

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДОНЕЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ВАСИЛЯ СТУСА

САЛАЦІНСЬКИЙ ВОЛОДИМИР ЛЕОНІДОВИЧ

Допускається до захисту:

в.о. завідувача кафедри
інформаційних технологій
канд. техн. наук, доцент

_____ О. В. Зелінська

« _____ » _____ 20__ р.

**ІНТЕРАКТИВНИЙ ДОДАТОК ДЛЯ ОПТИМІЗАЦІЇ РОБОТИ
ПЕДАГОГІЧНИХ ПРАЦІВНИКІВ**

Спеціальність 122 Комп'ютерні науки

Кваліфікаційна (бакалаврська) робота

Керівник:

І. В. Фриз, старший викладач
кафедри інформаційних технологій,
к. ф.-м. н.

Оцінка: _____ / _____ / _____

(бали/за шкалою ЄКТС/за національною шкалою)

Голова ЕК: _____

Вінниця – 2025

АНОТАЦІЯ

Салацінський В. Л. Інтерактивний додаток для оптимізації роботи педагогічних працівників. Спеціальність 122 «Комп'ютерні науки», освітня програма «Комп'ютерні науки». Донецький національний університет імені Василя Стуса, Вінниця 2025.

У кваліфікаційній (бакалаврській) роботі досліджено та проаналізовано процеси надання освітніх послуг працівниками педагогічних закладів. За допомогою таких технологій як Python, бібліотек Tkinter, Openpyxl розроблений інтерактивний додаток «Асистент», основним завданням якого є оптимізація роботи педагогічних працівників.

Ключові слова: педагогічні працівники, оптимізація, заклади освіти, Excel, Python, Tkinter, Openpyxl.

59 ст. 26 рис., 1 дод., 40 джерел.

ABSTRACT

Salatsinsky V. L. Interactive application for optimizing the work of the pedagogical staff. Specialty 122 «Computer Science», educational program «Computer Science». Vasyl Stus Donetsk National University, Vinnytsia 2025.

In the qualification (bachelor's) work, the processes of providing educational services by employees of pedagogical institutions are researched and analyzed. With the help of such technologies as Python, Tkinter, Openpyxl libraries, an interactive application «Assistant» was developed, the main task of which is to optimize the work of educators.

Keywords: pedagogical staff, optimization, educational institutions, Excel, Python, Tkinter, Openpyxl.

ЗМІСТ

ВСТУП	4
РОЗДІЛ 1. ДОСЛІДЖЕННЯ ПРОБЛЕМ ПЕДАГОГІЧНИХ ПРАЦІВНИКІВ ТА ФОРМУЛЮВАННЯ МОЖЛИВИХ ВАРІАНТІВ РІШЕННЯ ПРОБЛЕМИ	7
1.1 Окреслення глобальних проблем працівників закладів освіти	7
1.2 Дослідження робочого процесу викладачів та формування основного завдання для оптимізації	10
1.3 Вибір середовища розробки додатку	13
РОЗДІЛ 2. ПРОЕКТУВАННЯ ІНТЕРАКТИВНОГО ДОДАТКУ ДЛЯ ОПТИМІЗАЦІЇ РОБОТИ ПЕДАГОГІЧНИХ ПРАЦІВНИКІВ	16
2.1 Переваги та недоліки мови програмування Python та головне напрямлення при розробці коду	16
2.2 Створення прототипів інтерфейсу для інтерактивного додатку та структури журналу	19
2.3 Формування етапів роботи програми та їх функціоналу	21
РОЗДІЛ 3. РЕАЛІЗАЦІЯ ІНТЕРАКТИВНОГО ДОДАТКУ ТА ПЕРЕВІРКА ЙОГО ПРАЦЕЗДАТНОСТІ	24
3.1 Підготовка даних для подальшої активації книги	24
3.2 Створення класу, функції візуалізації інтерфейсу	31
3.3 Функції що відповідають за роботу з часом та вторинні функції	34
3.4 Функції, що відповідають за запуск та виведення інформації	42
3.6 Функції реакції та тестування роботи програми	46
ВИСНОВКИ	53
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	55
ДОДАТКИ	60

ВСТУП

Основою будь якої компанії, підприємства, закладу вищої чи середньої освіти є працівники (люди). Від їх зацікавленості, мотивації, прагнень до розвитку залежить розвиток організації та ефективності виконання нею поставлених задач. З кожним роком можна спостерігати зростаючу тенденцію до створення комфортних умов праці, відділи для підтримки мотивації персоналу чи регулярне проведення різних заходів для підтримання позитивного настрою серед працівників. Внаслідок такої політики сучасні компанії створюють міцний, дружній та ефективний робочий колектив, що дозволяє компанії швидко розвиватись. Внаслідок таких тенденцій підприємства почали все менше звертати уваги на рівень QOL.

Сьогодні термін QOL набирає все більше популярності, а його значення отримало значне розширення. Тепер під терміном QOL можуть мати на увазі рівень оптимізації вторинних процесів, що опосередковано впливають на головне робоче завдання працівників [2]. На перший погляд рівень QOL може здаватись не таким важливим. Однак на практиці оптимізація вторинних робочих процесів допомагає зменшити стресове навантаження працівників та покращити їх рівень зосередженості на роботі, що створює неймовірний вплив на ефективність роботи та дозволяє значно збільшити їх працездатність. Таким чином, з кожним роком буде збільшуватися необхідність у дослідженні та покращенні вторинних робочих процесів, з метою покращення індивідуальної ефективності і працездатності працівників не обмежуючись лише впливом колективу чи робочої атмосфери. Також підвищення рівня QOL може впливати на ефективність роботи ізольованих від колективу працівників по причинах дистанційного формату роботи, хворобі або різний профілактичних чи технічних заходах компанії.

Одним з засобів впровадження QOL у різні компанії є розробка програм помічників та їх інтеграція у повсякденну роботу. Представником компаній які потребують розробки подібних додатків є різні навчальні та вищі навчальні

заклади. Асистент» зможе стати такою допоміжною програмою, що зможе посприяти покращенню процесу викладання за різних обставин: як у період військового стану так і у мирний час.

Отже, метою дослідження є розробка інтерактивного додатку для оптимізації роботи педагогічних працівників навчальних закладів.

Завданнями дослідження є:

- визначення вторинних робочих процесів, які виконуються педагогічними працівниками та вивчити способи їх оптимізації, покращення, чи заміни за необхідністю;
- дослідити та проаналізувати інструменти розробки інтерактивних додатків;
- розробити та протестувати програму для оптимізації вторинних робочих процесів.

Об'єктом дослідження є процес проведення навчальних заходів (занять), а також визначення загальних робочих процесів у закладах вищої та середньої освіти. Метою охоплення такого великого об'єкту дослідження є повноцінна оцінка та аналіз робочих і адміністративних завдань педагогічних працівників.

