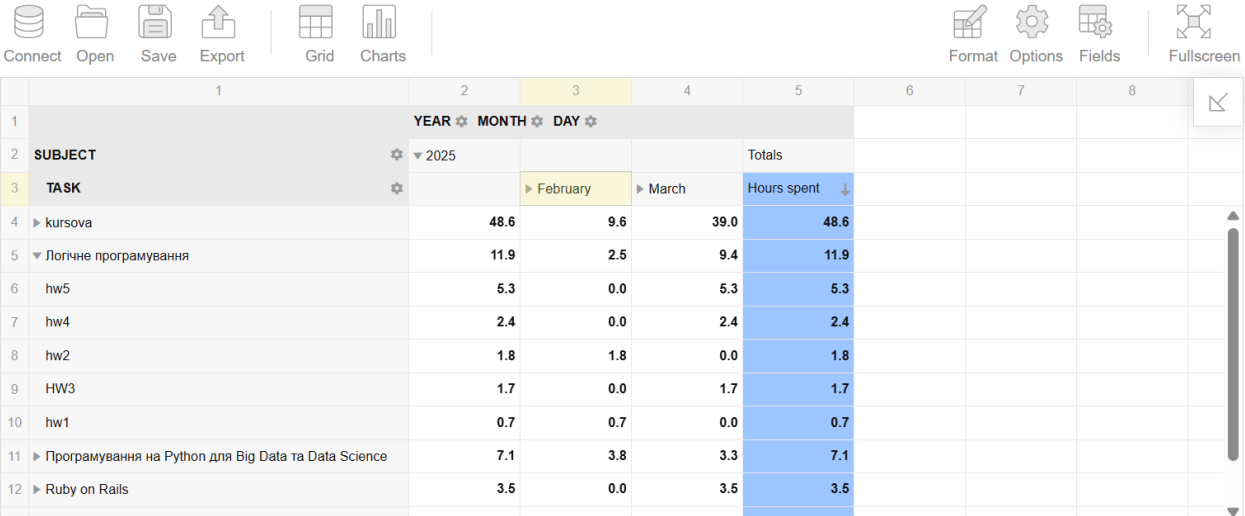
Така візуалізація спрощує сприйняття даних у хронологічному порядку та дозволяє виявити дні, які студент приділяє навчання найбільше.

2.2.2 Застосування зведених таблиць

Окрім діаграм та графіків поширеним методом для візуалізації даних є таблиці, зокрема зведені таблиці. Вони використовуються для групування та агрегування великих об'ємів даних та дозволяють детальніше проаналізувати дані під різними кутами [15]. Цей метод представлення даних є більш інтерактивним та більш універсальним за графіки та діаграми, адже він

дозволяє маніпулювати представленням даних у реальному часі та працювати з багатовимірними даними.

У системі Study time tracker дані про навчальну сесію містять її довжину, предмет, завдання, оцінку та дату початку. Такий набір атрибутів можна згрупувати по різному та досліджувати різні взаємозв'язки між ними. Наприклад, у зведеній таблиці на рисунку 2.6 дані згруповано за датою по колонках та за предметом і завданням по рядках, а за досліджувані значення було обрано суму тривалості сесій у годинах.



	1	2	3	4	5	6	7	8
		YEAR	MONTH	DAY				
1		2025			Totals			
2	SUBJECT							
3	TASK		February	March	Hours spent			
4	► kursova	48.6	9.6	39.0	48.6			
5	▼ Логічне програмування	11.9	2.5	9.4	11.9			
6	hw5	5.3	0.0	5.3	5.3			
7	hw4	2.4	0.0	2.4	2.4			
8	hw2	1.8	1.8	0.0	1.8			
9	HW3	1.7	0.0	1.7	1.7			
10	hw1	0.7	0.7	0.0	0.7			
11	► Програмування на Python для Big Data та Data Science	7.1	3.8	3.3	7.1			
12	► Ruby on Rails	3.5	0.0	3.5	3.5			

Рисунок 2.6 – Приклад зведеної таблиці

Однак можна замінити, скажімо, суму тривалості навчання на середнє значення оцінок та досліджувати загальну зміну успішності та зміну успішності по предметах.

Тож зведена таблиця дозволить користувачу самостійно дослідити залежності, які інші методи візуалізації, можливо, не висвітлили, хоча її сприйняття може бути дещо складнішим за сприйняття графіків та діаграм.

2.2.3 Застосування календаря

Хоча основним фокусом застосунку є візуалізація аналітики та статистики по витраченому часу, може виникати потреба переглядати дані про навчальні сесії в неагрегованому вигляді. Як вже було згадано вище, дані про навчальні сесії містять дату початку, тривалість, предмет і завдання, яке виконувалось. Це хронологічні дані, а, отже, важливо відобразити їх положення відносно шкали часу та тривалість. Дата початку зберігається у точності до секунд, тож найзручніше буде відобразити дані на календарі.

На відміну від календарної теплової карти, яка була розглянута раніше, звичайний календар дозволяє отримати детальну інформацію про усі навчальні сесії дня, а не тільки їх сумарну тривалість. Така можливість реалізовується через наявність різних режимів перегляду календаря, зокрема місячного, тижневого та добового. Відображення навчальних сесій на календарі в добовому та тижневому режимах дозволяє бачити їх положення в межах доби та порівнювати їх тривалості, при цьому зберігаючи число та місяць.

