

Міністерство освіти і науки України
Кам'янець-Подільський національний університет імені Івана Огієнка
Фізико-математичний факультет
Кафедра комп'ютерних наук

Кваліфікаційна робота бакалавра
з теми: “Технологія автоматизованої системи для розподілу навчального навантаження між викладачами кафедри”

Виконав: здобувач вищої освіти групи KN1-B21
спеціальності 122 Комп'ютерні науки
Козловський Іван Вікторович

Керівник: Іванюк Віталій Анатолійович,
доктор технічних наук, доцент,
завідувач кафедри комп'ютерних наук

Рецензент:
Чорна Оксана Григорівна, кандидат педагогічних наук, доцент, завідувач навчального відділу

АНОТАЦІЯ

Козловський І.В. Технологія автоматизованої системи для розподілу навчального навантаження між викладачами кафедри. — Кваліфікаційна робота бакалавра, 2025.

Кваліфікаційна робота присвячена актуальній проблемі автоматизації процесу розподілу навчального навантаження в закладах вищої освіти. Метою роботи є створення вебінтерфейсу автоматизованої системи, яка забезпечує ефективний, прозорий та зручний механізм формування індивідуальних викладацьких карток з використанням хмарних технологій Google Sheets і Google Apps Script.

У ході роботи проведено аналіз існуючих методів розподілу навантаження, виявлено недоліки традиційних підходів та обґрунтовано доцільність автоматизації. Розроблено архітектуру системи, модель взаємодії з даними, а також реалізовано функціонал створення, заповнення, експорту та резервного копіювання викладацьких карток. Проведено функціональне тестування та підготовлено інструкцію для користувачів.

Практичне значення роботи полягає в можливості масштабування запропонованого рішення на інші підрозділи університету для покращення управлінських процесів та зменшення адміністративного навантаження.

Ключові слова: автоматизація, навчальне навантаження, Google Apps Script, вебінтерфейс, заклад вищої освіти, викладач, Google Sheets.

ABSTRACT

Kozlovskiy I.V. Web Interface for an Automated System of Teaching Workload Distribution among University Department Instructors. – Bachelor's qualification thesis, 2025.

The qualification thesis addresses the pressing issue of automating the process of distributing teaching workloads in higher education institutions. The aim of the work is to develop a web interface for an automated system that enables efficient, transparent, and user-friendly generation of individual instructor workload cards using Google Sheets and Google Apps Script.

The study includes an analysis of existing approaches, identifies the limitations of manual methods, and substantiates the need for automation. The system architecture, data interaction model, and the functionality for generating, filling, exporting, and backing up instructor workload cards were developed and implemented. Functional testing was conducted and a user manual was prepared.

The practical significance of the work lies in the potential for scaling the solution across other university departments, improving administrative efficiency and reducing manual workload.

Keywords: automation, teaching workload, Google Apps Script, web interface, higher education institution, instructor, Google Sheets.

ЗМІСТ

ВСТУП	4
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ОБҐРУНТУВАННЯ НЕОБХІДНОСТІ АВТОМАТИЗАЦІЇ РОЗПОДІЛУ НАВЧАЛЬНОГО НАВАНТАЖЕННЯ.....	6
1.1. Огляд існуючих підходів та систем розподілу навчального навантаження.	6
1.2. Вимоги функціонального та нефункціонального забезпечення ...	8
Висновки до розділу 1	11
РОЗДІЛ 2. ПРОЄКТУВАННЯ АВТОМАТИЗОВАНОЇ СИСТЕМИ.....	12
2.1 Архітектура системи та опис компонентів.....	12
2.2 Модель даних і інтеграція з Google Sheets / Apps Script.....	15
2.3. Проєктування UX/UI: макети й принципи взаємодії користувача	20
2.4. Забезпечення доступу, авторизація та безпека даних	23
2.5. Модель формування картки викладача	26
Висновки до розділу 2	28
РОЗДІЛ 3. РЕАЛІЗАЦІЯ ТА ВАЛІДАЦІЯ АВТОМАТИЗОВАНОЇ СИСТЕМИ	30
3.1. Структура проєкту, основні модулі й їх реалізація (створення, заповнення, видалення карток).....	30
3.2. Генерація PDF-звітів, автоматична відправка електронних листів і друк карток.	37
3.3. Методика функціонального і UX-тестування; результати	40
3.4. Підготовка до впровадження системи; інструкція користувача.	43
Висновки до розділу 3	45
ВИСНОВКИ.....	47
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	48

ВСТУП

У сучасних умовах цифрової трансформації закладів вищої освіти особливого значення набуває автоматизація адміністративних процесів, зокрема розподілу навчального навантаження між викладачами кафедр. Традиційні підходи, що базуються на ручному введенні та опрацюванні даних у табличних процесорах (Excel, Word), є малоефективними, спричиняють значні часові витрати та можуть призводити до помилок у плануванні. За відсутності централізованої системи ускладнюється контроль над обсягом навантаження, його рівномірним розподілом, а також унеможлиблюється швидке формування звітності.

Особливої актуальності ця проблема набуває у контексті переходу до електронного документообігу, вимог до прозорості та об'єктивності в управлінні навчальним процесом, а також зростаючих обсягів навчального контингенту. Використання хмарних технологій, зокрема сервісів Google, відкриває широкі можливості для створення доступних, гнучких та масштабованих систем управління навчальним навантаженням. Саме тому постає необхідність розробки сучасного вебінтерфейсу, що автоматизує ключові етапи формування, перегляду, редагування й затвердження викладацького навантаження.

Отже, основною задачею є створення функціональної автоматизованої системи, що дозволить оптимізувати та стандартизувати процес формування індивідуальних навчальних карток викладачів, з можливістю подальшого масштабування на рівень факультетів або університету.

Метою роботи є розробка автоматизованої системи для розподілу навчального навантаження між викладачами кафедри, реалізованого на базі сервісів Google Apps Script та Google Sheets.

Для досягнення поставленої мети необхідно вирішити наступні завдання:

- проаналізувати існуючі рішення та підходи;

- визначити функціональні та нефункціональні вимоги до системи;
- спроектувати архітектуру вебінтерфейсу та модель даних;
- реалізувати функціонал створення, заповнення, видалення та експорту карток викладачів;
- провести тестування системи й оцінити її ефективність;
- підготувати систему до впровадження в освітній процес.

Об'єкт дослідження: Процес організації та обліку навчального навантаження у закладах вищої освіти.

Предмет дослідження: Інформаційна система для автоматизації розподілу навчального навантаження між викладачами з використанням вебінтерфейсу та хмарних технологій.

Методи дослідження: у роботі використовуються такі методи:

- аналіз та узагальнення літературних джерел і нормативних документів;
- моделювання структури даних та взаємодії між компонентами системи;
- експериментальне тестування функціоналу системи;
- UX-оцінювання з використанням критеріїв юзабіліті.

Практичне значення одержаних результатів: Результати дослідження мають прикладне значення для закладів вищої освіти, оскільки дозволяють спростити процедуру формування навчального навантаження, зменшити ймовірність помилок, а також забезпечити зручний інтерфейс для роботи з документами. Розроблена система може бути використана як у межах однієї кафедри, так і масштабована для застосування на факультетському рівні.

Результати роботи використовуються в роботі кафедри комп'ютерних наук Кам'янець-Подільського національного університету імені Івана Огієнка.

Структура роботи : Кваліфікаційна робота складається зі вступу, трьох розділів, висновків, списку використаних джерел та додатків. У першому розділі подано теоретичне обґрунтування теми та огляд аналогічних рішень. У другому — здійснено проектування системи: описано архітектуру, модель даних, макети інтерфейсу та безпеку. У третьому розділі розглянуто реалізацію, тестування, результати роботи і підготовку до впровадження.

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ОБГРУНТУВАННЯ НЕОБХІДНОСТІ АВТОМАТИЗАЦІЇ РОЗПОДІЛУ НАВЧАЛЬНОГО НАВАНТАЖЕННЯ

1.1. Огляд існуючих підходів та систем розподілу навчального навантаження.

Сучасні підходи до розподілу навчального навантаження умовно можна поділити на кілька основних категорій залежно від рівня автоматизації та типу використовуваних програмних продуктів [1].

Перший підхід – це традиційний ручний спосіб із використанням паперових документів або простих електронних таблиць (переважно Microsoft Excel). Цей спосіб досі є поширеним у багатьох закладах вищої освіти через свою простоту й низький поріг входження. Водночас, він має значні недоліки: велика кількість рутинних операцій, висока ймовірність помилок під час копіювання та внесення даних, відсутність можливості одночасної роботи кількох користувачів, складність швидкого внесення змін і коригувань. Крім того, використання статичних таблиць призводить до постійної необхідності повторних перевірок і контролю, що у свою чергу знижує загальну продуктивність адміністративних працівників.

Другий підхід пов'язаний з використанням спеціалізованих ERP-систем (Enterprise Resource Planning), які пропонують модулі, спеціально призначені для управління навчальними закладами. До таких систем належать Oracle PeopleSoft, SAP ERP, 1С-Університет тощо. ERP-рішення дозволяють інтегрувати різноманітні процеси університету, зокрема планування розкладу, формування звітів і контроль виконання планів. Перевагами цих систем є централізація даних, єдине інформаційне поле для всіх підрозділів, автоматизовані процеси генерації документів і формування звітів. Водночас,

ERP-системи характеризуються складністю впровадження та високою вартістю обслуговування, що робить їх доступними переважно для великих університетів із достатнім бюджетом на інформаційні технології.

Третій підхід передбачає застосування спеціалізованих програмних продуктів і платформ для автоматизації навчального навантаження, таких як WorkloadPro, UniTime, Timetabler та інші. Ці системи мають гнучкі можливості налаштування алгоритмів розподілу, зручні інтерфейси для створення графіків і можливість формування персоналізованих звітів. Вони дозволяють автоматизовано враховувати педагогічне навантаження, оптимізувати розклад та забезпечувати баланс навантаження викладачів. Недоліками подібних систем часто є необхідність локального серверного розгортання, підтримки додаткової інфраструктури, а також відносно високий поріг входження для адміністративного персоналу без технічних навичок. Крім того, такі рішення нерідко є платними й потребують постійної технічної підтримки, що не завжди доступно кафедрам з обмеженими ресурсами.