Предметом дослідження є програми ведення електронної документації для подальшої розробки інтерактивного додатку та його інтерфейсу.

Розроблене програмне забезпечення має великі можливості для інтеграції у закладах освіти через його індивідуальну направленість на педагогічних працівників та відсутність необхідності у прямій взаємодії розробленого додатку з іншими навчальними системами. Також розроблений додаток є локально напрумленим, що говорить про відсутність необхідності підключення до глобальної мережі Інтернету для його роботи, що буде корисним у зонах з нестабільною роботою мережі, електропостачання, чи необхідності раптової зміни місця проведення занять. Разом з тим є можливість збору та проведення простого аналізу даних для створення статистик чи аналізу тенденцій у певний період часу. До того ж враховуючи, що програма є персональною це робить дані розміщені у ній конференційними.

Бакалаврська робота складається з трьох розділів, висновків, списку використаних джерел, додатків. У першому розділі досліджуються проблеми педагогічних працівників, які виникають під час навчального процесу, та способи їх вирішення. У другому розділі розглядається проектування інтерактивного додатку, функцією якого є зменшення навантаження педагогічних працівників та підвищення їхньої працездатності. У третьому розділі описана реалізація та тестування запропонованого додатку.



РОЗДІЛ 1

ДОСЛІДЖЕННЯ ПРОБЛЕМ ПЕДАГОГІЧНИХ ПРАЦІВНИКІВ ТА ФОРМУЛЮВАННЯ МОЖЛИВИХ ВАРІАНТІВ РІШЕННЯ ПРОБЛЕМИ

1.1 Окреслення глобальних проблем працівників закладів освіти

Проаналізуємо основні проблеми, з якими зіштовхуються педагогічні працівники у своїй повсякденній діяльності та детальніше розглянемо їх.

Першою із таких розповсюджених проблем у освітніх закладах є слабка матеріально-технічна база. Внаслідок цього більшість кабінетів, залів чи аудиторій не мають відповідного технічного обладнання (проекторів, комп'ютерів чи іншого аудіо-візуального обладнання). Тому розповсюдилась практика використання власних персональних пристроїв серед працівників освітніх закладів, що може викликати проблеми іншого характеру такі як вірусні загрози, відсутність уніфікованої системи підключення чи неможливість взаємодії різних технічних пристроїв (внаслідок відсутності відповідних драйверів чи несумісності операційної системи).

Другою розповсюдженою проблемою є відсутність постійного робочого кабінету у педагогічних працівників. У багатьох навчальних закладах, за рахунок більшої кількості педагогічних працівників у порівнянні з кількістю робочих кабінетів, надається перевага прив'язки кабінету до класу. Це створює необхідність у постійному переході між кабінетами викладачів підчас перерв, що зменшує час на підготовку викладача до заняття. А також, враховуючи раніше описану проблему, можна говорити про необхідність постійного переносу власного обладнання чи матеріалів для проведення уроку.

Третьою проблемою є технічні несправності з локальною мережею чи перебої в енергопостачанні, що може спричиняти різні завади навчальному процесу (демонстрації різних навчальних аудіо-візуальних матеріалів, роботи з різними онлайн платформами для тестування чи передачі та розповсюдження навчального матеріалу, використання системи онлайн звітності). Це може

викликати необхідність у надмірній підготовці до занять, що збільшує навантаження на педагогічних працівників.

У своїй статті Ганна Бондаренко визначає декілька ключових викликів сучасної освіти в Україні. Першим згадано відключення електроенергії, що призводить до відсутності інтернету та неспроможності проведення онлайн занять. Інший – недостатнє фінансування сфери освіти у військовий час [1]. Щодо проблем із матеріально-технічною базою, то можна навести такі факти. По-перше, військові дії в Україні призвели до досить важких наслідків для інфраструктури закладів освіти: за рік зруйновано та пошкоджено 3 170 об'єктів, загальна сума збитків сягнула майже 9 млрд доларів США. По-друге держава більше не має можливості фінансувати сферу освіти і науки на довоєнному рівні, тому видатки загального фонду державного бюджету, заплановані у 2022 р. на її фінансування, було скорочено на більш ніж 14 млрд грн [36].

На сьогоднішній день, знаючи про існування цих проблем, їх вирішення залишається непростим завданням. Дослідивши їх можна навести кілька варіантів вирішення для деяких з них.

Одним із таких способів є використання інновацій у навчальному процесі для його оптимізації.

Термін «інновація» вперше було застосовано в економічній сфері, а саме у роботі австрійського вченого Йозефа Шумпетера «Теорія економічного розвитку». Завдяки розвитку економіки з'явилися суміжні терміни «новація», «нововведення», які нерідко ототожнюють, хоч між ними і є певна різниця. Слово «innovation» (англ.) походить від латинського *inovatis* (*in* – в, *novus* – новий) – запроваджене нововведення, що забезпечує якісне зростання ефективних процесів чи продукції, затребуване ринком [18]. Існує цілий ряд тлумачень терміну «інновація»: сукупність виробничих, технічних і комерційних заходів, які ведуть до появи на ринку нових та вдосконалених промислових процесів і обладнання [30]; впровадження чогось нового відносно до організації або її безпосереднього оточення, будучи особливим випадком процесу зміни в організації [34]; новостворені (застосовані) і (або) вдосконалені

конкурентоспроможні технології, продукція або послуги, а також організаційно-технічні рішення виробничого, адміністративного, комерційного або іншого характеру, що істотно поліпшують структуру та якість виробництва і (або) соціальної сфери [21].

Вирішення проблеми із стабільним підключення у результаті різних обставин можна за рахунок розробки різних мобільних персональних додатків, що матимуть змогу працювати як за підключення до всесвітньої мережі Інтернет, так і без доступу до неї. Це допоможе викладачам продовжувати свою роботу не відволікаючись на технічні несправності. Сучасні технології надають можливість для покращення процесу навчання та розвитку здобувачів освіти різних рівнів та областей знань [13]. Зростаюча роль освіти в суспільному розвитку обумовлює потребу в кваліфікованих спеціалістах та збільшенням вимог до їх професійної підготовки, що вимагає безперервного навчання та удосконалення навичок впродовж всього життя [4]. Однак для більшості педагогічних працівників, особливо старшого покоління або з проблемами зору, буде складно адаптуватись до нових інтерактивних додатків та використовувати їх на постійній основі.

Що стосується проблеми відсутності постійного робочого кабінету у педагогічних працівників, то змінивши систему прив'язки кабінетів з учнів на викладачів можна було б розв'язати цю проблему, однак велика кількість навчальних закладів не зможе виконати зазначену зміну через недостатню кількість обладнання чи робочих кабінетів, на що вказує аналіз фінансового забезпечення закладів вищої та загальної середньої освіти в Україні в умовах збройного конфлікту проведеного у роботах [26] і [27].