Тижневий перегляд, приклад якого представлено на рисунку 2.7, також дозволяє візуально порівняти розподіл навчального часу протягом тижня: наприклад, в який день було більше навчальних сесій, який предмет домінував, чим була зайнята поточна година минулого дня тощо. Крім того, кожна сесія на календарі набуває кольору свого предмета, що також полегшує сприйняття даних.

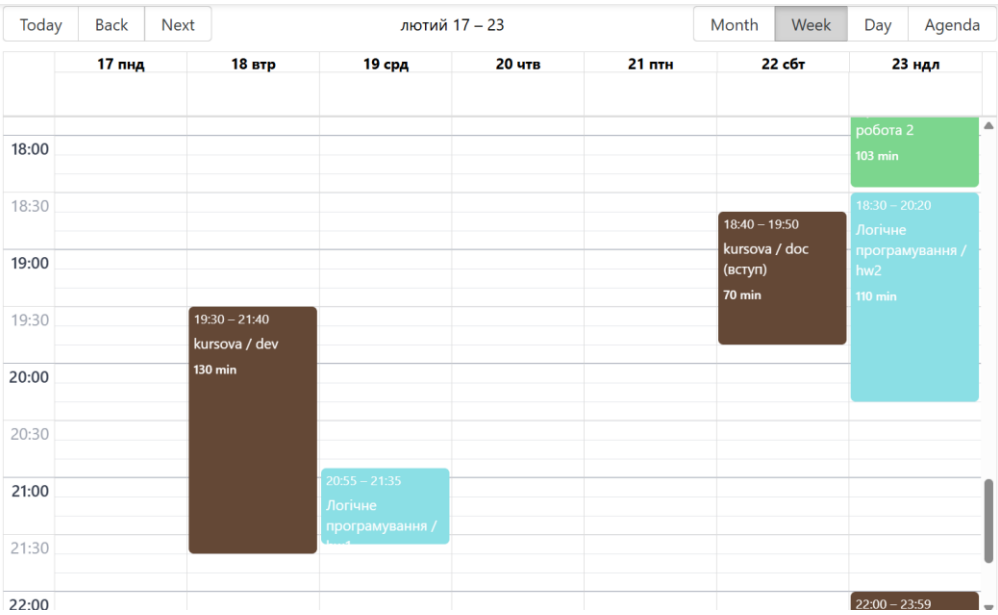


Рисунок 2.7 – Тижневий перегляд навчальних сесій на календарі

Місячний перегляд є менш інформативним з неагрегованими даними, проте він також дозволяє побачити кількість сесій кожного дня та предмет і завдання деяких з них (рисунок 2.8).

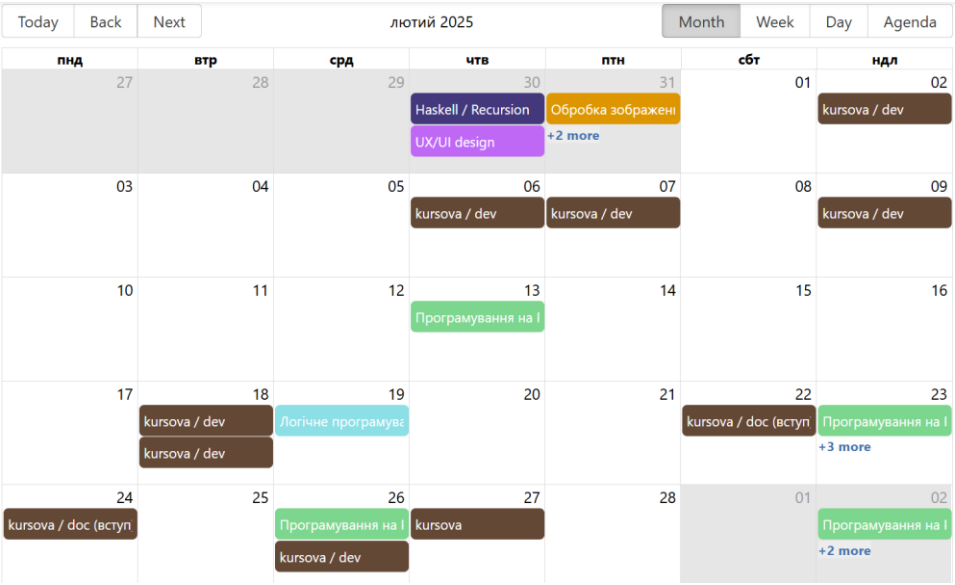


Рисунок 2.8 – Місячний перегляд навчальних сесій на календарі

Отже, календар – це оптимальний метод для візуалізації неагрегованих даних про час, адже він має різні шкали вимірювання, зберігає хронологічну послідовність, дозволяє порівнювати об’єкти за різними атрибутами за допомогою кольору, розміру та положення.

2.2.4 Застосування карток та списків

Розглянуті вище методи представляють інформацію переважно графічно або у вигляді таблиць та відображають повні дані за звітний період, проте іноді є потреба переглянути якусь частину даних, яка є найбільш релевантною для користувача. В такому випадку в пригоді стає структуроване текстове представлення у вигляді списків та карток. Картки являють собою згруповані у прямокутній області текстові дані, які включають тільки релевантну інформацію про певний об’єкт (рисунок 2.9). Вони не виділяються як окремий метод візуалізації даних, проте в цій роботі їм було знайдено широке та ґрунтовне застосування.