Четвертий підхід охоплює хмарні рішення на базі популярних онлайн-сервісів (Google Sheets, Microsoft Office 365, Airtable та ін.). Такі системи дозволяють забезпечити одночасну спільну роботу кількох користувачів, підтримують централізоване зберігання даних та версіонування змін, автоматизують багато рутинних процесів, включаючи формування звітів та розсилку сповіщень. Переваги цього підходу полягають у простоті розгортання, низьких фінансових витратах і високій гнучкості налаштування. При цьому хмарні рішення дозволяють легко інтегруватися з іншими інструментами та сервісами, що використовуються університетом. Проте, для ефективного впровадження таких систем потрібні певні знання у сфері налаштування хмарних інструментів і вміння роботи з програмуванням (наприклад, через скриптові розширення).

Аналізуючи перелічені підходи, можна дійти висновку, що традиційні методи вже не відповідають сучасним вимогам щодо швидкості та якості управління освітніми процесами. ERP-системи, попри їх потужні можливості,

не завжди виправдовують високі витрати, складність та тривалість впровадження. Спеціалізовані програмні рішення потребують додаткової ІТ-інфраструктури та часто складні у використанні адміністративним персоналом. Водночас хмарні рішення на базі Google Sheets та Google Apps Script поєднують у собі гнучкість, доступність і достатній рівень автоматизації, що робить їх найбільш перспективними для застосування в середніх і малих університетах, а також на окремих кафедрах великих навчальних закладів.

Таким чином, актуальним і доцільним є розробка веборієнтованої автоматизованої системи на основі хмарних технологій Google, що дозволить поєднати простоту використання, доступність впровадження, ефективну автоматизацію рутинних процесів і зручність спільної роботи. Вибір саме такого рішення забезпечить суттєве скорочення часових витрат адміністративних працівників, підвищить точність і прозорість розподілу навчального навантаження, а також дозволить оперативно реагувати на зміни у навчальних планах і кадровому складі кафедр.

1.2. Вимоги функціонального та нефункціонального забезпечення

При розробці будь-якої автоматизованої системи важливо чітко визначити перелік вимог, які висуваються до її функціонального та нефункціонального забезпечення. Це дозволяє забезпечити якість кінцевого продукту, зручність його використання та відповідність поставленим задачам. Зокрема, вимоги до автоматизованої системи розподілу навчального навантаження між викладачами кафедр слід розділити на дві групи: функціональні та нефункціональні [2].

Функціональні вимоги визначають набір основних функцій, які повинна забезпечувати система для реалізації процесу розподілу навчального навантаження. До ключових функціональних вимог належать:

1. Створення, редагування та видалення карток викладачів. Система повинна забезпечувати зручний механізм додавання нових викладачів, а також

швидке внесення змін і коригувань у вже наявні картки. При цьому кожна картка має містити персональну інформацію (прізвище, ім'я, по-батькові, посада), навчальні дисципліни, кількість годин, тип занять, додаткові методичні та організаційні обов'язки.

2. Централізоване зберігання даних. Усі дані про викладачів, дисципліни, години та інші параметри повинні бути доступні в єдиному централізованому сховищі (Google Sheets), що дозволяє забезпечити цілісність і достовірність інформації.

3. Автоматична генерація звітів. Система повинна мати можливість автоматично формувати PDF-документи на основі введених даних. Це можуть бути індивідуальні картки навантаження викладачів, загальні звіти по кафедрі, зведені таблиці навантаження, які використовуються для внутрішнього контролю та звітності перед керівництвом.

4. Автоматичні оповіщення користувачів. Важливо, щоб система могла надсилати електронні повідомлення з використанням Gmail API у випадках внесення змін до розкладу, формування звітів або інших важливих подій. Це дозволяє оперативно інформувати викладачів і адміністративний персонал кафедри про будь-які зміни або оновлення.

5. Можливість друку карток. Користувачі повинні мати змогу легко друкувати необхідні документи безпосередньо з веб-інтерфейсу без додаткових налаштувань або конвертацій.

6. Гнучке налаштування шаблонів і звітних форм. Система має передбачати можливість легкої адаптації шаблонів карток і звітів до специфіки окремих кафедр або факультетів без змін у програмному коді.

7. Одночасна спільна робота. Необхідно забезпечити можливість одночасного редагування та перегляду даних кількома користувачами, що дозволяє уникнути конфліктів при одночасній роботі.

Нефункціональні вимоги характеризують загальні атрибути якості системи, що безпосередньо не пов'язані з виконанням функціональних задач,

але суттєво впливають на ефективність її експлуатації. До основних нефункціональних вимог належать такі:

1. Зручність та інтуїтивність інтерфейсу (Usability). Вебінтерфейс повинен бути максимально простим і зрозумілим для користувачів без технічних навичок, що дозволить мінімізувати час на навчання персоналу.

2. Швидкодія та продуктивність (Performance). Система повинна забезпечувати швидкий відгук на запити користувачів навіть за умови великого обсягу даних, що дозволить уникнути затримок під час роботи.

3. Надійність і стабільність (Reliability). Система повинна бути стабільною і надійною, з мінімальним ризиком втрати даних або збоїв під час виконання критичних операцій.

4. Захищеність даних (Security). Важливо забезпечити захист персональних даних викладачів і конфіденційність інформації шляхом застосування авторизації через корпоративні акаунти Google та розподілу доступу за ролями.

5. Адаптивність (Adaptability). Вебінтерфейс повинен бути адаптивним для зручної роботи з різних пристроїв (ПК, ноутбуків, планшетів і смартфонів), що дозволить користувачам працювати з системою незалежно від місця перебування.

6. Масштабованість (Scalability). Система має легко адаптуватися до зростання обсягу інформації, кількості користувачів і складності операцій без значних змін у її архітектурі.

7. Підтримка (Maintainability). Повинна бути можливість легкої підтримки і модернізації системи, внесення нових функцій і змін без суттєвих витрат ресурсів.

Таким чином, визначення чітких функціональних і нефункціональних вимог на етапі проектування дозволяє створити ефективне рішення, яке максимально відповідає потребам сучасних університетів. Врахування зазначених вимог забезпечує високу якість роботи розробленого вебінтерфейсу, полегшує процес його впровадження, а також забезпечує

комфортну та продуктивну роботу кінцевих користувачів – співробітників кафедр і адміністративного персоналу [3].

Висновки до розділу 1

На основі проведеного аналізу предметної області встановлено, що традиційні ручні методи та існуючі напівавтоматизовані підходи до розподілу навчального навантаження у закладах вищої освіти характеризуються значними часовими затратами, високою ймовірністю помилок, складністю внесення оперативних змін та недостатньою прозорістю. Розглянутий порівняльний аналіз наявних програмних рішень, включаючи ERP-системи та спеціалізовані додатки, підтвердив, що вони часто є дорогавартісними, складними у впровадженні та не завжди достатньо гнучкими для адаптації до унікальних потреб конкретної кафедри чи університету.

Таким чином, виникла обґрунтована потреба у розробці нової автоматизованої системи, здатної ефективно вирішити виявлені проблеми, забезпечити більшу точність, прозорість та гнучкість процесу розподілу навчального навантаження. Для забезпечення її високої якості та відповідності цілям були чітко сформульовані функціональні та нефункціональні вимоги, які стануть основою для подальшого проектування та розробки системи. Це дозволить створити інструмент, що оптимізує використання кадрових ресурсів, підвищить мотивацію викладачів та загальну ефективність освітнього процесу.

РОЗДІЛ 2. ПРОЄКТУВАННЯ АВТОМАТИЗОВАНОЇ СИСТЕМИ

2.1 Архітектура системи та опис компонентів

Автоматизована система розподілу навчального навантаження, представлена у цій роботі, побудована на основі хмарних технологій Google, а саме – Google Sheets і Google Apps Script. Вона створена з метою спрощення процесу планування та розподілу навантаження між викладачами кафедр, а також забезпечення ефективною взаємодією користувачів без необхідності спеціалізованих технічних знань.

Загальна архітектура системи складається з кількох логічних компонентів, які тісно взаємодіють між собою:

- Google Sheets – використовується як центральне сховище даних [8].
- Google Apps Script – забезпечує програмну логіку системи та взаємодію між таблицями й веб-інтерфейсом [7].
- Вебінтерфейс користувача – забезпечує простий та зрозумілий доступ до всіх функцій системи.

Кожен із цих компонентів має власну роль і функції, взаємодія між ними забезпечує повноцінне функціонування системи.

Google Sheets як сховище даних

Для зберігання та управління даними використовується таблиця Google Sheets, структура якої складається з кількох спеціалізованих листів:

- *SV* – цей лист містить загальну інформацію про викладачів (імена, коди, електронні адреси). Це базовий аркуш, з якого система бере інформацію для подальших операцій.
- *K_d_1* та *K_d_2* – листи для інформації про курсові роботи, практики та інші позанавчальні активності викладачів для першого та другого семестрів відповідно.

- *N_d_s1* та *N_d_s2* – листи, які зберігають інформацію про навчальні дисципліни першого та другого семестрів денної форми навчання.
- *N_d_z1* та *N_d_z2* – аналогічні листи, що містять дисципліни першого та другого семестрів заочної форми навчання.
- *Roz2* – аркуш, де узагальнюється інформація про розподілене навантаження всіх викладачів кафедри.

Така структура забезпечує чітке розмежування даних за призначенням, що полегшує їх подальше використання та керування.

Google Apps Script як серверна логіка системи

Серверна логіка реалізована за допомогою Google Apps Script – хмарної платформи JavaScript, яка дозволяє автоматизувати взаємодію з Google-таблицями. Основні завдання, що реалізуються через Google Apps Script:

- Обробка запитів від вебінтерфейсу користувача.
- Генерація документів у форматі PDF на основі створених карток навантаження викладачів.
- Автоматична розсилка електронних листів із вкладеннями (картки навантаження) на електронні адреси викладачів.
- Створення та керування файлами й папками у Google Drive (наприклад, автоматичне створення копій таблиць з фіксацією часу створення).

Скрипти забезпечують логіку таких операцій, як створення, заповнення, редагування, видалення карток навантаження, а також автоматичне створення звітних PDF-файлів і відправлення їх електронною поштою чи на друк.