Однак в місцях, що змогли змінити систему, спостерігається вагоме поліпшення робочої атмосфери, ефективності освітнього процесу та збільшення працездатності викладачів.

Тому існує необхідність у створенні нових рішень для зазначених вище проблем як варіант розробка інтерактивної програми, яка відповідатиме таким умовам:

- надає можливість вільно запускатись на більшості персональних пристроях;
- не потребує встановлення;
- займає відносно не великий розмір;
- надає можливість перенесення на різні носії;
- не потребує значних ресурсів системи;
- бути локальною програмою.

1.2 Дослідження робочого процесу викладачів та формування основного завдання для оптимізації

Як зазначають Литвинова С., Мельник О. та Сухіх А. [25], на теперішній час використання електронних журналів чи подібних програм стало не від'ємною частиною роботи педагогічних працівників. Результати дослідження з розробки й упровадження електронного журналу обліку успішності курсантів Національної академії Державної прикордонної служби України ім. Б. Хмельницького як дієвого інструмента раціоналізації навчально-виховного процесу, дозволяє ефективніше використовувати робочий час науково-педагогічного колективу, підвищити об'єктивність оцінювання, покращити своєчасність виявлення навчальних проблем, прогнозувати та коригувати навчальну діяльність студентів, підвищувати їх мотивацію під час роботи на заняттях [20]. Подібні програми використовуються для таких дій:

- виставлення відміток про присутність на початку уроку або перед його закінчення;
- знаходження імен учнів (студентів) протягом уроку для коректного звернення чи запису оцінки;
- регулярна перевірка розкладу для встановлення відповідного часу та формування плану подальших дій;
- оцінка активності учнів для надання менш активним чи часто відсутнім можливості демонстрації чи перевірки їх знання;

- підведення підсумків для формування підсумкових балів (оцінок).

Систематичний моніторинг навчальних досягнень дозволяє викладачам та керівництву закладів освіти своєчасно отримувати дані про успішність учнів (студентів), що допомагає прийняти ґрунтовні рішення та розробити стратегії для покращення освітнього процесу [14]. Своєчасний і повноцінний контроль та об'єктивне оцінювання забезпечують учням якість засвоєння знань, дають можливість зрозуміти помилки, неточності, вчасно їх виправити і краще зрозуміти наступний навчальний матеріал, а також сформувати здібність до самоконтролю. Щодо педагога такий контроль дає інформацію про перебіг та якість засвоєння матеріалу, типові помилки, інтереси та здібності учнів, що дозволяє бачити свої дидактичні промахи і своєчасно вносити корективи в навчально-виховний процес [28]. Одним з найбільш складних завдань для педагога є справедливий контроль та оцінювання знань учнів ефективність якого значною мірою залежить від реалізації певних принципів, незалежно від того, у якому типі навчального закладу і в межах якої освітньої програми він працює [35].

Проблему оцінювання навчальних досягнень здобувачів освіти досліджували А. Гіроль [22], А. Гривко [17], В. Динько [15], В. Кирикович [22], О. Ковтун [23], В. Козяр [22], О. Осередчук [33], С. Сидоренко [23] та ін.

На даний час спостерігається тенденція того, що робота з журналами є невід'ємною частиною робочого процесу педагогічних працівників. Їм необхідний постійний доступ до журналу для ефективного проведення заняття. Однак це не завжди є можливим.

Розглянемо сучасний електронний журнал, що на сьогоднішній день використовується у багатьох закладах середньої освіти, а саме «Нові знання» [39].

На сам перед необхідно зазначити, що це не тільки журнал для виставлення оцінок, це скоріше щоденник для налагодження взаємодії між батьками, учнем та викладачем. Викладачам необхідно постійно додавати нові матеріали: домашні завдання на наступне заняття; теми уроків, що були проведені чи

будуть; попереджати про контрольні чи самостійні роботи; додавати посилання на необхідні матеріали чи тести.

Регулярна необхідність роботи з сайтом викладачів та учнів (отримання оцінок, перевірка, домашнього завдання тощо) створює постійне навантаження на сайт, що закономірно приводить до появи помилок чи тимчасового припинення роботи. А якщо для розгляду взяти час підведення підсумків, наприклад кінець навчального року чи семестру, можна спостерігати стабільні неполадки у роботі сайту протягом доби. Також не можна забувати про технічні проблеми, що можуть виникати під час військового стану, а саме вимкнення живлення чи глобальної мережі або необхідність перенесення місця проведення заняття до місць з поганим чи повністю відсутнім зв'язком з глобальною мережею Інтернет.

Не менш важливим етичним питанням є захист конфіденційності результатів моніторингу навчальних досягнень здобувачів освіти. Результати оцінювання містять конфіденційну особисту інформацію про академічні досягнення здобувачів, зокрема, осіб з особливими освітніми потребами, які потребують ретельного захисту від несанкціонованого доступу або розголошення [14]. Здобувачі освіти повинні бути поінформовані про процедури захисту конфіденційності результатів оцінювання та мати право на доступ і перегляд своїх персональних даних. Нецільове використання персональних даних здатне завдати здобувачам великої шкоди [11].

Тому більшість викладачів віддають перевагу створенню додаткового журналу (зазвичай паперового) та періодичним перенесенням відповідних даних до головного журналу, що є причиною затримок робочого процесу педагогічних працівників. Враховуючи вище сказане головною функцією нашого інтерактивного додатку є можливість роботи із журналом за будь яких обставин та зручність в його використанні.

1.3 Вибір середовища розробки додатку

Python – це досить проста та популярна мова програмування яка вивчається у більшості навчальних закладах та активно використовується розробниками, тому програму розроблено саме на її основі.

Програми написані на Python можуть спокійно запускатись майже на будь-яких системах. Після встановлення стандартного дистрибутиву мови програмування Python надає просте, але разом із тим досить потужне середовище розробки, яке називається IDLE, що дозволяє не встановлювати стороні IDE чи редактори коду. Однак для більшої ефективності розробки програми рекомендується використання IDE через їх розширений функціонал у порівнянні з стандартним IDLE. Розглянемо декілька з них:

а) PyCharm Community Edition.

PyCharm кросплатформне, потужне інтегроване середовище розробки, створене міжнародною компанією JetBrains, що спеціалізується на розробці інструментів для програмування різними мовами, як-от Java, Kotlin, C#, F#, C++, Ruby, Python, PHP, JavaScript і багатьох інших. Вперше була опублікована бета-версія у 2010, через три місяці була випущена версія 1.0. Подальшому отримала ще два глобальних оновлення отримавши версію 3.0. Через невеликий проміжок часу була випущена PyCharm Community Edition, безкоштовна версія з відкритим початковим кодом була опублікована 2013.

Ключовими особливостями є:

- автоматичне форматування коду та виправлення помилок;
- аналізатор коду (авто-завершення, підказки, підсвічування синтаксису, рекомендації щодо виправлення помилок...);
- вбудовані термінали різних напрямленостей;
- широкий список налаштувань інтерфейсу;
- можливості налагодження (підтримує точки зупинки, покрокове налагодження та перевірку змінних);
- зберігати імпортовані бібліотеки у папці проекту.