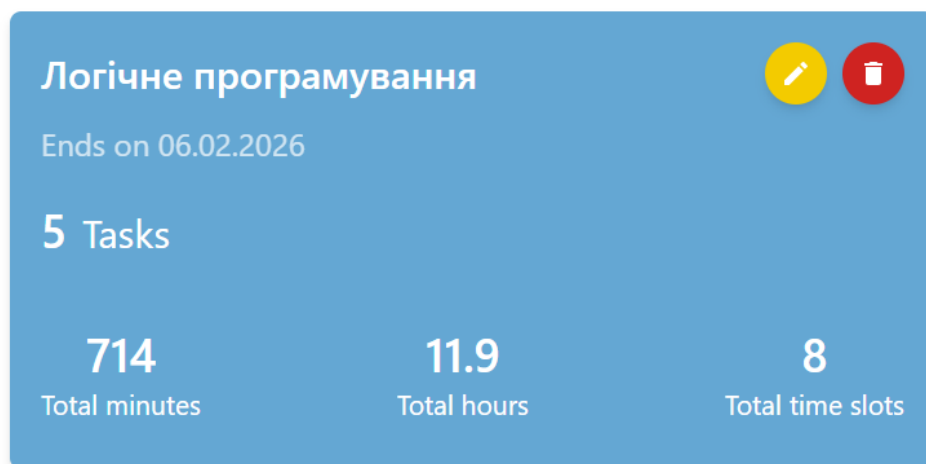


Рисунок 2.9 – Приклад картки, як методу візуалізації даних про предмет

Немає єдиної структури для картки, у ній можна відображати будь-які дані у будь-якій кількості, проте варто уникати інформаційного перенавантаження. Основна мета цього методу – показати корисну для користувача інформацію

про об'єкт у компактному та структурованому вигляді. Картка має містити заголовок з атрибутом, який її ідентифікує, та числові дані, які описують представлений об'єкт. Числові дані можуть бути як агрегованими, так і неагрегованими, та повинні мати текстовий підпис, який пояснює, що ці числа представляють. Також колір фону картки може представляти певний атрибут об'єкта. Наприклад, навчальні предмети мають атрибут з кольором, який допомагає їх ідентифікувати серед інших предметів на календарі або серед переліку карток.

Крім того, картки можна використовувати у ролі елементів списку. Якщо список впорядкований, картки також мають містити свій порядковий номер. Поєднання цих двох методів забезпечує візуалізацію розрізу впорядкованих даних для огляду релевантної інформації. Це корисно тоді, коли хронологічна послідовність та тенденції не важливі, але важлива впорядкованість за певною характеристикою та деякі деталі, а також нема потреби оглядати всі дані. Наприклад, на рисунку 2.10 зображено впорядкований список карток, який відображає три найбільш завантажені навчанням дні тижня студента.

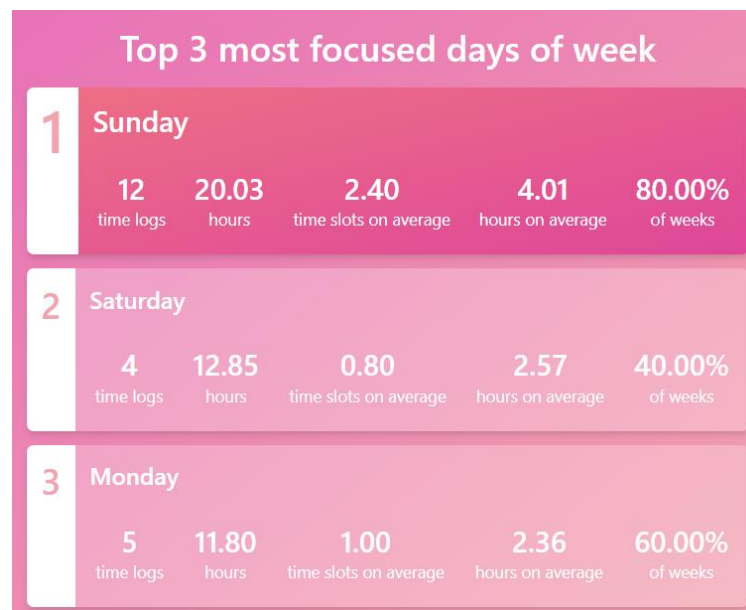


Рисунок 2.10 – Приклад використання впорядкованих списків карток

Хоча ці ж дані можна відобразити на лінійному графіку або стовпчиковій діаграмі, вони не дадуть стільки інформації про дні, скільки картки, а також візуальний пошук трьох найбільш навантажених днів може бути дещо складнішим ніж перегляд впорядкованого по цьому параметру списку.

Варто зазначити, що один і той самий об'єкт може бути представлений картками у різних списках. Оскільки картки не мають чіткого переліку даних для відображення, їх можна змінювати залежно від аспекту аналізу. Наприклад, завдання можна представити у вигляді картки із зазначенням загального часу виконання, якщо треба відобразити список найбільш часовитратних завдань, а в списку найбільш успішних завдань можна зазначати також оцінку.

Тож картки є оптимальним методом візуалізації даних про певний об'єкт або сутність, а впорядковані списки карток найкраще підходять для огляду розрізу відсортованих за певною характеристикою даних з деталями, які релевантні для завдань цього конкретного аналізу.