Вебінтерфейс користувача

Вебінтерфейс є головною точкою взаємодії користувачів із системою. Він має зрозумілу структуру і надає доступ до всіх основних функцій, що забезпечує простоту користування навіть для людей без спеціальних технічних навичок. Інтерфейс містить такі компоненти та функції:

- Обрати таблицю – дозволяє користувачу обрати Google-таблицю, з якою він планує працювати.

- Додати викладачів до списку – після натискання цієї кнопки список викладачів із вибраної таблиці завантажується та відображається в інтерфейсі.
- Створити картку викладача – створює окремі аркуші для викладачів, які користувач вибрав у списку, та готує їх для заповнення.
- Заповнити картки викладачів – автоматично заповнює створені картки інформацією про навчальне навантаження, яка береться з основних аркушів.
- Видалити створені картки та Видалити заповнення картки – ці функції дозволяють легко виправити помилки та переробити створені раніше документи.
- Відправити картки на друк – автоматично генерує PDF-картки та зберігає їх у спеціально створеній папці на Google Drive.
- Відправити на пошту – автоматично надсилає сформовані картки на електронні адреси викладачів.
- Створити копію таблиці – створює резервну копію таблиці з поточним станом, що дозволяє уникнути втрат інформації під час подальших редагувань.

Кожна з цих функцій викликає відповідні скрипти Google Apps Script, які працюють безпосередньо з Google Sheets та Drive, забезпечуючи ефективну автоматизацію робочих процесів.

Схема взаємодії компонентів системи

Загальна взаємодія компонентів системи виглядає так:

- Користувач відкриває вебінтерфейс, авторизується через свій Google-акаунт.
- За допомогою інтерфейсу він вибирає потрібну таблицю (Google Sheets), з якою система буде взаємодіяти.
- Після цього Google Apps Script отримує дані з вибраної таблиці, обробляє їх, створює необхідні аркуші, картки, звіти та генерує PDF-документи.

- Готові документи автоматично зберігаються на Google Drive та/або надсилаються на електронні адреси користувачів.
- Всі зміни в таблицях автоматично зберігаються з історією версій, що забезпечує надійність роботи системи.

Таким чином, архітектура системи забезпечує максимально зручну, швидку й надійну взаємодію користувачів із даними, що зберігаються в Google-таблицях, автоматизує рутинні процеси та значно скорочує час адміністрування розподілу навчального навантаження [4].

2.2 Модель даних і інтеграція з Google Sheets / Apps Script

Важливим етапом проєктування автоматизованої системи розподілу навчального навантаження є розробка моделі даних, яка забезпечує впорядковане та ефективне зберігання інформації, необхідної для роботи системи. У даному пункті детально описується модель даних, що використовується в системі, а також процес інтеграції цієї моделі з інструментами Google Sheets та Google Apps Script [5].

Модель даних

Модель даних системи реалізовано у вигляді табличних структур у сервісі Google Sheets. Вона розподілена між кількома логічними аркушами, кожен з яких виконує певну функцію та відповідає за окрему категорію даних.

До складу моделі даних входять такі основні аркуші:

- *Аркуш «SV» (список викладачів)* – є ключовим елементом моделі, оскільки він містить базову інформацію про викладацький склад кафедри. В ньому зберігаються прізвище, ім'я, по батькові, посада, код викладача та його електронна адреса, яка використовується для розсилки повідомлень та карток навантаження (рис. 2.1).

A	B	C	D	E	F
q	w		r	t	
a	s		f	g	
Викладачі	ШИФР	Пошта	План	ставка	всього годі
Бараннік В.В., професор	БВВ	@kpnpu.edu.ua	0,15	600	90,0
Баранник В.В., професор	БВВ1	@kpnpu.edu.ua			0,0
Філатов А., фахівець-практик	ФАС	@kpnpu.edu.ua	0,14	600	84,0
Федорчук В.А., професор	ФВА	@kpnpu.edu.ua	1	560	560,0
Федорчук В.А., професор, гарант	ФВА1	@kpnpu.edu.ua	0,06	600	36,0
Громик А.П.	ГАП	@kpnpu.edu.ua	0	570	0,0
Гончар Д., фахівець-практик	ГДО	@kpnpu.edu.ua	0	600	0,0
Іванюк В.А., доцент, гарант, завідувач	ІВА	@kpnpu.edu.ua	1,00	530	530,0
Іванюк В.А., доцент, гарант, завідувач	ІВА1	@kpnpu.edu.ua	0,37	600	222,0
Іванюк В.А., доцент, гарант, завідувач	ІВА2	@kpnpu.edu.ua			0,0
Ковцун А.Г., асистент	КАГ	@kpnpu.edu.ua	0,1	600	60,0
Кушнір О.К., доцент	КОК	@kpnpu.edu.ua	0,05	600	30,0
Кух О.М., асистент	КОМ	@kpnpu.edu.ua	1	600	600,0
Кух О. М., асистент	КОМ1	@kpnpu.edu.ua	0,1	600	60,0
Мястковська М.О., старший викладач	ММО	@kpnpu.edu.ua	1	600	600,0
Мястковська М.О., старший викладач	ММО1	@kpnpu.edu.ua	0,33	600	198,0
Моцик Р.В., доцент	МРВ	@kpnpu.edu.ua	1	590	590,0
Моцик Р.В., доцент	МРВ1	@kpnpu.edu.ua	0,25	600	150,0
Мендограло В.В., заступник міського голови	МВВ	@kpnpu.edu.ua	0	600	18,0
Оптасюк С.В., доцент	ОСВ	@kpnpu.edu.ua	0,15	600	90,0
Понеділок І.С., асистент	ПІС	@kpnpu.edu.ua	1	600	600,0
Пилипюк Т.М., доцент, гарант	ПТМ	@kpnpu.edu.ua	1	590	590,0
Пилипюк Т.М., доцент	ПТМ1	@kpnpu.edu.ua	0,35	600	210,0
Понеділок В. В., фахівець-практик	ПВВ	@kpnpu.edu.ua	0,25	600	150,0
Романюк В.М., асистент	РВМ	@kpnpu.edu.ua	0,05	600	32,0
Смалько О.А., доцент	СОА	@kpnpu.edu.ua	1	590	590,0
Смалько О.А., доцент	СОА1	@kpnpu.edu.ua	0,24	600	144,0
Слободянюк О. В., старший викладач	СОВ	@kpnpu.edu.ua	1	590	590,0
Слободянюк О. В., старший викладач	СОВ1	@kpnpu.edu.ua	0	600	0,0
Сидорук В.А., старший науковий співробітник	СВА	@kpnpu.edu.ua	0,09	600	54,0
Щирба В. С., професор	ЩВС	@kpnpu.edu.ua	0,7	590	413,0
Щирба В. С., професор	ЩВС1	@kpnpu.edu.ua	0,06	600	36,0

Рис. 2.1. Аркуш «SV» (список викладачів).

- Аркуші «K_d_1» та «K_d_2» (курсів роботи та інші активності) – ці аркуші зберігають інформацію про позанавчальні активності викладачів (наприклад, керівництво курсовими роботами, практиками, дипломними проєктами тощо). Дані поділені за семестрами, що дозволяє чітко та оперативно формувати навантаження на кожен навчальний період (рис. 2.2).

Інтеграція системи з Google Sheets передбачає використання вбудованих можливостей цього сервісу для зберігання, спільного редагування, версіонування та безпечного зберігання інформації. Google Sheets дозволяє адміністраторам кафедри зручно додавати та редагувати інформацію, а також контролювати історію змін, що є надзвичайно важливим для забезпечення прозорості та відповідальності в процесі розподілу навантаження [9].

Google Sheets підтримує одночасну спільну роботу багатьох користувачів, забезпечуючи можливість швидкого внесення коригувань, що значно спрощує процес узгодження та затвердження розподілу навантаження. Крім того, Google Sheets легко інтегрується з іншими сервісами Google, такими як Google Drive для зберігання PDF-документів та Gmail для розсилки повідомлень.

Інтеграція з Google Apps Script

Для забезпечення автоматизації всіх процесів, пов'язаних з роботою із даними, було використано платформу Google Apps Script. Це середовище дозволяє програмувати логіку взаємодії з даними таблиць безпосередньо у хмарному середовищі Google.

Інтеграція з Google Apps Script забезпечує наступні ключові функції системи:

- Автоматичне створення аркушів викладачів – для кожного викладача, обраного в вебінтерфейсі, створюється окремий аркуш, який потім автоматично заповнюється даними про його навантаження.
- Автоматичне заповнення карток навантаження – Apps Script обробляє інформацію з аркушів «K_d», «N_d» та автоматично заповнює відповідні картки викладачів.
- Формування PDF-документів – на основі створених карток навантаження формуються PDF-файли, які потім автоматично зберігаються в Google Drive.

- Розсилка електронних повідомлень – автоматичне надсилання PDF-карток навантаження викладачам на електронні адреси, зазначені в аркуші «SV».
- Резервне копіювання – можливість автоматично створювати резервні копії таблиці з відміткою часу для збереження робочого стану на момент копіювання.

Взаємодія Google Apps Script із Google Sheets відбувається за допомогою вбудованих API, що дозволяють зчитувати, обробляти, записувати та передавати інформацію між таблицями та іншими сервісами Google. Завдяки цьому забезпечується повна автоматизація процесу розподілу навантаження, значно скорочуються часові витрати персоналу та підвищується точність даних.

Таким чином, запропонована модель даних та її інтеграція з інструментами Google Sheets і Google Apps Script створюють ефективну, гнучку та просту в користуванні систему, яка дозволяє максимально автоматизувати та оптимізувати процес управління навчальним навантаженням на кафедрах університету.

2.3. Проектування UX/UI: макети й принципи взаємодії користувача

Проектування інтерфейсів користувача (User Experience та User Interface – UX/UI) є одним із ключових етапів розробки веборієнтованих систем, оскільки від цього залежить зручність роботи з додатком, ефективність взаємодії та загальне сприйняття продукту кінцевими користувачами. Саме тому під час розробки вебінтерфейсу автоматизованої системи розподілу навчального навантаження значну увагу було приділено створенню зрозумілого, інтуїтивного та привабливого дизайну [10].