б) Visual Studio Code.

Visual Studio Code (VS Code) – це безкоштовний, відкритий і крос-платформний редактор коду, розроблений компанією Microsoft. Вперше він був випущений у квітні 2015 року і з тих пір набув широкої популярності серед розробників.

Ключовими особливостями є:

- відсутня спеціалізація;
- легкість та висока продуктивність;
- інтелектуальні пропозиції коду (інтелектуальне завершення коду, підсвічування синтаксису та виявлення помилок);
- широкий ринок розширень (можуть встановлювати теми, мовну підтримку, інструменти налагодження та додаткові функції для покращення робочого процесу);
- вбудований термінал;
- можливості налагодження (підтримує точки зупинки, покрокове налагодження та перевірку змінних);

в) Thonny.

Thonny – це просте IDE для Python, призначене для початківців. Воно підтримує різні способи покрокового виконання коду, покрокове обчислення виразів, детальну візуалізацію стеку викликів і режим пояснень концепцій посилань.

Ключовими особливостями є:

- мінімальний вплив на систему;
- простий інтерфейс;
- автодоповнення та налагоджування;
- спеціальні функції для навчання (пояснення концепцій посилань, візуалізація стеку викликів, довідка та підказки для новачків).

Дослідивши проблеми з якими постійно зустрічаються педагогічний працівник та проаналізувавши можливі способи їх вирішення, сформовані основні критерії для інтерактивного додатку «Асистент»

На основі аналізу робочого процес педагогічних працівників встановлено, що є постійна необхідність у доступі до робочого журналу або його аналогів, тому вирішено зробити основною функцією інтерактивного додатку роботу з журналом.

Розглянувши декілька популярних IDE з різних сторін, та порівнявши їхні ключові особливості, обрано для створення інтерактивного додатку «Асистент», IDE PyCharm.



РОЗДІЛ 2

ПРОЕКТУВАННЯ ІНТЕРАКТИВНОГО ДОДАТКУ ДЛЯ ОПТИМІЗАЦІЇ РОБОТИ ПЕДАГОГІЧНИХ ПРАЦІВНИКІВ

2.1 Переваги та недоліки мови програмування Python та головне напрявлення при розробці коду

Python – інтерпретована об'єктно-орієнтована мова програмування високого рівня із суворою динамічною типізацією, тобто основна частина перевірок типів виконується під час виконання програми, а не під час компіляції. У динамічній типізації, значення мають типи, а змінні – ні, тому змінна може містити значення будь-якого типу.

Срінатх К.Р. у своїй статті вказує [4], що Python стає мовою першого вибору як в академічних колах, так і в комерційних проектах. Основними причинами цього визнано простоту синтаксису, багату бібліотечну підтримку, кросплатформеність та активну спільноту розробників.

Як було зазначено вище мова програмування Python має зручну систему модулів, а їх кількість та напрямленість дозволяє покрити майже будь які завдання. Однак Python має декілька суттєвих недоліків серед яких повільність, порівняно з іншими мовами програмування, та сильне обмеження при роботі з потоками. Сьогодні багатопотокове програмування є найбільш широко використовуваною парадигмою паралельного програмування для використання паралелізму в програмних додатках, які використовують переваги багатоядерних процесорів. Така техніка програмування може призвести до труднощів використання та проблем, які потребують вирішення [3]. Також при розробці великих додатків потрібно враховувати доволі не ефективне використання оперативної пам'яті, що може викликати різні проблеми. Однак для розробленої програми це недоліки не є суттєвими.

Таким чином мова програмування Python є одним з найкращих виборів при розробці нашої інтерактивної програми.

Розглянемо глобальні модулі мови програмування Python, що використані при розробці інтерактивного додатку «Асистент».

Tkinter є стандартним модулем мови програмування Python для створення GUI. Графічний інтерфейс користувача (англ. Graphical User Interface або GUI) – тип інтерфейсу, який дозволяє користувачеві взаємодіяти з комп'ютерною програмою або електронним пристроєм через графічні зображення та візуальні вказівки, на відміну від текстових інтерфейсів [10]. Модуль широко використовується у різних операційних системах, що однозначно говорить про його універсальність.

Перевагами модуля є:

- проста, що робить його зручним для початківців;
- широкий вибір різних віджетів, що значно полегшує створення графічного інтерфейсу;
- зручна система розміщення елементів у вікні за допомогою простих і зрозумілих методів;
- проста та ефективна система обробки подій, що дозволяє легко прив'язувати події до віджетів;
- застаріла і проста система візуальних налаштувань, що дозволить новим користувачам як старшого покоління так і молодшого швидко освоїтись у ньому.

Недоліки модуля є:

- недостатній набір налаштувань для редагування зовнішнього вигляду програми та загальна застарілість елементів;
- не призначений для використання його при розробці мобільних додатків.

Отже, не зважаючи на недоліки модуль Tkinter є одним з найпопулярніших модулів при розробці графічного інтерфейсу. А також цей модуль є ідеальним варіантом для розробки додатку «Асистент».

Модуль Openpyxl – це потужна бібліотека Python, призначена для читання та запису файлів Excel. Модуль був розроблений о 2010 році і нажаль на даний

момент часу більше офіційно не підтримується, однак це модуль став проектом з відкритим кодом і до сьогодні підтримується волонтерами у вільний час.

Переваги модуля:

- відсутність необхідності у встановленні Excel;
- підтримка всіх сучасних форматів Excel;
- максимальна незалежність від програмного забезпечення;
- повний контроль елементів книги, що дозволяє редагувати чи змінювати майже будь-який її аспект (клітинки, аркуші, стилі, формули, діаграми...);
- швидкість та ефективність у створенні різних звітів;
- можливість редагувати чи обробляти великі об'єми даних.

Недоліки модуля:

- надмірне навантаження на оперативну пам'ять;
- відсутність у модуля можливості зчитувати результати роботи комірки з функцією.

Враховуючи всі переваги та недоліки бібліотеки Openpyxl, можна стверджувати, що для створення локальної інтерактивної програми ця бібліотека є однією з найкращих варіантів через широкий асортимент взаємодії з книгою Excel, що дозволяє ефективно створювати звіти та реалізовувати автоматичне оновлення даних у них. Саме тому Excel буде використовуватися для збереження обробки та часткової демонстрації результатів.

Excel є однією з базових програм пакету Microsoft Office спроектована для роботи з електронними таблицями (табличний калькулятор). Вона може бути використана для обрахунків, створення простих таблиць, графіків, діаграм, баз даних, а також для створення статистик на базі великої кількості статистичних функцій [16]. Microsoft Excel, завдяки своїм множинним функціям та можливостям, є невід'ємною частиною робочого процесу в різних сферах [5, 6].