2.3 Застосування LLM у візуалізації даних

За останні роки штучний інтелект набув широкого застосування у повсякденному та професійному житті [16]. З 2017 року, після того як дослідники Google запропонували модель трансформера [17], почався активний розвиток базованих на ній моделей GPT (Generative Pre-trained Transformer), які здатні генерувати зв'язний і логічний текст природною мовою [18].

Зокрема компанією OpenAI була представлена LLM модель GPT-3, яка є першою моделлю штучного інтелекту, що стала доступною для користувачів через API, тим самим заснувавши новий режим доступу – Model as a Service (MaaS) [19]. Можливість використання через API дозволила застосування

великих мовних моделей у різних програмних рішеннях та напрямках, у тому числі у візуалізації даних та аналітиці.

2.3.1 Обґрунтування та суть методу

У попередньому розділі було розглянуто різні методи візуалізації даних, але вони всі мають спільний недолік. Картки та календар допомагають підсумувати та зручно і структуровано відобразити дані про певні об'єкти та підходять для огляду з деталями, графіки та діаграми надають можливість досліджувати тенденції та співвідношення даних протягом поточного звітного періоду, а зведені таблиці дозволяють самостійно групувати та аналізувати дані, проте жоден з цих методів не підтримує порівняльну аналітику з попереднім періодом.

Хоча графіки та діаграми надають можливість відобразити одночасно дані поточного та попереднього періоду та, погортавши календар, можна подивитись попередні навчальні сесії, це представлення не є дуже інформативним та вимагає від користувача самостійної обробки та порівняння даних. Проте можливості LLM дозволяють це зробити за нього.

Суть методу полягає у тому, щоб засобами інженерії промптів (від англ. “prompt engineering”) змусити LLM обробити дані за поточний та попередній звітні періоди та згенерувати короткий та структурований письмовий звіт із загальними досягненнями користувача за поточний період, змінами у навчальних тенденціях відносно попереднього періоду та рекомендаціями на їх основі. Далі згенерований звіт виводиться користувачу у вигляді тексту із заголовками, списками та елементами текстової стилізації.

Для генерації використовується вищезгаданий OpenAI API, адже він дозволяє легко взаємодіяти з моделлю, отримуючи готові текстові відповіді від сервісу. Комунікація з моделлю відбувається шляхом HTTP-запитів, які містять назву моделі, масив повідомлень (промптів) та, за потреби, додаткові

налаштування. У відповідь API надсилає JSON об'єкт, який містить масив варіантів згенерованих моделлю повідомлень у вигляді тексту у форматі markdown.

2.3.2 Інженерія пром프트 для отримання якісного звіту

Як вже було зазначено, моделі відправляється пром프트 – інструкції, які встановлюють правила для моделі та забезпечують певні якості згенерованого результату [20]. Те, як побудований пром프트, безпосередньо впливає на якість відповіді від LLM та її відповідність бажаним критеріям [21].

Існує кілька практик та технік інженерії пром프트 для покращення якості результату. Однією з технік є використання шаблонів пром프트, які містять змінні, що при створенні пром프트 будуть замінені певними текстовими даними [22]. Промпти, які використовуються у Study Time Tracker побудовані на шаблоні, який містить змінні для часових меж поточного звітної періоду та, власне, масивів часових сесій за поточний та попередній звітні періоди (рисунок 2.11). Такий підхід дозволяє використовувати один і той самий пром프트 з різними даними.

```
const prompt = `Analyze the current timelogs data for period from `${timeRange.start}` to `${timeRange.end}` and provide insights on concentration tendencies. Compare to previous period.  
Timelogs data: `${JSON.stringify(timelogs)}`  
Previous timelogs data: `${JSON.stringify(prevTimelogs)}``;
```

Рисунок 2.11 – Приклад використання шаблонів в промпті

Інші техніки включають надання моделі контексту: прикладів, ролі, деталізованих інструкцій та стилю або формату викладу [22]. При побудові пром프트 для генерації звіту по навчальному часу моделі було надано роль дата-аналітика, який має на меті допомогти користувачам отримати цінну інформацію про їх навчальні звички та тенденції. Також було надано детальні

інструкції про те, яку інформацію треба аналізувати, які критерії порівнювати та що описувати у звіті, а саме: дослідити кореляцію між годинами доби та оцінками, отриманими за завдання, які в них виконувались, порівняти тенденції поточного періоду з попереднім, визначити найскладніші предмети та рекомендації для покращення ефективності. Моделі також були надані приклади розрахунку успішності завдання та вказівки щодо формату та структури викладу інформації та обсягу і простоти тексту. Оновлений промпт зображено на рисунках 2.12 та 2.13.

```
const prompt = `Analyze the current timelogs data for period from ${timeRange.start} to ${timeRange.end} and provide insights on concentration tendencies. Also analyze how studying at specific hours of the days affected the score (grade/maxGrade) of graded tasks, that were studied at those hours. Also consider how duration of studying on task impacts the score. Do not mind the quantity of grades, compare scores only among graded tasks. Mind that the score is calculated by dividing grade of a task by the maxGrade of the task (e.g. 10/10 is as good as 100/100) and is displayed in percents (%). Highlight changes and tendencies since the previous period of time (previous timelogs data is listed below), include percentages and comparisons. Highlight numbers in the data. Keep it really brief. Write this in simple language as if you are explaining this to a friend who's data you are analyzing, structure it in bullet points. Mind that dates are in UTC, convert them to Ukrainian time and format. According to your data, suggest which subject is the hardest and which should be studied more or less. Compare to previous period. Timelogs data: ${JSON.stringify(timelogs)} Previous timelogs data: ${JSON.stringify(prevTimelogs)}`;
```