Принципи проектування інтерфейсу

Під час розробки дизайну були використані сучасні принципи UX/UI-дизайну:

- Інтуїтивність – всі елементи управління чітко позначені та легко зрозумілі навіть для користувачів, які вперше працюють із програмою.
- Зручність навігації – мінімальна кількість кліків для виконання стандартних задач (наприклад, створення чи заповнення карток).
- Адаптивність – інтерфейс зручний у використанні на пристроях із різними розмірами екранів.
- Лаконічність – відсутність зайвих елементів, які можуть відволікати користувача.
- Послідовність – єдиний стиль оформлення всіх елементів інтерфейсу для уникнення плутанини.

Опис та аналіз UX/UI-макету програми

Початковим етапом проектування інтерфейсу було створення макета в програмі Figma, що дозволило попередньо візуалізувати вигляд і основні сценарії взаємодії користувачів із системою (рис 2.6).

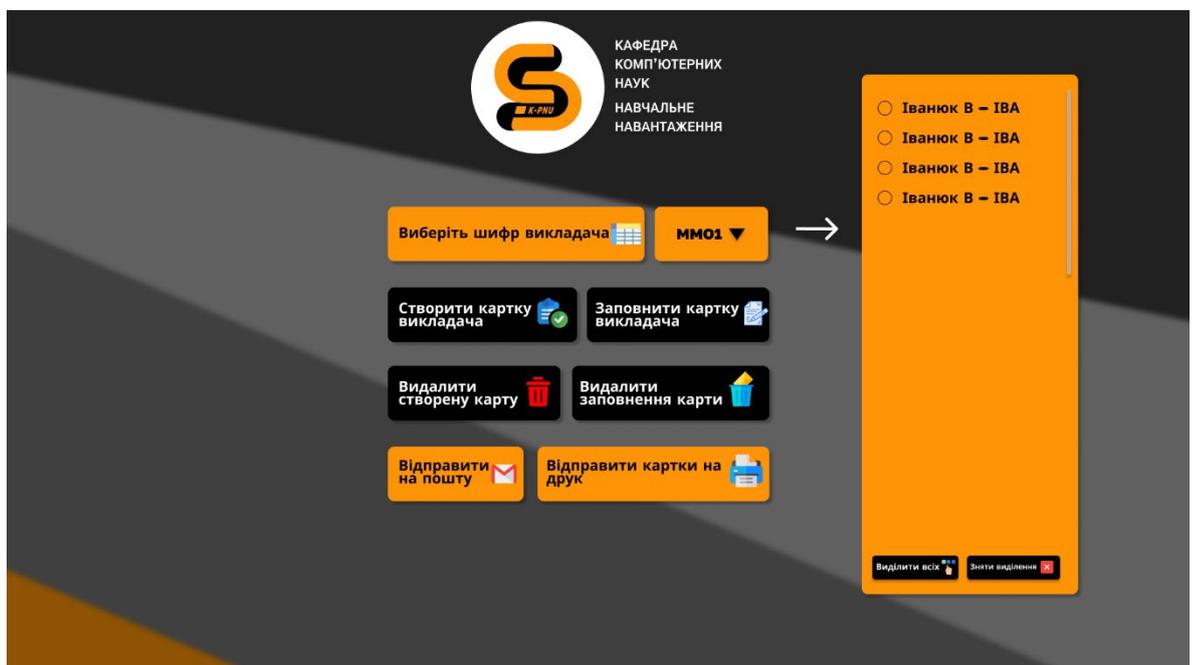


Рисунок 2.6 – Прототипу програми створений у Figma.

На основі створеного прототипу було передбачено такі основні елементи інтерфейсу (як наведено на скріншоті):

- Головна панель управління – центральна частина інтерфейсу, яка включає основні функціональні кнопки для роботи з системою. Панель

управління виконана у привабливих контрастних кольорах (чорний, помаранчевий), що чітко виділяє активні кнопки.

- Кнопка вибору шифру викладача – відкриває випадючий список, де користувач може швидко вибрати потрібного викладача для подальших операцій.
- Функціональні кнопки – кнопки для створення, заповнення, видалення карток та їх заповнення, а також для відправки документів на пошту чи друк. Кожна кнопка супроводжується зрозумілими іконками, що додатково покращує сприйняття та інтуїтивність інтерфейсу.
- Список викладачів – окремий елемент праворуч, який з'являється після натискання кнопки «Додати викладачів до списку». Цей список дозволяє користувачеві швидко вибрати одного чи кількох викладачів для створення або заповнення карток навантаження. Нижче списку знаходяться кнопки «Виділити всіх» та «Зняти виділення», що зручно під час роботи з великою кількістю викладачів.

Основні сценарії взаємодії користувачів з інтерфейсом

На основі документації користувача та макету в Figma можна виділити кілька ключових сценаріїв взаємодії:

Сценарій 1. Початок роботи та вибір таблиці

Користувач спочатку вибирає потрібну Google-таблицю для роботи з нею. Це здійснюється натисканням кнопки «Обрати таблицю». Після цього з'являється вікно з можливістю вибору таблиці.

Сценарій 2. Додавання викладачів до списку

Після вибору таблиці користувач натискає кнопку «Додати викладачів до списку». У правій частині інтерфейсу з'являється список викладачів, отриманий із таблиці. Тут є можливість зручного виділення необхідних викладачів.

Сценарій 3. Створення та заповнення карток

Виділивши потрібних викладачів, користувач натискає «Створити картку викладача». Далі, після створення карток, виконується їх автоматичне заповнення натисканням кнопки «Заповнити картку викладача».

Сценарій 4. Відправка та друк

Користувач може одразу після заповнення натиснути «Відправити картки на пошту», після чого сформовані PDF-файли автоматично будуть надіслані викладачам. Аналогічно, за допомогою кнопки «Відправити картки на друк» формується папка з PDF-файлами на Google Drive.

Сценарій 5. Управління версіями

Користувач також має змогу створити резервну копію поточної версії таблиці, натиснувши кнопку «Створити копію таблиці». Це дозволяє уникнути втрати інформації під час редагування.

Візуальний стиль і оформлення інтерфейсу

Графічний стиль інтерфейсу відповідає сучасним тенденціям дизайну вебдодатків: використовуються яскраві акценти (помаранчевий колір для основних кнопок), контрастний фон для зручності читання та мінімалістичний підхід у використанні елементів управління. Іконки та графічні позначення допомагають швидко зорієнтуватися у функціональних можливостях програми, що робить інтерфейс максимально зручним для широкого кола користувачів, незалежно від їхніх технічних навичок.

Таким чином, завдяки продуманому підходу до UX/UI-проектування було створено інтуїтивний і функціональний інтерфейс, який значно спрощує та прискорює процес взаємодії користувачів із системою, дозволяючи швидко й ефективно виконувати завдання з розподілу навчального навантаження між викладачами кафедри.

2.4. Забезпечення доступу, авторизація та безпека даних

В умовах впровадження автоматизованих інформаційних систем, особливо тих, що пов'язані з персональними даними співробітників навчальних закладів, питання забезпечення доступу, авторизації та захисту

інформації набувають особливої ваги. Саме тому у процесі проектування та реалізації автоматизованої системи розподілу навчального навантаження були враховані сучасні вимоги до безпеки даних та організації доступу до них.

Організація доступу користувачів

Система передбачає централізоване управління доступом до інформації через хмарні сервіси Google. Це дозволяє чітко контролювати, хто саме та які дані може переглядати чи редагувати. Всі користувачі для роботи з вебінтерфейсом мають бути авторизовані за допомогою власних Google-акаунтів, що суттєво підвищує безпеку та унеможлиблює доступ сторонніх осіб.

Для цього в системі реалізований механізм, що автоматично перевіряє користувачів за адресою електронної пошти, яка має бути заздалегідь додана до списку дозволених користувачів. Адміністратор кафедри або відповідальна особа може легко керувати списком авторизованих користувачів через налаштування доступу в Google Sheets.

Розподіл ролей користувачів та їхні права доступу

З метою забезпечення безпечного та впорядкованого управління доступом до даних реалізовано модель розподілу ролей користувачів. У системі визначено такі основні ролі:

- Адміністратор – має повний доступ до всіх функцій системи, включаючи редагування таблиць, створення й видалення карток, управління користувачами, резервне копіювання даних.
- Редактор (асистент) – може створювати, заповнювати й редагувати картки навантаження, але не має права керувати доступом інших користувачів.
- Переглядач – має право лише переглядати сформовані дані та звіти без можливості внесення змін.

Такий підхід дозволяє мінімізувати ризики випадкових або навмисних змін важливих даних та сприяє впорядкованості процесу управління навчальним навантаженням.

Авторизація користувачів у вебінтерфейсі

Авторизація в системі реалізована за допомогою стандартної авторизації через Google OAuth, що забезпечує високий рівень безпеки та зручність для користувачів. Після переходу за посиланням до вебінтерфейсу користувач автоматично переадресовується на сторінку авторизації Google, де має увійти за допомогою свого корпоративного облікового запису.

Після успішної авторизації користувач отримує доступ до інтерфейсу та функцій, які відповідають його ролі та правам доступу. Це дозволяє легко та швидко надати або обмежити доступ до інформації відповідно до вимог безпеки [11].

Захист персональних даних та безпека інформації

З огляду на те, що система оперує персональними даними викладачів (імена, посади, електронні адреси), було приділено значну увагу питанням захисту та конфіденційності. Система забезпечує такі заходи безпеки:

- Зберігання даних у хмарі Google Sheets, яке гарантує високий рівень захисту від втрати інформації завдяки автоматичному резервному копіюванню та версіонуванню змін.
- Використання Google Drive для зберігання PDF-файлів із картками викладачів, що забезпечує безпечний доступ до цих документів.
- Автоматичне журналювання дій користувачів (створення, зміни та видалення даних), що дозволяє оперативно реагувати на будь-які несанкціоновані або помилкові дії.
- Шифрування даних при передачі через захищені канали зв'язку (протокол HTTPS), який використовується під час роботи з вебінтерфейсом та Google-сервісами.
- Використання корпоративної авторизації Google для доступу до системи гарантує, що доступ отримують тільки авторизовані користувачі з відповідного домену організації.