Існують різні підходи до розробки програмного забезпечення. Одним із найбільш популярних та ефективних є об'єктно-орієнтоване програмування (ООП). Python є мовою програмування, яка активно підтримує об'єктно-

орієнтованого програмування (ООП) та надає зручні механізми для створення класів, об'єктів та взаємодії між ними. ООП у Python реалізовано у гнучкій формі, що робить його зручним для моделювання та масштабованих проєктів [9]. ООП – це підхід до розробки програмного забезпечення, зосереджений на об'єктах, а не на функціях. Тобто програма розбивається на набір об'єктів, які взаємодіють один з одним. Структура ООП складається з об'єктів, класів, атрибутів і методів.

Принцип об'єктно-орієнтованого програмування базується на чотирьох головних елементах:

- інкапсуляція дає змогу приховувати його внутрішню реалізацію та надавати доступ тільки до інтерфейсу для подальшої взаємодії, що використовується для створення надійних програм запобігаючи несанкціонованому доступу;
- успадкування дає змогу створювати нові класи на основі існуючих для формування при необхідності ієрархії класів, що сприяє підвищенню ефективності розробки та забезпечує логічну організацію коду;
- поліморфізм дає змогу обробляти дані різного типу за допомогою єдиного інтерфейсу, що полегшує написання та підтримку коду;
- абстракція дає змогу розробникам приховати складність реалізації та зосередитися на ключових аспектах об'єкта.

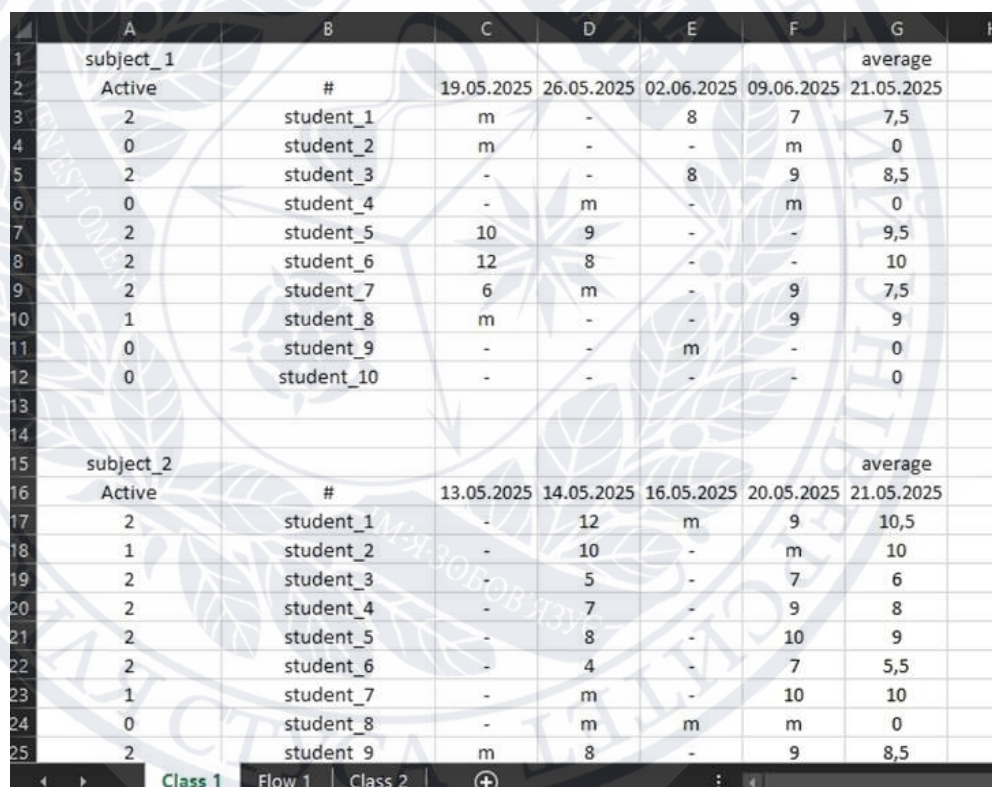
2.2 Створення прототипів інтерфейсу для інтерактивного додатку та структури журналу

Прототипування інтерфейсу – це процес створення попередньої, часто інтерактивної, моделі інтерфейсу для демонстрації чи тестування, перш ніж його буде розроблено остаточно. Цінність прототипу залежить від різних переваг, які він має. Такі, як здатність показати результат, зниження ймовірності неправильного розуміння або окремих її складових, точність, адаптивність, виявлення дефектів дизайну та роботи систем на ранній стадії. Крім того,

створення прототипу економить час, зусилля та гроші, а також зменшує обсяги відходів. При розробці прототипу важливо розуміти, для чого ви його створюєте [40].

Для створення чіткого уявлення інтерактивного додатку «Асистент» створено прототипи головних елементів програми. Це є одним з найважливіших етапів розробки програм, веб-сайтів тощо. Також це створює уявлення про відповідні елементи об'єкту які необхідні при розробці.

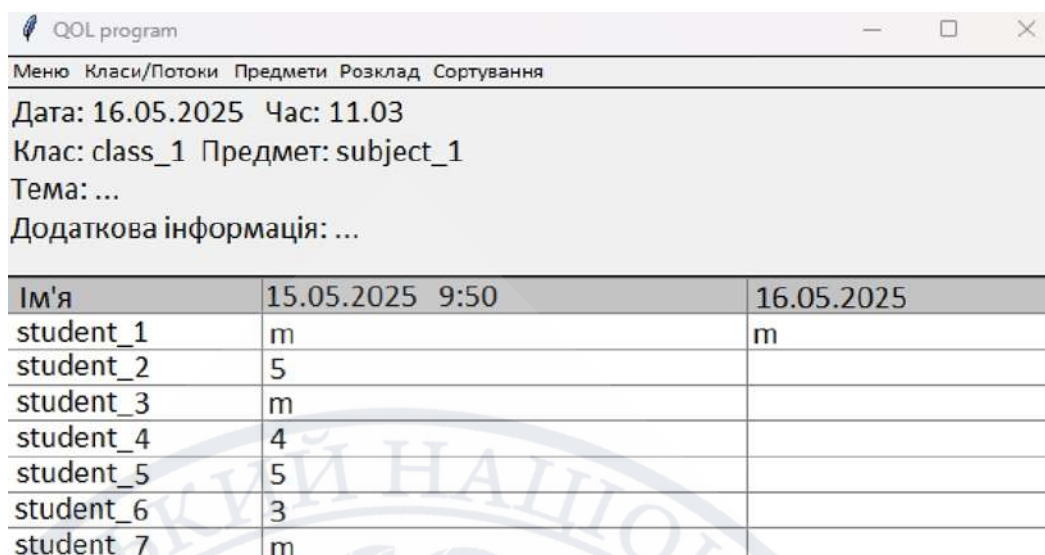
На рисунку 2.1 наведено прототип журналу для класу по певним предметам. Як можливо спостерігати продемонстрована можливість оцінки активності, автоматичний запис дат, та підведення підсумків по предметам.