Рисунок 2.12 – Розширений контекстом промпт

```
messages: [
  { role: "system", content: "You are a data analytic helping user gain valuable insights from their study timelogs" },
  { role: "user", content: prompt }
],
```

Рисунок 2.13 – Задання ролі у промпті

Насамкінець найкращі практики інженерії промптів включають ітеративне вдосконалення промптів [23]. Його суть полягає у тестуванні промпту на моделі та його поетапному коригуванні залежно від відповідей і їх відповідності очікуванням. Зокрема шаблон промпту у Study Time Tracker

проходив численні ітерації коригувань та вдосконалень із застосуванням різних технік інженерії промптів, доки згенеровані звіти не стали достатньо чіткими, інформативними та точними.

2.3.3 Візуалізація звіту

Тож було розглянуто перший етап методу – генерацію звіту, проте є також друга частина – його візуалізація. Звіт подається на сторінку у вигляді віконця, яке закріплене у правому нижньому кутку сторінки з аналітикою. Це віконце відкривається на вимогу користувача кнопкою, закріпленою в тому самому кутку. Такий формат доступу було впроваджено задля економії ресурсів системи та уникнення інформаційного перенавантаження користувача.

Як вже було згадано, API OpenAI повертає відповідь від моделі у форматі markdown, оскільки експериментальним шляхом було виявлено, що якість результату, згенерованого великою мовною моделлю суб'єктивно краща, ніж при формуванні промпту, який має видавати HTML. Проте, у вебзастосунку вивід результату має відбуватися в форматі HTML. Цього можна досягнути за допомогою пост-опрацювання результату за допомогою готових інструментів або бібліотек, конвертуючи markdown у HTML.

Відповідно до вимог прописаних в промпті, звіт структуровано за допомогою заголовків та списків, також ключові слова та значення виділено жирним шрифтом, що дозволяє легше сприймати інформацію (рисунк 2.14).

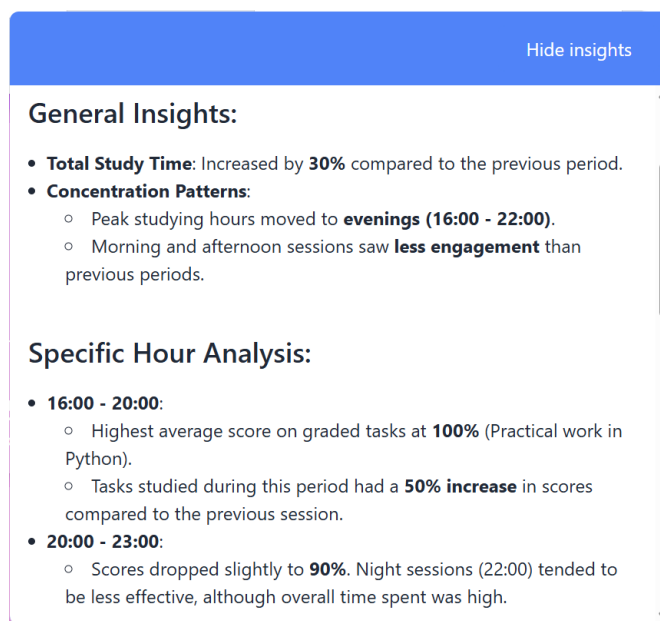


Рисунок 2.14 – Фрагмент звіту, згенерованого LLM

2.4 Новизна та цінність використаних підходів

Аналогічні застосунки, які були розглянуті в першому розділі роботи, також використовують різні методи візуалізації даних. Згідно з порівняльною таблицею (див. таблицю 1.2), більшість застосунків використовують графіки та діаграми для відображення статистики, деякі використовують також картки, списки або звичайні таблиці. Однак жоден з них не включає теплові карти, зведені таблиці та списки карток, використані в системі, розробленій в рамках цієї курсової роботи. У двох аналогів реалізовано інтерактивний календар, але вони мають менш широку аналітику, що є ще одним вагомим досягненням розробленої системи.

У поточній роботі використано поєднання різних методів візуалізації даних, нетипове серед аналогів. Також успішно застосовано LLM для аналізу навчальних тенденцій та генерації звітів, що, у поєднанні з відповідними промптами має більше залучати та мотивувати студента.

3 РОЗРОБКА СИСТЕМИ STUDY TIME TRACKER

В результаті практичної частини дослідження було розроблено вебзастосунок Study Time Tracker. Клієнтська та серверна частини написані на JavaScript фреймворку NextJS, а для зберігання даних було створено базу даних на MongoDB. Для написання стилів клієнтської частини було використано CSS фреймворк Tailwind CSS. У системі реалізована аутентифікація користувачів з дотриманням стандарту JWT та підтримка сеансів за допомогою технології Cookies. Для візуалізації даних було використано такі інструменти та бібліотеки:

- Flexmonster Pivot Table & Charts – для зведених таблиць, графіків та діаграм;
- Highcharts – для відображення графіків та діаграм в інтеграції з Flexmonster Pivot Table & Charts;
- React Big Calendar – для відображення інтерактивного календаря;
- React-markdown – для перетворення створеного LLM звіту з markdown у HTML для відображення на сторінці.