Таким чином, завдяки інтеграції з Google-сервісами, автоматизована система розподілу навчального навантаження забезпечує високий рівень

безпеки, чіткий розподіл прав доступу та захист персональних даних користувачів, відповідаючи сучасним вимогам до безпечного зберігання й обробки інформації.

2.5. Модель формування картки викладача

Автоматизоване розподілення навчального навантаження в межах розробленої системи реалізується за допомогою алгоритмічного аналізу даних у Google Sheets з використанням Google Apps Script. Система працює за принципом модульного заповнення викладацьких карток на основі даних, попередньо впорядкованих у шаблонних аркушах [13].

Основу логіки становить функція fillTeacherCards, яка опрацьовує список вибраних шифрів викладачів. Для кожного з них створюється або оновлюється індивідуальний аркуш-картка, в який автоматично переносяться всі відповідні дані з таких джерел:

- SV — базові дані про викладача та загальне річне навантаження.
- K_d_1, K_d_2 — дисципліни кафедрального рівня (з деталізацією типу занять і додатковою інформацією).
- N_d_s1, N_d_s2 — денна форма навчання за семестрами.
- N_d_z1, N_d_z2 — заочна форма навчання за семестрами.

Алгоритм виконує такі основні кроки:

1. Ідентифікація аркуша викладача за шифром. Якщо аркуш відсутній, система генерує повідомлення про помилку.
2. Зчитування персональних даних з аркуша SV (ПІБ, загальне навантаження, шифр).
3. Фільтрація й запис дисциплін відповідно до шифру з кожного джерельного листа. Дані заповнюються построчно, з динамічним відстеженням номеру дисципліни.
4. Класифікація навантаження за типами: лекції, практичні, лабораторні заняття, консультації, курсові роботи тощо. Значення заносяться в окремі колонки викладацької картки відповідно до умов коду.

5. Післяобробка: видалення порожніх рядків, нормалізація кодів, обробка блоку “разом” з підсумковими годинами.

6. Синхронізація з Roz2 — підсумкові дані автоматично копіюються у загальний зведений лист для аналітики або звітності.

Блок-схема алгоритму заповнення картки викладача (Рис 2.7).

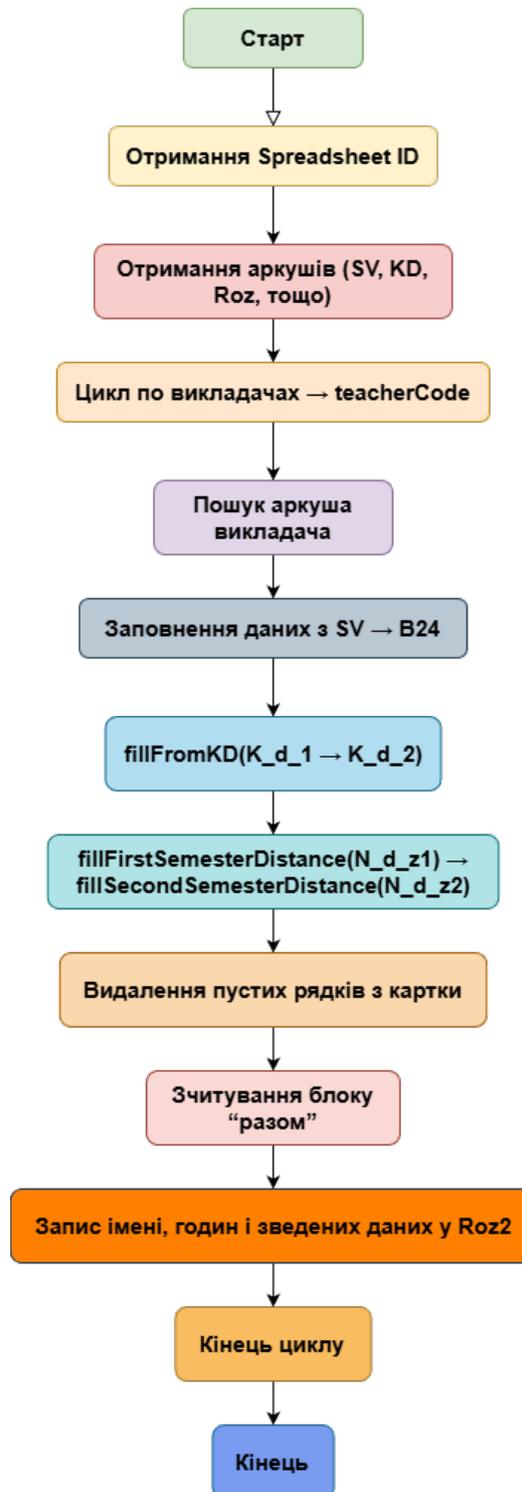


Рисунок 2.7 – Блок-схема алгоритму заповнення картки викладача.

Ключовою перевагою реалізованої моделі є її модульність та адаптивність: обробка здійснюється за конкретним викладачем, що дозволяє легко масштабувати систему на інші кафедри чи факультети. Наявність підфункцій для кожного джерела даних (`fillFromKD`, `fillFirstSemester`, `fillSecondSemester`, тощо) забезпечує гнучку архітектуру та полегшує модифікацію алгоритму в разі зміни структури таблиць.

Таким чином, розроблений алгоритм реалізує детермінований, структурований підхід до формування викладацьких навантажень з мінімальним втручанням користувача, що значно підвищує ефективність та прозорість усього процесу [12].

Висновки до розділу 2

У другому розділі було здійснено комплексне проектування автоматизованої системи розподілу навчального навантаження, що базується на хмарних технологіях Google. Запропонована архітектура системи включає взаємодію ключових компонентів: Google Sheets як центрального сховища даних, Google Apps Script для реалізації програмної логіки, а також вебінтерфейсу користувача для зручної взаємодії. Детально описана модель даних, що розподілена між спеціалізованими аркушами Google Sheets, забезпечує структуроване та цілісне зберігання інформації про викладачів, дисципліни та інші параметри навантаження.

Особливу увагу приділено проектуванню UX/UI, що дозволило створити інтуїтивний та адаптивний інтерфейс, який значно спрощує процес взаємодії користувачів із системою, про що свідчить опис основних сценаріїв використання. Крім того, були ретельно розроблені механізми забезпечення доступу, авторизації та безпеки даних, з використанням стандартів Google OAuth та розподілу ролей, що гарантує конфіденційність та захист персональної інформації.

Найважливішим аспектом проектування стала розробка алгоритмів розподілу навантаження, які дозволяють автоматично, ефективно та

справедливо призначати навчальні години викладачам. Це дало змогу закласти основу для автоматизації складних оптимізаційних задач, мінімізуючи людський фактор та підвищуючи точність розподілу.

Таким чином, у межах цього розділу було сформовано повноцінну концепцію та технічне обґрунтування автоматизованої системи, що повністю відповідає поставленим вимогам.

РОЗДІЛ 3. РЕАЛІЗАЦІЯ ТА ВАЛІДАЦІЯ АВТОМАТИЗОВАНОЇ СИСТЕМИ

3.1. Структура проєкту, основні модулі й їх реалізація (створення, заповнення, видалення карток)

Розроблена автоматизована система розподілу навчального навантаження побудована з урахуванням принципів модульності та функціональної автономності кожного компонента. Це забезпечує не лише спрощення процесу розробки та тестування, а й полегшує підтримку та масштабування системи в майбутньому. Усі компоненти взаємодіють між собою через спільне середовище зберігання даних — Google Sheets, а програмна логіка реалізована з використанням Google Apps Script. У цьому підрозділі наведено опис кожного з основних функціональних модулів системи.

Модуль вибору таблиці

Даний модуль є стартовим елементом взаємодії користувача із системою. Його основне завдання полягає в наданні користувачу можливості обрати Google-таблицю, яка містить всю необхідну інформацію для формування карток викладачів та подальших операцій з ними. Після натискання відповідної кнопки у вебінтерфейсі відкривається вікно вибору файлу з Google Drive.

Результатом виконання модуля є підключення до обраної таблиці та її подальше використання іншими модулями системи. Назва вибраного документа також відображається у вікні вебінтерфейсу для зручності користувача (рис 3.1).

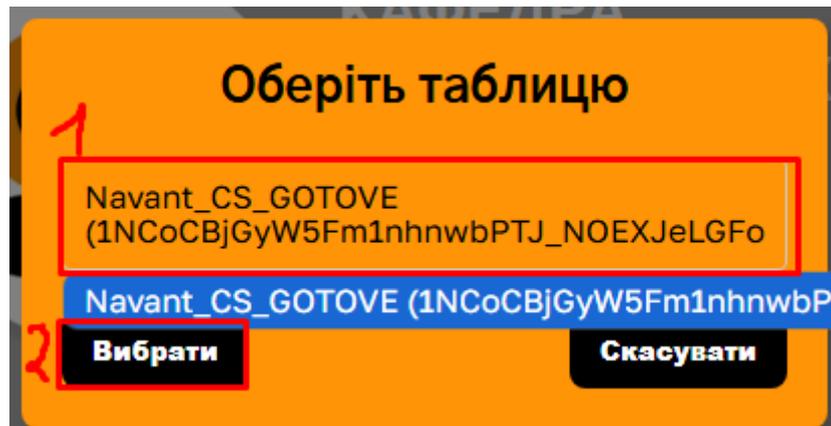


Рисунок 3.1 – Інтерфейс вибору таблиці.

Модуль зчитування списку викладачів

Цей модуль виконує обробку аркуша «SV», який містить узагальнений перелік викладачів кафедри. Після активації функції зчитування система автоматично витягує з таблиці імена, коди та електронні адреси викладачів, формуючи з них інтерактивний список у вебінтерфейсі. До списку додаються чекбокси, що дозволяють відзначити одного або кількох викладачів для подальших дій, а також кнопки «Виділити всіх» та «Зняти виділення».

Ця реалізація забезпечує зручний механізм вибору без потреби ручного введення даних, що значно підвищує швидкість і точність роботи (рис 3.2).