	A	B	C	D	E	F	G	H
1	subject_1						average	
2	Active	#	19.05.2025	26.05.2025	02.06.2025	09.06.2025	21.05.2025	
3	2	student_1	m	-	8	7	7,5	
4	0	student_2	m	-	-	m	0	
5	2	student_3	-	-	8	9	8,5	
6	0	student_4	-	m	-	m	0	
7	2	student_5	10	9	-	-	9,5	
8	2	student_6	12	8	-	-	10	
9	2	student_7	6	m	-	9	7,5	
10	1	student_8	m	-	-	9	9	
11	0	student_9	-	-	m	-	0	
12	0	student_10	-	-	-	-	0	
13								
14								
15	subject_2						average	
16	Active	#	13.05.2025	14.05.2025	16.05.2025	20.05.2025	21.05.2025	
17	2	student_1	-	12	m	9	10,5	
18	1	student_2	-	10	-	m	10	
19	2	student_3	-	5	-	7	6	
20	2	student_4	-	7	-	9	8	
21	2	student_5	-	8	-	10	9	
22	2	student_6	-	4	-	7	5,5	
23	1	student_7	-	m	-	10	10	
24	0	student_8	-	m	m	m	0	
25	2	student_9	m	8	-	9	8,5	

Рисунок 2.1 – Макет готового журналу

На рисунку 2.2 продемонстровано прототип користувацького інтерфейсу. Як можна спостерігати тут продемонстрована можливість виводу інформації про поточне заняття та списку студентів з відмітками за попереднє заняття.



QOL program

Меню Класи/Потоки Предмети Розклад Сортвання

Дата: 16.05.2025 Час: 11.03
Клас: class_1 Предмет: subject_1
Тема: ...
Додаткова інформація: ...

Ім'я	15.05.2025 9:50	16.05.2025
student_1	m	m
student_2	5	
student_3	m	
student_4	4	
student_5	5	
student_6	3	
student_7	m	

Рисунок 2.2 – Макет інтерфейсу програми

2.3 Формування етапів роботи програми та їх функціоналу

Роботу програми поділено на два етапи: підготовчий та робочий етап. Ціль підготовчого етапу – це створення умов для будь яких запланованих дій користувача, таких як оновлення відповідних змінних або виведення відповідної інформації на екран. Робочий етап – це виконання відповідних функцій, що були викликані внаслідок дії користувача.

Опишемо необхідні елементи для проходження підготовчого етапу:

- відкриття робочої книги чи заповненого шаблону для подальшої роботи;
- функції отримання та оновлення теперішнього часу;
- функції знаходження матеріалу в залежності від встановленого часу;
- функції отримання та збереження відповідних матеріалів в залежності від результатів роботи з розкладом;
- функції виведення на екран.

Опишемо необхідні елементи для проходження робочого етапу:

- функція оновлення елементів (зміна тексту, заповнення та виведення нової інформації);
- функція реакції (вибір елементу, натиск на кнопку, тощо);

- паралельна функція для оновлення відповідних параметрів (часу).

Розглянемо підготовчі функції.

- а) Робоча книга чи початковий шаблон для подальшої роботи.

Для створення нової робочої книги необхідно слідувати шаблону та заповнити відповідну інформацію до нової книги Excel. Вирішено винести заповнення базової інформації за функціонал інтерактивного додатку «Асистент». Це зроблено через відсутність необхідності у постійному чи періодичному редагуванні базової інформації. Програма розрахована на разове заповнення інформації на початку навчального року та використання цієї інформації протягом всього навчального року. Після створення та заповнення робочої книги Excel можна продовжувати роботу програми. У цій функції відбувається також ініціалізація книги для подальшого звернення до її сторінок.

- б) Функція отримання та оновлення теперішнього часу.

Функція проводить обрахунок та внесення відповідної інформації для подальшого використання наступних функцій. Використовується модуль datetime для обрахування типу тижня та запису теперішнього часу.

- в) Функції знаходження матеріалу в залежності від встановленого часу.

Функція проводити пошук відповідних записів з робочої книги. За умови успішного виконання попередньої функції повертає відповідні параметри (клас, предмет, номер заняття, кількість занять сьогодні). За умови невдачі виконує перевірку чи проводяться сьогодні заняття та передає відповідну інформацію.

- г) Функція створення матеріалів для виведення їх на робоче вікно інтерактивного додатку.

Функція проводить перевірку журналу на наявність відповідного запису. Якщо запис знайдено, то повертає адрес комірки відповідного предмету, інакше створює відповідний запис самостійно.

- д) Функції виведення на екран.

Створює робоче вікно з необхідними елементами та виводить інформацію в залежності від результатів роботи попередніх функцій.

Розглянемо функції робочого етапу:

а) Функція оновлення елементів.

Проводить повне очищення робочого вікна від усіх елементів та створює їх знову з новою інформацією

б) Функція реакції (вибір елементу, натиск на кнопку, тощо).

В залежності від дій користувача запускає відповідну функцію:

- функцію вибору елементу;
- функцію натиску на кнопку;
- функцію вибору елемента меню;
- функцію додавання чи оновлення даних.

в) Паралельна функція для оновлення відповідних параметрів (часу).

Запускає годинник для регулярного оновлення інформації про час.

г) Функція зміни книги.

Дозволяє обрати книгу з відповідної папки та перевіряє чи заповнена книга інформацією та проводить заповнення її додатковими параметрами.

Проаналізувавши всі переваги та недоліки мови програмування Python, визначено, що для реалізації інтерактивного додатку «Асистент» мова програмування Python та його модулі Tkinter та Openpyxl є найкращим вибором. Також обрано головний напрямок при розробці інтерактивного додатку, а саме ООП (об'єктно орієнтоване програмування).

На основі аналізу сформованих вимог додатку створенні прототипи для його інтерфейсу та робочої книги Excel, що допомогло покращити розуміння необхідних процесів та функцій для його реалізації. Таким чином, сформовано основні етапи розробки додатку «Асистент».

РОЗДІЛ 3

РЕАЛІЗАЦІЯ ІНТЕРАКТИВНОГО ДОДАТКУ ТА ПЕРЕВІРКА ЙОГО ПРАЦЕЗДАТНОСТІ

3.1 Підготовка даних для подальшої активації книги

Під час першого запуску інтерактивного додатку «Асистент» активується шаблон робочої книги Excel. Для початку повноцінної роботи необхідно: створити нову книгу (скопіювати книгу-шаблон) та заповнити її відповідно книзі шаблону; активувати її через інтерактивний додаток «Асистент».

На рисунку 3.1 продемонстровано сторінку книги шаблону з прикладом заповнення сторінки «technical_page».