Вебінтерфейс системи реалізований англійською мовою та розподілений на 6 сторінок:

1. сторінка для запуску навчальної сесії (Track time);
2. сторінка зі статистикою (Dashboard);
3. сторінка з календарем (Calendar);
4. сторінка управління предметами (Subjects);
5. сторінка управління завданнями (Tasks);
6. сторінка акаунту (Account).

3.1 Функціональні можливості програми

В межах системи реалізовані функціональні можливості для взаємодії з даними та їх представленням. База даних, окрім користувачів, містить таблиці навчальних предметів, завдань та навчальних сесій. Для кожної таблиці реалізовано перелік CRUD операцій.

На сторінці предметів реалізовані такі функціональні можливості взаємодії з предметами:

- Перегляд предметів та стислої інформації про них: їх назву, колір, дату закінчення (за наявності), кількість завдань, загальну кількість витрачених годин та хвилин, загальну кількість навчальних сесій;
- Додавання нових предметів;
- Редагування предметів: їх назви, кольору та дати закінчення курсу;
- Видалення предметів;
- Перегляд повної інформації про предмет: те ж саме, що у стислій версії, та список завдань по предмету з їх стислою інформацією;

На сторінці завдань реалізований аналогічний перелік можливостей:

- Перегляд завдань погрупованих по предметах з їх інформацією: назвою, станом (оцінено чи не оцінено), оцінкою (за наявності), максимальною допустимою оцінкою, загальною кількістю затрачених годин та хвилин, загальною кількістю навчальних сесій;
- Створення нових завдань;
- Редагування завдань: їх назви, оцінки та максимальної оцінки;
- Видалення завдань;

На сторінці календаря реалізовані можливості для взаємодії з навчальними сесіями:

- Перегляд навчальних сесій у чотирьох режимах календарного представлення: денному, тижневому, місячному та порядку денного. Представлення сесій на календарі містять час початку та закінчення, назву навчального предмета та його колір, назву завдання (за наявності) та тривалість у хвилинах;
- Додавання нових сесій шляхом виділення потрібного проміжку часу мишею та зазначенням предмета та завдання у формі;
- Редагування сесій з календаря: зміна дати та часу початку і кінця сесії, її предмету та завдання;
- Видалення сесій;

На сторінці зі статистикою реалізовані можливості для взаємодії з даними та їх поданням. Контрольна панель статистики дозволяє налаштовувати такі параметри:

- Часовий діапазон для звітування. Можна обрати один з заготовлених варіантів, таких як цей або минулий тиждень, місяць, рік, семестр, або ввести власний часовий проміжок. За замовченням встановлено поточний місяць;
- Фільтр за предметом. Дозволяє обрати усі предмети (за замовченням) або один зі списку;
- Фільтр за завданням. Дозволяє обрати усі завдання (за замовченням) або одне завдання зі списку завдань предмета, який наразі обраний;
- Кількість елементів для відображення з вершини списків. За замовченням встановлено 3;

Крім того, графіки та діаграми дозволяють маніпулювати своїм відображенням, наприклад, переглядати їх в повноекранному режимі, прибирати деякі елементи, тощо. Зведені таблиці надають повну свободу для редагування та налаштування звіту.

Додатково є можливість згенерувати текстовий звіт з підсумуванням результатів звітного періоду, тенденціями, у порівнянні з попереднім періодом, та порадами для покращення ефективності.

На сторінці для запуску навчальної сесії можна запустити таймер або секундомір, підв'язаний під певний предмет та, опціонально, завдання. Також присутній режим глибокого фокусування, який відкриває таймер або секундомір у повноекранному режимі задля мінімізації відволікання на інші вкладки браузера та програми. Крім того, компактна версія секундоміра присутня у верхній панелі кожної сторінки для швидкого початку сесії.

На сторінці акаунту є можливість вийти з акаунту та переглянути дані свого профілю.

Варто також зазначити, що жодна з перелічених сторінок не доступна неавторизованим користувачам. У випадку переходу на них без авторизації, користувача буде перенаправлено на сторінку входу в систему.

3.2 Браузерне розширення для інтеграції з Moodle

Основний фокус системи – на відстеженні часу та візуалізації аналітики та статистики по отриманим даним, що є неможливим без інформації про навчальні предмети та завдання. Хоча функціональні можливості системи включають додавання предметів та завдань через інтерфейс користувача, це не єдиний шлях заповнення цих даних. Ручне створення є необхідним мінімальним функціоналом, проте займає додатковий час у користувача, що може погіршити досвід користування системою, оскільки вимагає від нього додаткових дій. Враховуючи те, що багато навчальних закладів, зокрема НаУКМА, мають платформи для організації дистанційного навчального процесу, які вже містять інформацію про курси та завдання, доцільним рішенням буде автоматизувати додавання цих даних у системі Study Time Tracker.

Інтеграція з зовнішніми платформами зазвичай передбачає використання API або прямий обмін даними між серверами. Проте такий варіант взаємодії не завжди можливо реалізовувати, адже він вимагає відкритого API або ключів доступу, а також документації. Університетські системи, побудовані на основі Moodle, такі як DistEdu, зазвичай не мають публічного доступу до API, тож реалізація такої інтеграції потребувала б безпосереднього втручання співробітників навчального закладу та надання розробнику певних прав доступу до внутрішньої системи.

Таке рішення є щонайменше непрактичним, адже вимагатиме зусиль від третіх сторін при кожній інтеграції з новими системами, або ж розробнику може бути взагалі відмовлено в доступі. Тож необхідний інший підхід, який не вимагає серверної комунікації з джерелом даних. Таким рішенням може бути браузерне розширення. У цій роботі реалізовано розширення для інтеграції з платформою Moodle, проте за цим методом можна інтегруватись з будь-якою іншою платформою.