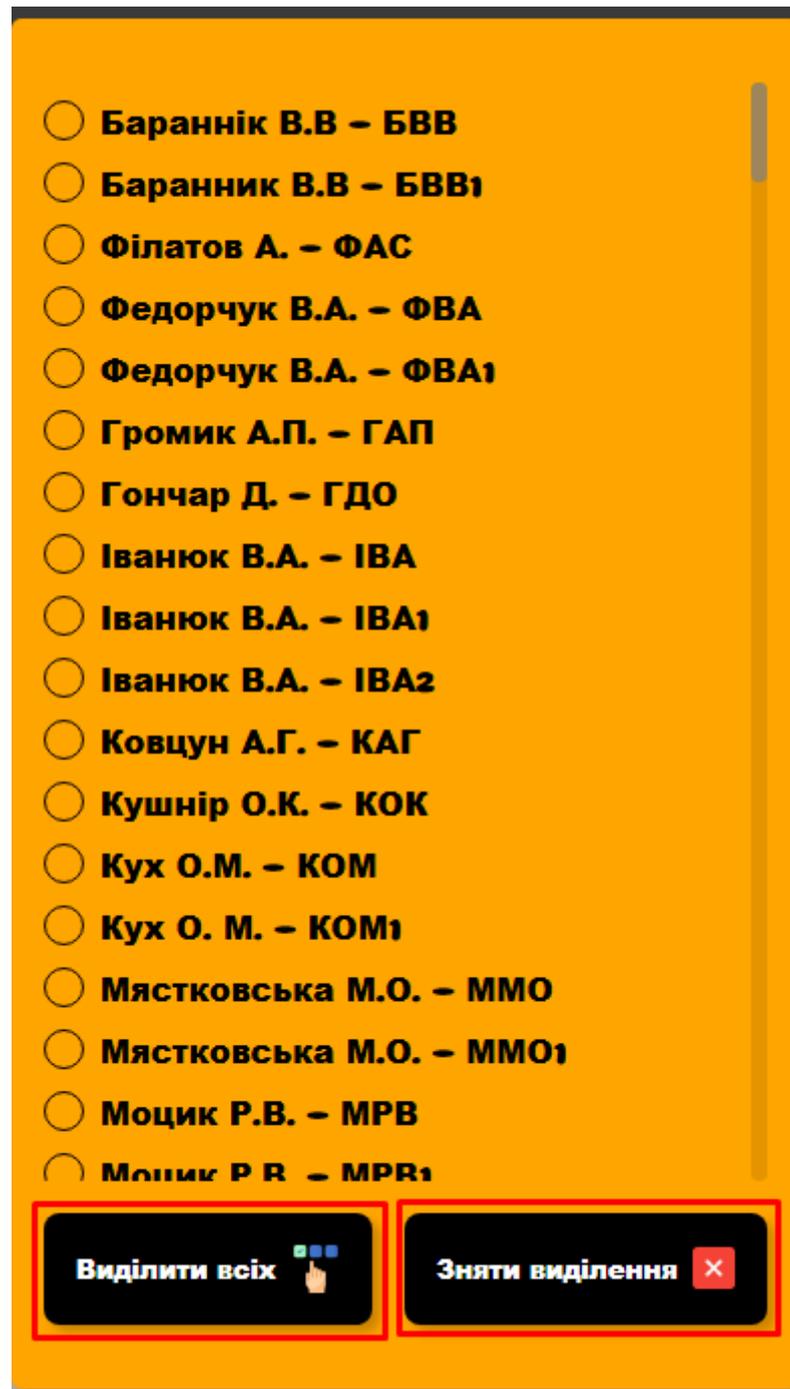


Рисунок 3.2 – Відображення списку викладачів у вебінтерфейсі.

Модуль створення карток викладачів

Після вибору викладачів користувач має можливість автоматично створити картки навантаження для кожного з них. Для цього система створює окремий аркуш у таблиці, ім'я якого формується у форматі «[Код] – [ПІБ]». У разі, якщо аркуш для певного викладача вже існує, його створення не дублюється.

викладачу, формує з нього PDF-файл і зберігає його в окремій папці на Google Drive.

Назва папки автоматично відповідає назві обраної таблиці, що забезпечує зручність у пошуку та організації документів. Цей модуль дозволяє підготувати всі картки до друку або архівації у вигляді офіційної документації (рис 3.6).

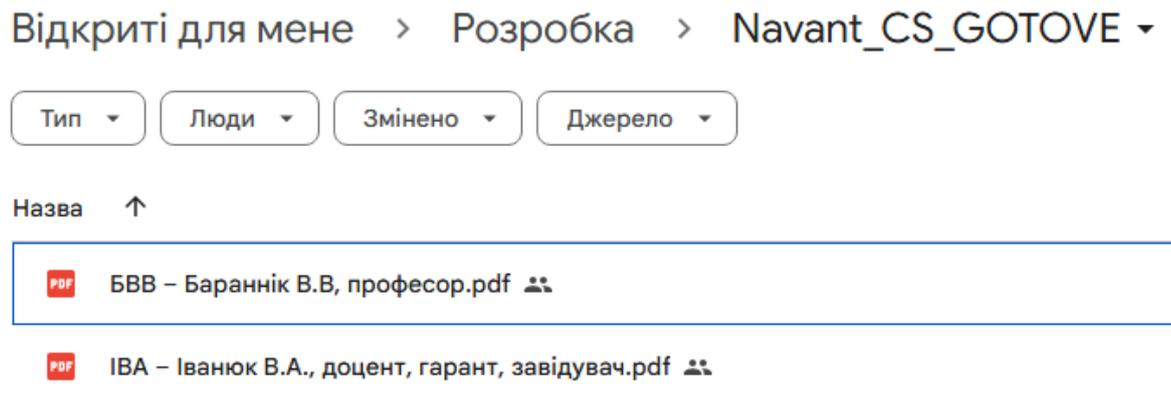


Рисунок 3.6 – Папка з PDF-файлами на Google Drive.

Модуль надсилання електронною поштою

Після формування PDF-документів система дозволяє автоматично надіслати їх на електронні адреси викладачів. Адреси зчитуються з таблиці «SV», а листи формуються індивідуально для кожного отримувача із прикріпленим PDF-документом.

Ця функція значно полегшує комунікацію, особливо у випадках, коли потрібно розіслати документи великій кількості викладачів одночасно (рис 3.7).

Ваша картка викладача Вхідні x



@kpnu.edu.ua

кому мені ▾

Доброго дня,

Ваша картка навантаження.

Один вкладений файл • Перевірено в Gmail ⓘ



← Відповісти

→ Переслати

Рисунок 3.7 – Приклад листа з PDF-карткою у вкладенні.

Модуль створення резервної копії таблиці

З метою збереження історії змін і підвищення надійності було реалізовано модуль створення резервної копії. Він дозволяє зберегти копію активної таблиці з автоматичним додаванням поточної дати й часу до її назви. Така реалізація дозволяє у будь-який момент повернутися до попереднього стану даних (рис 3.8).

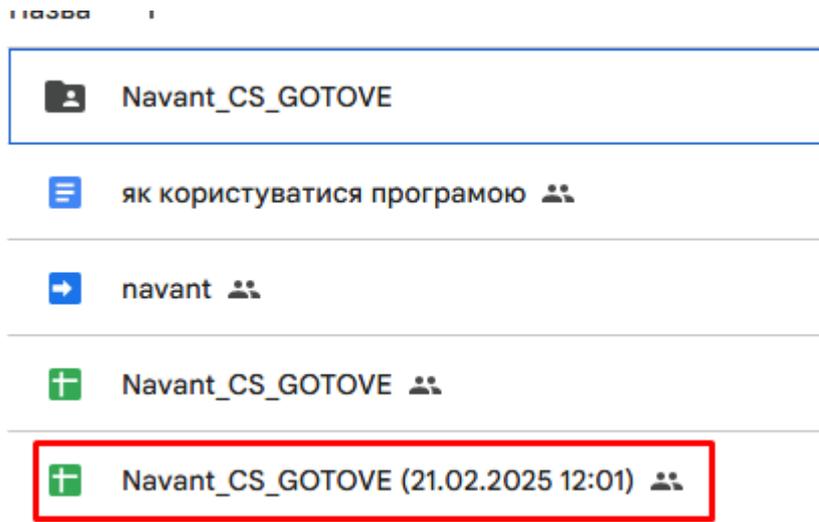


Рисунок 3.8 – Створена копія таблиці з позначенням часу.

Впровадження кожного з описаних модулів дозволило побудувати гнучку, зручну та ефективну систему, що значно спрощує процес розподілу навчального навантаження. Чітке розмежування функцій забезпечує логічну організацію коду, полегшує модифікацію системи та адаптацію під потреби конкретної кафедри або факультету. Інтеграція з Google-сервісами надає широкі можливості для автоматизації рутинних операцій, спільної роботи та захисту даних, що робить систему практичним і сучасним інструментом в управлінні освітнім процесом.

3.2. Генерація PDF-звітів, автоматична відправка електронних листів і друк карток.

Однією з ключових переваг автоматизованої системи є можливість формування та поширення підсумкових документів без участі сторонніх програм або ручного втручання. У цьому пункті розглянуто реалізацію трьох взаємопов'язаних функцій: **генерація PDF-файлів, автоматична розсилка листів викладачам, та формування папки для друку карток [15]**. Усі ці операції реалізовані за допомогою Google Apps Script та сервісів Google Drive / Gmail.

Генерація PDF-карток викладачів

Після створення та заповнення карток викладачів система дозволяє автоматично згенерувати документи у форматі PDF. Цей процес реалізується так:

- Кожен аркуш, який відповідає викладачу (наприклад, «ІВА – Іванюк В.А.»), обробляється скриптом.
- На основі вмісту аркуша формується PDF-документ із відповідною розміткою, збереженням таблиць та візуального оформлення.
- Назва PDF-файлу створюється у форматі: **[Назва аркуша] – [ПІБ викладача].pdf**, наприклад: ІВА – Іванюк В.А.pdf.

Формування документів відбувається у фоновому режимі, без потреби взаємодії з додатковими сервісами (рис 3.9).