	A	B	C	D	E	F	G	H	I	J
1	08:00	09:00	10:00	11:00	12:00	13:00	14:00	15:00	16:00	
2	subject_1	subject_2	subject_3	subject_4						
3										
4										

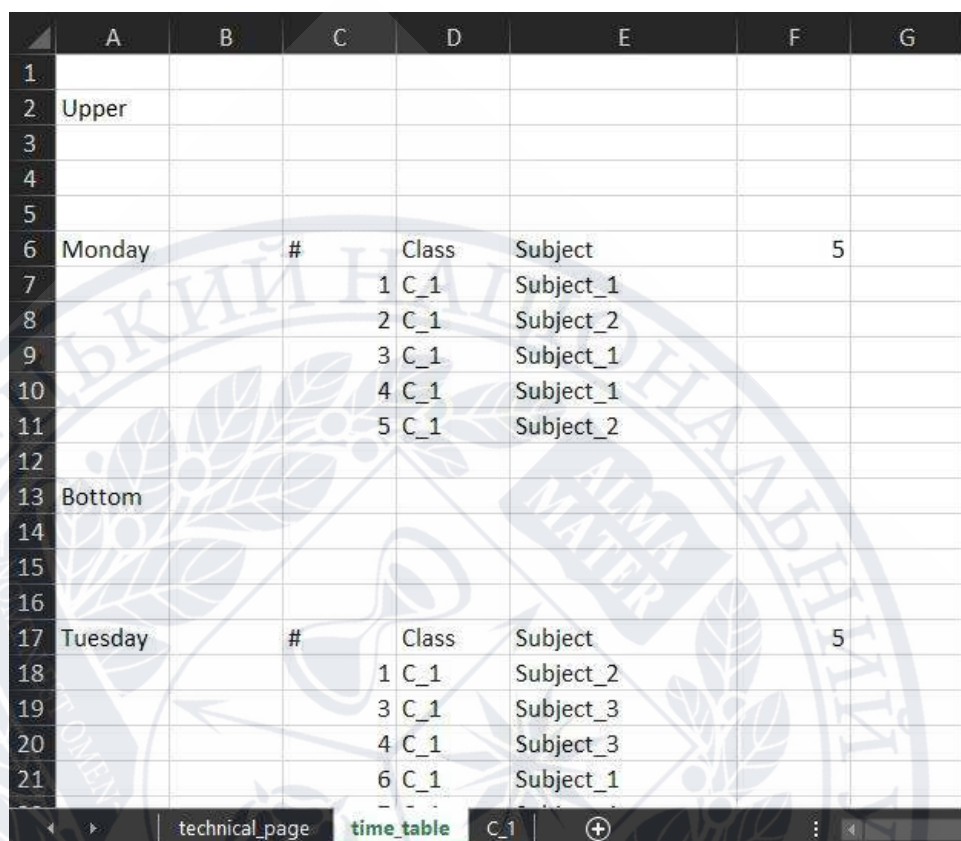
Рисунок 3.1 – Сторінка книги шаблону «technical_page»

На рисунку 3.2 демонструється сторінка книги шаблону з прикладу заповнення сторінки класу.

	A	B	C	D	E	F	G
1							
2			#	Im'a			
3				1 Name_student_1			
4				2 Name_student_2			
5				3 Name_student_3			
6				4 Name_student_4			
7				5 Name_student_5			
8				6 Name_student_6			
9				7 Name_student_7			
10				8 Name_student_8			
11				9 Name_student_9			
12				10 Name_student_10			
13				11 Name_student_11			
14				12 Name_student_12			
15				13 Name_student_13			
16				14 Name_student_14			
17				15 Name_student_15			
18				16 Name_student_16			
19				17 Name_student_17			
20				18 Name_student_18			
21				19 Name_student_19			
22				20 Name_student_20			
23				21 Name_student_21			
24				22 Name_student_22			
25				23 Name_student_23			

Рисунок 3.2 – Сторінка книги шаблону «class_name»

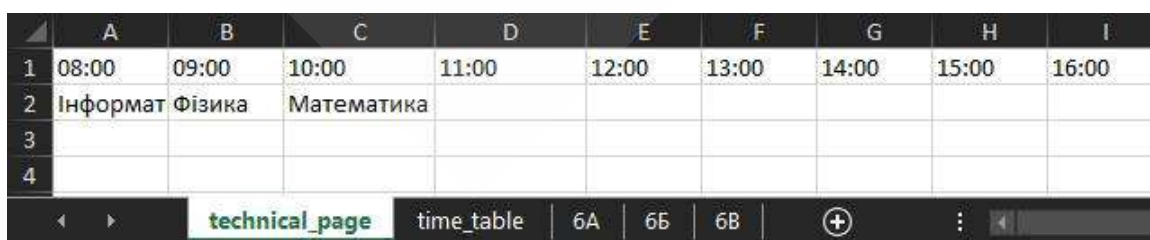
На рисунку 3.3 демонструється шаблон сторінки «time_table» з прикладом внесення відповідної інформації.



	A	B	C	D	E	F	G
1							
2	Upper						
3							
4							
5							
6	Monday	#	Class	Subject		5	
7			1 C_1	Subject_1			
8			2 C_1	Subject_2			
9			3 C_1	Subject_1			
10			4 C_1	Subject_1			
11			5 C_1	Subject_2			
12							
13	Bottom						
14							
15							
16							
17	Tuesday	#	Class	Subject		5	
18			1 C_1	Subject_2			
19			3 C_1	Subject_3			
20			4 C_1	Subject_3			
21			6 C_1	Subject_1			

Рисунок 3.3 – Сторінка книги шаблону «time_table»

Для коректної роботи програми потрібно заповнити Excel файл, що використовуватиметься як робоча книга відповідно до книги шаблону. Детальна інструкція наведена в Додатку Б. На рисунку 3.4 продемонстрована заповнена сторінка «technical_page» для створення нової робочої книги.