3.2.1 Принцип роботи розширення

Завданням розширення є отримати дані про курси та завдання на платформі Moodle. Воно не має доступу до серверної частини платформи, проте має доступ до відкритих вкладок у браузері користувача та їх клієнтської частини, зокрема вмісту сторінок та мережевих запитів з їх заголовками та тілом [24], [25]. Таким чином, кожен раз, коли студент заходить у свій кабінет на платформі, розширення отримує доступ до вмісту на сторінці та запитів, які надсилаються в її межах, у тому числі відповідей від API, які містять інформацію про усі курси студента або конкретний курс.

Для перехоплення запитів у розширенні використовується webRequest API, проте він не надає доступу до тіла запиту, що унеможливорює отримання інформації про курси із запитів з його використанням. Єдиний шлях отримати тіло запитів – це використовувати chrome.debugger API. Він надає ширший

доступ до мережі та інструментів розробника через Chrome DevTools Protocol, дозволяючи отримувати тіло запитів у активній вкладці [26]. Таким чином, розширення в фоновому режимі прослуховує HTTP-відповіді на активній вкладці Moodle, отримує їх вміст і, якщо URL запиту відповідає потрібному, надсилає отримані з тіла запиту дані далі.

3.2.2 Комунікація розширення та програми. Вебсокети

Отже, розширення отримує дані про курси та завдання, проте воно існує незалежно від системи Study Time Tracker, тож система не має прямого доступу до них. Для отримання даних від розширення потрібен канал комунікації з вебзастосунком.

Для розв'язання цієї задачі існує два підходи: polling, або ж періодичне опитування, та використання технології WebSockets [27]. Суть методу polling полягає у тому, що вебдодаток із фіксованим інтервалом надсилає HTTP-запити, щоб перевірити наявність оновлень. Такий механізм є доволі простим у реалізації, проте має кілька недоліків:

- постійне навантаження на клієнтську та серверну частини системи навіть за відсутності нових даних;
- затримки у відображенні змін, що залежать від частоти опитування;
- неефективне використання мережевих ресурсів.

Оновлення від розширення надходитимуть тільки при відвідуванні сторінок навчальної системи Moodle, що є нерегулярною подією, отже при застосуванні методу polling треба було б робити часті запити, створюючи невиправдане навантаження на серверну частину Study Time Tracker та недоцільне використання ресурсів, або значно збільшити інтервал опитування, що призвело б до затримок у відображенні змін. Тож більш оптимальним рішенням буде застосування технології WebSockets. Вона дозволяє встановити постійне двостороннє з'єднання між клієнтською стороною та локальним

сервером, через який дані надходять від розширення до застосунку. Це дозволяє миттєво реагувати на нові події без потреби у додаткових HTTP-запитах, зменшуючи затримки та підвищуючи загальну продуктивність системи.

Тож при відкритті вкладки Moodle розширення отримує дані про курси та надсилає їх на вебсокет сервер, який також прослуховує клієнтська сторона Study Time Tracker. Після отримання повідомлення від вебсокета клієнт робить запит на серверну частину системи, де дані від розширення обробляються та до бази даних системи Study Time Tracker додаються, відповідно, нові курси або завдання.

3.2.3 Перспективи для подальших інтеграцій

У цьому розділі було розглянуто інтеграцію з платформами на базі Moodle на прикладі системи навчання DistEdu НаУКМА за допомогою браузерного розширення та вебсокетів. Проте даний підхід створює можливості для інтеграції з багатьма іншими системами, які містять корисні для відстеження навчального процесу дані, як для студентства НаУКМА, так і інших вишів.

Зокрема подальші шляхи інтеграції можуть включати доповнення системи версією для викладачів. Така версія містила б статистику по курсу, яка включає не тільки оцінки студентів, але й агреговані дані про час, який вони витрачають на цей курс. Інтеграція з університетською платформою має забезпечити цілісність та єдність даних про курси поміж студентів, що дозволить легко згрупувати дані про навчальні сесії за кодом курсу для створення статистики для викладачів.

ВИСНОВКИ

Отже, в рамках цієї роботи було досліджено методи для візуалізації даних та виокремлено оптимальні методи для візуалізації даних про навчальний час. Крім того, було досліджено можливості застосування LLM для аналізу зібраних системою даних та візуалізації інформативного звіту, а також було запропоновано метод інтеграції зі сторонніми системами та отримання певних даних навіть без ключів доступу.

Результатом роботи є вебзастосунок Study Time Tracker, який дозволяє користувачам відстежувати свої навчальні тенденції за допомогою аналітики, створеної на основі даних про навчальні предмети та завдання, які користувач заносить у систему або отримує автоматично завдяки інтеграції з DistEdu, та даних про затрачений час, які записуються після використання таймера чи секундоміра, або самостійно додаються користувачем на календарі. Для відображення аналітики було синтезовано та застосовано оптимальне поєднання різних методів візуалізації даних.