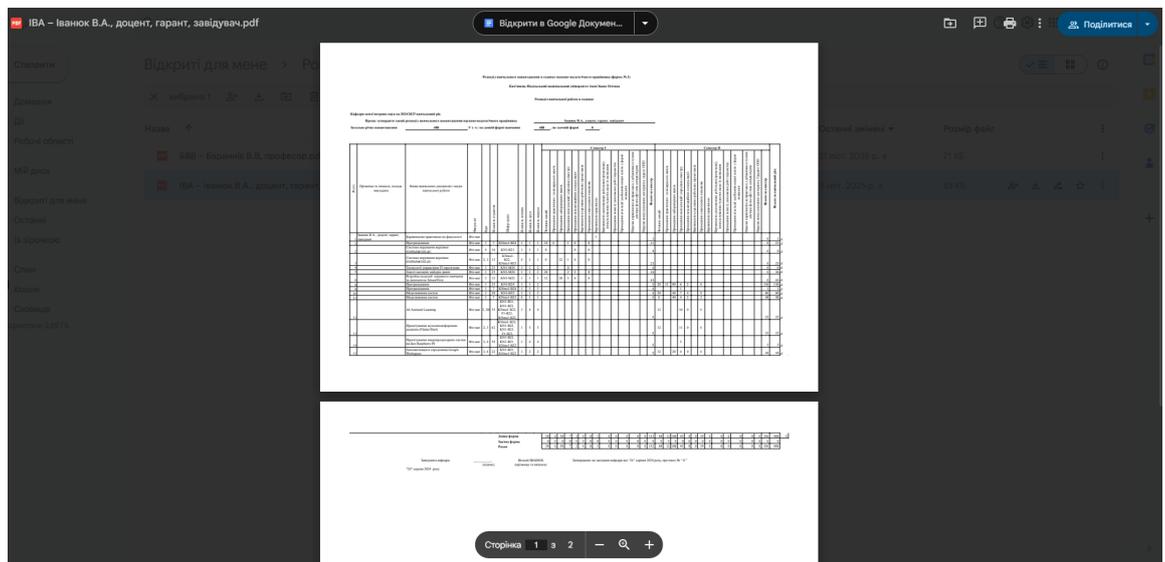


Рисунок 3.9 – Приклад PDF-файлу сформованої картки викладача.

Автоматична розсилка електронних листів

Після формування PDF-файлів користувач може скористатися функцією автоматичного надсилання листів викладачам. Це особливо актуально, коли потрібно оперативно поширити документи між кількома десятками працівників.

Алгоритм дій:

- Електронна адреса викладача зчитується з аркуша SV за унікальним кодом.

- До листа автоматично прикріплюється PDF-картка відповідного викладача.
- Вміст листа може бути уніфікований або змінений у скрипті (наприклад: “Шановний викладачу, надсилаємо Вам оновлену картку навчального навантаження на поточний семестр”).
- Лист надсилається з акаунту, під яким працює користувач у Google.

Реалізація цієї функції дозволяє позбутись необхідності ручної розсилки, яка вимагає багато часу і часто супроводжується помилками (рис 3.10).

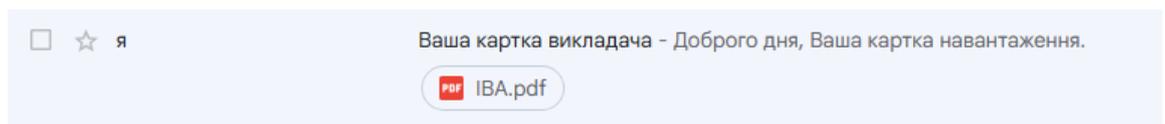


Рисунок 3.10 – Приклад листа, отриманого викладачем з прикріпленою PDF-карткою.

Друк сформованих карток

Ще однією можливістю є **підготовка карток до друку**. Після натискання кнопки «Відправити картки на друк» система виконує наступні дії:

- Генерує всі PDF-документи у фоновому режимі (як описано вище).
- Створює на Google Drive **папку з назвою таблиці**, до якої додає всі згенеровані файли.
- Папка автоматично відкривається в новій вкладці для перегляду або завантаження.
- Користувач має можливість швидко завантажити всі файли для передавання в друкарню або локальний принтер.

Ця функція актуальна у випадках, коли документи потрібно зберегти у паперовому вигляді для звітності, архівації або підпису керівництва (рис 3.11).



Рисунок 3.11 – Папка з PDF-документами у Google Drive, готова до завантаження/друку.

Особливості реалізації

- Усі PDF-файли створюються засобами Google Apps Script без сторонніх бібліотек.
- Обробка відбувається у циклі з індикатором прогресу (в CacheService), що дозволяє відслідковувати стан генерації файлів.
- Механізм обробки помилок дозволяє уникнути збоїв у випадку, якщо для певного викладача відсутній email або аркуш картки.
- Користувач бачить повідомлення про успішне завершення формування PDF або розсилки.

Завдяки реалізації модулів формування PDF, автоматичної розсилки листів та друку сформованих карток, система повністю охоплює фінальний етап розподілу навантаження — комунікацію та документування. Ці можливості дозволяють значно зменшити адміністративне навантаження на співробітників, знизити ризик помилок, забезпечити чіткий контроль документообігу та зручне зберігання архівів.

3.3. Методика функціонального і UX-тестування; результати

З метою перевірки працездатності автоматизованої системи розподілу навчального навантаження та оцінки її зручності для кінцевих користувачів було проведено комплексне тестування. Тестування охоплювало як **функціональні характеристики** розроблених модулів, так і **досвід користувача** (User Experience, UX) — тобто, оцінку зручності, логіки інтерфейсу та швидкості виконання дій. Проведення такого тестування є

критично важливим етапом для впровадження системи у реальних умовах експлуатації.

Функціональне тестування.

Функціональне тестування (functional testing) було зосереджене на перевірці кожного модуля системи згідно з вимогами, описаними у попередніх розділах.

Методика:

- Для кожної функції було сформульовано **тест-кейси** (тестові сценарії), які включали вхідні дані, очікуваний результат та реальний результат.
- Тестування проводилося у реальному середовищі Google Workspace, з використанням тестової таблиці, заповненої навчальними дисциплінами та викладачами.
- Кожен модуль тестувався на різні варіанти вхідних даних — як коректних, так і граничних або некоректних (наприклад, відсутність email-адреси чи дублікати кодів викладачів).

Результати тестування подано в таблиці 3.1

Назва модуля	Кількість тестів	Пройдено успішно	Виявлені помилки
Вибір таблиці	3	3	—
Імпорт списку викладачів	4	4	—
Створення карток	5	5	—
Заповнення карток	6	6	—
Видалення карток / заповнення	4	4	—
Генерація PDF	4	4	—

Розсилка листів	3	3	Виявлено проблему з email без @ (виправлено)
Резервне копіювання таблиці	2	2	—

Таблиця 3.1 – Результати тестування.

Усі функції, окрім однієї незначної помилки у форматі email, пройшли тестування успішно. Виявлена помилка була оперативно усунена шляхом додаткової перевірки email-адреси на валідність перед надсиланням листа.

UX-тестування (оцінка взаємодії користувача).

UX-тестування було спрямоване на перевірку зручності використання інтерфейсу, логічності розміщення елементів, а також часу, необхідного для виконання типових операцій.

Методика:

Було сформовано невелику фокус-групу з 5 осіб: один завідувач кафедр, 1 викладач та три студенти. Учасники отримали коротку інструкцію користувача (додаток Б) і мали самостійно пройти повний цикл роботи з системою: від вибору таблиці до відправки карток на пошту. Після завершення завдання кожному учаснику було запропоновано оцінити такі критерії за 5-бальною шкалою:

- Зрозумілість інтерфейсу
- Логіка взаємодії
- Швидкість виконання завдань
- Задоволеність від використання

Результати UX-оцінки подано в таблиці 3.2.

Критерій	Середня оцінка (з 5)
Зрозумілість інтерфейсу	4.8
Логіка взаємодії	4.6

Швидкість виконання	4.9
Загальна задоволеність	4.8

Таблиця 3.2 – Результати UX - оцінки

Учасники відзначили, що найбільш зручною є функція автоматичної генерації PDF та їх розсилки, яка значно спрощує рутинну роботу. Дрібні зауваження стосувалися переважно дизайну (наприклад, бажання бачити підказки при наведенні на кнопки), що може бути реалізовано у подальших ітераціях.

Тестування показало, що розроблена система є стабільною, функціонально повною та зручною у використанні. Усі ключові модулі працюють відповідно до поставлених вимог, забезпечуючи ефективне вирішення завдань з автоматизації розподілу навчального навантаження. Проведене UX-тестування підтвердило, що інтерфейс є інтуїтивно зрозумілим, а система в цілому — доступною навіть для користувачів без технічної підготовки.

Сукупно це дозволяє стверджувати, що система готова до впровадження у реальному середовищі на рівні кафедри або факультету.

3.4. Підготовка до впровадження системи; інструкція користувача.

На момент завершення розробки система автоматизованого розподілу навчального навантаження перебуває на етапі підготовки до впровадження в реальну роботу кафедри. Було завершено всі ключові технічні компоненти, проведено тестування в умовах симуляції реального навантаження, створено інтерфейс користувача та підготовлено супровідну документацію. Наступним етапом є впровадження системи у повсякденну роботу, яке заплановано після початку нового навчального періоду [14].

Організаційні передумови

На етапі розробки була сформована **базова структура робочої таблиці**, до якої внесено дані викладачів, дисциплін, семестрів, форм навчання. Таблиця структурована відповідно до моделі, описаної в розділі 2.2, та

відповідає внутрішнім форматам обліку навчального навантаження, які застосовуються на кафедрі.

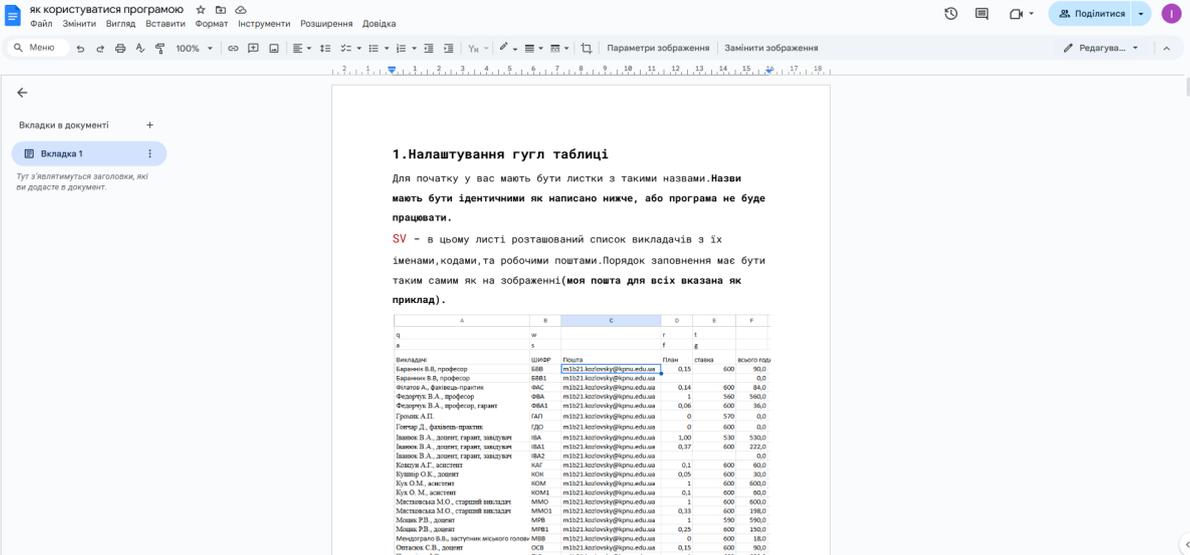
Було протестовано роботу всіх основних модулів: створення карток, їх заповнення, генерація PDF-документів, електронна розсилка та резервне копіювання. Попередні результати засвідчили повну працездатність системи в ізольованому середовищі на прикладах тестових даних.

Інструкція користувача

Для спрощення освоєння системи розроблено повноцінну інструкцію користувача, у якій викладено:

- порядок підключення таблиці до інтерфейсу;
- процес додавання та вибору викладачів;
- покроковий алгоритм створення та заповнення карток;
- пояснення функцій друку, розсилки та резервного копіювання.

Інструкція проілюстрована знімками інтерфейсу, що полегшує сприйняття інформації навіть користувачами без досвіду роботи з Google Apps Script або Google Sheets (рис 3.12).



1. Налаштування гугл таблиці

Для початку у вас мають бути листки з такими назвами. Назви мають бути ідентичними як написано нижче, або програма не буде працювати.

SV - в цьому листі розташований список викладачів з їх іменами, кодами, та робочими поштами. Порядок заповнення має бути таким самим як на зображенні (моя пошта для всіх вказана як приклад).

	A	B	C	D	E	F
q		sv		f	t	
r		s	f	#		
Викладач	ШКОР	Години	План	Станов	Всього годин	
Баранчик В.В., професор	688	m1312.koc@yandex.ru	0,15	600	90,0	
Баранчик В.В., професор	6881	m1312.koc@yandex.ru			0,0	
Виноградов А.А., викладач-практик	6882	m1312.koc@yandex.ru	0,14	600	84,0	
Виноградов В.А., професор	6883	m1312.koc@yandex.ru		1	360	
Виноградов В.А., професор, гігант	6884	m1312.koc@yandex.ru	0,06	600	36,0	
Григорук А.П.	6885	m1312.koc@yandex.ru	0	370	0,0	
Григорук Д., викладач-практик	6886	m1312.koc@yandex.ru	0	600	0,0	
Заванюк В.А., доцент, гігант, замучити	6887	m1312.koc@yandex.ru	1,00	530	530,0	
Заванюк В.А., доцент, гігант, замучити	6888	m1312.koc@yandex.ru	0,17	600	222,0	
Заванюк В.А., доцент, гігант, замучити	6889	m1312.koc@yandex.ru			0,0	
Козлов А.Г., доцент	6890	m1312.koc@yandex.ru	0,1	600	60,0	
Козлов О.С., доцент	6891	m1312.koc@yandex.ru	0,05	600	30,0	
Кук О.М., асистент	6892	m1312.koc@yandex.ru	1	600	600,0	
Кук О.М., асистент	6893	m1312.koc@yandex.ru			0,0	
Маслаков М.О., старший викладач	6894	m1312.koc@yandex.ru	1	600	600,0	
Маслаков М.О., старший викладач	6895	m1312.koc@yandex.ru	0,13	600	198,0	
Молова Р.В., доцент	6896	m1312.koc@yandex.ru		1	590	
Молова Р.В., доцент	6897	m1312.koc@yandex.ru	0,25	600	150,0	
Менделєєв В.В., науковий викладач	6898	m1312.koc@yandex.ru	0	600	18,0	
Попов С.В., доцент	6899	m1312.koc@yandex.ru	0,15	600	90,0	
Попов С.В., доцент	6900	m1312.koc@yandex.ru	1	600	600,0	

Рисунок 3.12 – Вигляд інструкції для користувача.

Очікувані результати впровадження

Після впровадження системи очікується досягнення таких результатів в таблиці 3.3.

Критерій	Ручний підхід (Excel)	Автоматизована система
Час створення картки	1-4 хв	5-20 сек
Ризик помилки	Високий	Мінімальний
Масштабованість	Обмежена	Висока
Потрібні знання	Excel	Базові

Таблиця 3.3 – Переваги автоматизованої системи

Проведені тестування підтвердили її працездатність, а позитивні відгуки тестувальників засвідчили готовність до впровадження в робочі процеси кафедри. Система має високий потенціал адаптації, здатна значно полегшити адміністративну роботу та може бути поширена на інші підрозділи закладу вищої освіти.

Висновки до розділу 3

У третьому розділі було детально описано реалізацію автоматизованої системи розподілу навчального навантаження між викладачами кафедри на базі хмарних технологій Google. Реалізація проводилась поетапно, з чітким розмежуванням модулів, що забезпечує гнучкість системи, її масштабованість і простоту технічної підтримки.

Розроблені модулі охоплюють усі ключові етапи процесу — від підключення таблиці та створення карток викладачів до автоматичної генерації PDF-документів, розсилки електронних листів і створення резервних копій даних. Особлива увага приділялась забезпеченню зручності для

кінцевого користувача, що відобразилось у структурі вебінтерфейсу та логіці взаємодії з системою.

Проведене функціональне тестування підтвердило стабільну роботу всіх основних компонентів, а UX-оцінювання продемонструвало високий рівень задоволеності з боку потенційних користувачів. Незначні зауваження, виявлені під час тестування, були усунуті ще до завершення розробки.

Узагальнюючи результати, можна зробити висновок, що розроблена система повністю відповідає поставленим цілям та технічним вимогам, а її використання дозволить суттєво оптимізувати роботу з формування навчального навантаження в межах кафедри та потенційно — у масштабах усього навчального закладу.

ВИСНОВКИ

У межах кваліфікаційної роботи було розроблено автоматизовану систему для розподілу навчального навантаження між викладачами кафедри. Система забезпечує зручне створення, заповнення та облік індивідуальних викладацьких карток з використанням хмарних технологій Google Sheets і Google Apps Script.

Запропоноване рішення поєднує інтуїтивний вебінтерфейс, автоматизовану логіку опрацювання даних, а також підтримку інтеграції з наявною інфраструктурою Google Workspace. В основу функціонування покладено модульну обробку даних з шаблонних аркушів, з урахуванням семестрової структури, типів занять та індивідуальних шифрів викладачів. Реалізовано повний цикл роботи з навантаженням: від імпорту даних і формування PDF-звітів до автоматичної розсилки та резервного копіювання.

Система підтримує багатокористувацький доступ, забезпечує прозорість дій, мінімізує помилки ручного введення та суттєво скорочує час обробки даних. Розроблено користувацьку інструкцію, а також проведено функціональне тестування, що підтвердило стабільність і коректність роботи основних модулів. Станом на момент завершення проєкту система повністю готова до впровадження в роботу кафедри, з перспективою масштабування на інші структурні підрозділи.

Загалом, реалізоване рішення сприяє цифровізації управлінських процесів в освіті, зменшує адміністративне навантаження та підвищує якість внутрішнього документообігу. Архітектурна відкритість і гнучкість системи створюють передумови для її подальшого розвитку — зокрема розширення функціоналу, аналітики навантаження, рольового доступу, інтеграції з календарями, розкладом та академічними базами.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Закон України «Про вищу освіту». – Відомості Верховної Ради України, 2014, №37–38, ст.2004.
2. Беленький В.Ю., Шевченко Н.О. Інформаційні системи в управлінні вищим навчальним закладом. – Київ: КНЕУ, 2018. – 152 с.
3. Андреев А.І., Гринько В.А. Основи автоматизованого документообігу у ВНЗ. – Харків: ХНУРЕ, 2020. – 132 с.
4. Пономаренко В.С. та ін. Автоматизація управлінських процесів у закладах освіти. – Харків: Вид-во «Магістр», 2021. – 228 с.
5. Nielsen J. Usability Engineering. – San Francisco: Morgan Kaufmann, 1994. – 362 p.
6. Krug S. Don't Make Me Think. – New Riders, 2014. – 216 p.
7. Google Apps Script Documentation – <https://developers.google.com/apps-script>
8. Google Sheets API Reference – <https://developers.google.com/sheets/api>
9. Kurnosov V., Plynyskyi R. Cloud technologies in the education system: prospects and implementation models // Information Technologies and Learning Tools. – 2022. – №2 (88). – С. 122–133.
10. W3C Web Accessibility Initiative. Web Content Accessibility Guidelines (WCAG) – <https://www.w3.org/WAI/standards-guidelines/wcag/>
11. Петренко І.В. Розробка інформаційних систем на базі хмарних сервісів Google // Інформаційні технології і засоби навчання. – 2021. – №6 (86). – С. 94–102.
12. Шапран О.І. Автоматизовані системи управління навчальним процесом: огляд та порівняння // Вісник КНУ імені Тараса Шевченка. – 2020. – № 2 (43). – С. 55–61.
13. Holubiev A., Melnyk A. Cloud Computing in Education: Challenges and Opportunities. // International Journal of Information Technologies and Systems Approach. – 2022. – Vol. 15. – P. 40–52.

14. Український центр оцінювання якості освіти. Методичні рекомендації щодо планування навчального навантаження у ЗВО. – Київ: УЦОЯО, 2021.
15. Сидоренко С.П. Розподіл навчального навантаження: ручні методи vs автоматизація // Освітній простір України. – 2022. – №3(99). – С. 45–49.