Подальше дослідження включає розширення даних реченнями завдань та розробку алгоритмів для виявлення впливу різних чинників (починаючи з часу до реченців, закінчуючи часом доби та тривалістю сесій) на продуктивність та оцінки, навчальних звичок тощо. Також заплановано впровадження елементів гейміфікації та дослідження їх впливу на мотивацію та залученість користувачів. Інші напрямки включають автоматизацію отримання даних про навчальний час за допомогою фонового відстеження активності користувача та дослідження відповідності цих даних дійсності.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

- [1] І. Б. Чичкало-Кондрацька і І. В. Кандзюба, «Дистанційне навчання та його вплив на особистісний розвиток», PhD Thesis, Національний університет "Полтавська політехніка імені Юрія Кондратюка", 2024.
- [2] Т. Ярошенко, О. Подденежний, О. Бершадська, Я. Вознюк, О. Слобусь, і Т. Бабич, «Методичні рекомендації щодо організації змішаного/дистанційного навчання в НаУКМА», 2021.
- [3] F. Cirillo, *The Pomodoro technique: The acclaimed time-management system that has transformed how we work*. Crown Currency, 2018.
- [4] А. О. Медведєва, «Використання елементів гейміфікації у навчанні польської мови», *Українські Студії В Європейському Контексті 36 Наук Пр 2024 № 8 382 С*, с. 233, 2024.
- [5] Є. В. Антонов, «Гейміфікація як засіб мотивації освітнього процесу», 2024.
- [6] C. Ware, *Information visualization: perception for design*. Morgan Kaufmann, 2019.
- [7] T. Munzner, *Visualization analysis and design*. CRC press, 2014.
- [8] В. В. Дядичев і К. М. Сафронов, «Методи візуалізації інформаційних структур», *Наукові Вісті Дніпровського Університету*, вип. 4, 2011.
- [9] B. Shneiderman, «The eyes have it: A task by data type taxonomy for information visualizations», в *The craft of information visualization*, Elsevier, 2003, с. 364–371.
- [10] S. K. Card, J. Mackinlay, і B. Shneiderman, *Readings in information visualization: using vision to think*. Morgan Kaufmann, 1999.
- [11] E. Boling, J. P. Brown, S. D. Ray, A. Erwin, і S. Kirkley, «Visual Design for Interactive Learning Tools: Representation of Time-Based Information.», 1998.
- [12] C. Ahlberg і E. Wistrand, «IVEE: An information visualization and exploration environment», в *Proceedings of Visualization 1995 Conference*, IEEE, 1995, с. 66–73.
- [13] О. О. Окунькова, «Візуалізація даних. Від простого до складного», *ВЧЕНІ ЗАПИСКИ*, с. 3202261, 2022.
- [14] Z. Gu, «Complex heatmap visualization», *iMeta*, вип. 1, вип. 3, с. e43, Вер 2022, doi: 10.1002/imt2.43.
- [15] B. Jelen і M. Alexander, *Pivot Table Data Crunching: Microsoft Excel 2010*. Pearson Education, 2010.
- [16] S. Caner і F. Bhatti, «A conceptual framework on defining businesses strategy for artificial intelligence», *Contemp. Manag. Res.*, вип. 16, вип. 3, с. 175–206, 2020.
- [17] A. Vaswani *et al.*, «Attention is all you need», *Adv. Neural Inf. Process. Syst.*, вип. 30, 2017.

- [18] L. Floridi i M. Chiriatti, «GPT-3: Its Nature, Scope, Limits, and Consequences», *Minds Mach.*, вип. 30, вип. 4, с. 681–694, Груд 2020, doi: 10.1007/s11023-020-09548-1.
- [19] S. Kublik i S. Saboo, *GPT-3: The ultimate guide to building NLP products with OpenAI API*. Packt Publishing Ltd, 2023.
- [20] J. White *et al.*, «A prompt pattern catalog to enhance prompt engineering with chatgpt», *ArXiv Prepr. ArXiv230211382*, 2023.
- [21] G. Marvin, N. Hellen, D. Jjingo, i J. Nakatumba-Nabende, «Prompt Engineering in Large Language Models», в *Data Intelligence and Cognitive Informatics*, I. J. Jacob, S. Piramuthu, i P. Falkowski-Gilski, Ред., в Algorithms for Intelligent Systems. , Singapore: Springer Nature Singapore, 2024, с. 387–402. doi: 10.1007/978-981-99-7962-2_30.
- [22] S. Schulhoff *et al.*, «The Prompt Report: A Systematic Survey of Prompt Engineering Techniques», 26, Лютий 2025, *arXiv*: arXiv:2406.06608. doi: 10.48550/arXiv.2406.06608.
- [23] S. Ekin, «Prompt Engineering For ChatGPT: A Quick Guide To Techniques, Tips, And Best Practices», 04, Травень 2023. doi: 10.36227/techrxiv.22683919.
- [24] «Content scripts | Chrome Extensions», Chrome for Developers. Дата звернення: 09, Квітень 2025. [Online]. Доступний у: <https://developer.chrome.com/docs/extensions/develop/concepts/content-scripts>
- [25] «chrome.webRequest | API», Chrome for Developers. Дата звернення: 09, Квітень 2025. [Online]. Доступний у: <https://developer.chrome.com/docs/extensions/reference/api/webRequest>
- [26] «chrome.debugger | API», Chrome for Developers. Дата звернення: 09, Квітень 2025. [Online]. Доступний у: <https://developer.chrome.com/docs/extensions/reference/api/debugger>
- [27] K. E. Ogundeyi i C. Yinka-Banjo, «WebSocket in real time application», *Niger. J. Technol.*, вип. 38, вип. 4, с. 1010–1020, 2019.