	A	B	C	D	E	F	G	H	I
1	08:00	09:00	10:00	11:00	12:00	13:00	14:00	15:00	16:00
2	Інформат	Фізика	Математика						
3									
4									

Рисунок 3.4 – Сторінка заповненої книги шаблону «echnical_page»

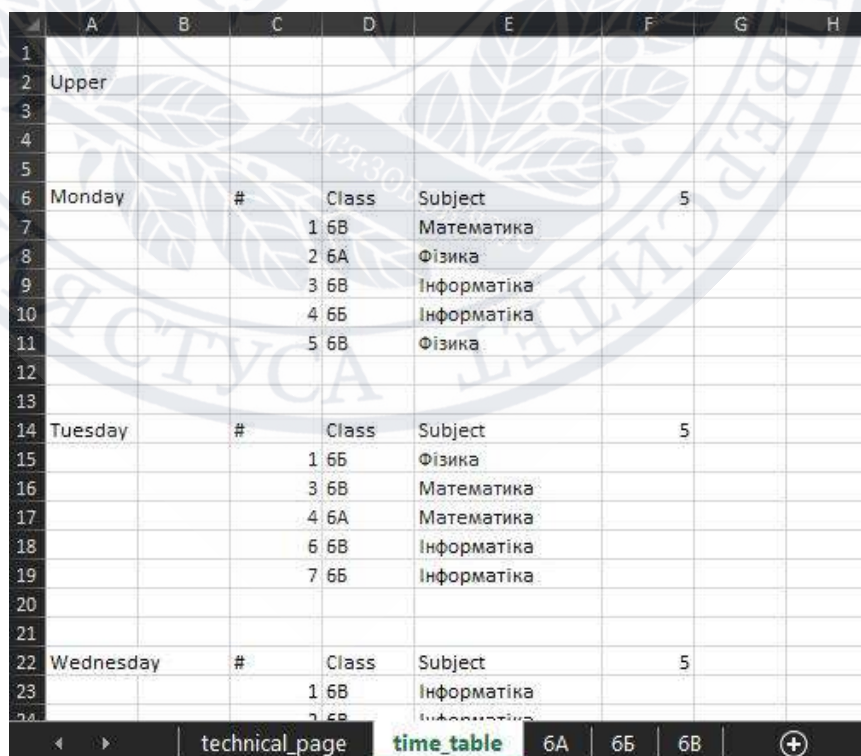
На рисунку 3.5 наведений приклад заповненої сторінки класу.



	A	B	C	D	E	F	G
1							
2			#	Ім'я			
3			1	А. Аліна			
4			2	Б. Артем			
5			3	Б. Богдан			
6			4	Б. Владислав			
7			5	Б. Олексій			
8			6	Б. Олександра			
9			7	В. Марія			
10			8	В. Ахмад			
11			9	Д. Аріна			
12			10	З. Андрій			
13			11	З. Дмитро			
14			12	З. Артем			
15			13	І. Вероніка			
16			14	К. Маргарита			
17			15	К. Домініка			
18			16	К. Максим			
19							
20							

Рисунок 3.5 Сторінка заповненої книги шаблону «class_name»

На рисунку 3.6 наведений приклад заповнення сторінки «time_table».



	A	B	C	D	E	F	G	H
1								
2	Upper							
3								
4								
5								
6	Monday		#	Class	Subject	5		
7			1	6B	Математика			
8			2	6A	Фізика			
9			3	6B	Інформатика			
10			4	6B	Інформатика			
11			5	6B	Фізика			
12								
13								
14	Tuesday		#	Class	Subject	5		
15			1	6B	Фізика			
16			3	6B	Математика			
17			4	6A	Математика			
18			6	6B	Інформатика			
19			7	6B	Інформатика			
20								
21								
22	Wednesday		#	Class	Subject	5		
23			1	6B	Інформатика			
24			2	6B	Інформатика			

Рисунок 3.6 – Сторінка заповненої книги шаблону «time_table»

Після заповнення робочої книги необхідно пройти її активацію, що проаналізує заповнені сторінки та на їх основі сформує необхідну інформацію для роботи інтерактивного додатку «Асистент». У нище наведеному фрагменті коду відбувається виведення на екран списку доступних для активації книг:

```
def change_work_book(self):
    self.reload(2)
    Assistant.var_c = 1
    list_f = os.listdir("data")
    heddd = ['WorkBook:']

    self.set_menu
    self.label_info_1 = tk.Label(Assistant.window,
font=("Times New Roman", 16), justify=tk.CENTER, text = "Оберіть
робочу книгу!\n Після підтвердження програма буде закрита
автоматично!")
    self.btn_1 = tk.Button(Assistant.window, text="Змінити
робочу книгу", font=("Times New Roman", 16),
command=lambda:self.activate_book())

    self.create_wg()

    self.label_info_1.place(relwidth=0.5, relheight=0.10,
relx=0.25, rely=0.24)
    self.btn_1.place(relheight=0.08, relx=0.4, rely=0.36)
    self.table_lable(list_f, heddd)
    self.time_now_d()
```

Далі потрібно перевірити чи правильно заповнена книга та виконати додавання необхідної інформації до сторінки «technical_page»:

```
while True:
    if sheet_tp.cell(row=1, column=i).value != None:
        add_inf = add_inf + str(sheet_tp.cell(row=1,
column=i).value) + " "
    else:
        base_inf.append(add_inf[:-1])
        break
    i = i + 1

add_inf = add_inf.split(" ")
list_wt = Assistant.basic_inf[3]
list_wt = list_wt.split(" ")

if sheet_tt_cn.cell(row=2, column=1).value == None:
    self.label_info_1.configure(text="Не коректно
заповнена книга!")
```

```

        return
    else:
        if sheet_tt_cn.cell(row=2, column=1).value ==
list_wt[1]:
            i = 3
            while i <= sheet_row:
                if sheet_tt_cn.cell(row=i, column=1).value
== list_wt[0]:
                    sheet_tp.cell(row=3, column=3,
value=str(i))
                    break
                i = i + 1
            sheet_tp.cell(row=3, column=2, value=2)

```

У фрагменті наведеному далі відбувається створення списку існуючих класів та запис кількості їх учнів до сторінки «technical_page». Також здійснюється заповнення спеціальної змінної «base_inf» для оновлення файлу з базовими значеннями.

```

i = 5
    while i <= sheet_row:
        if sheet_tt_cn.cell(row=i, column=3).value == "#":
            j = 1
            while sheet_tt_cn.cell(row=i + j,
column=3).value != None:
                time = int(sheet_tt_cn.cell(row=i + j,
column=3).value)
                sheet_tt_cn.cell(row=i + j, column=2,
value=add_inf[time - 1])
                j = j + 1
            i = i + 1
        i = 2
        add_inf = ''

    while i < len(s_list):
        sheet_tt_cn = work_book[s_list[i]]
        sheet_tt_cn.column_dimensions['D'].width = 30
        add_inf = add_inf + s_list[i] + "|"
        j = 3
        while sheet_tt_cn.cell(row=j, column=3).value !=
None:
            j = j + 1
        sheet_tp.cell(row=i + 2, column=1,
value=s_list[i])
        sheet_tp.cell(row=i + 2, column=2, value=j - 3)
        i = i + 1

```

```

base_inf.append(Assistant.basic_inf[3])
base_inf.append(Assistant.basic_inf[4])
base_inf.append(add_inf[:-1])

add_inf = ''
i = 1
while True:
    if sheet_tp.cell(row=2, column=i).value != None:
        add_inf = add_inf + str(sheet_tp.cell(row=2,
column=i).value) + "|"
    else:
        base_inf.append(add_inf[:-1])
        break
    i = i + 1

```

Зберігаємо активовану книгу, перезаписуємо файл з базовою інформацією та закриваємо вікно для оновлення даних програми:

```

work_book.save(f'./data/{Assistant.value[0]}')
write_base_inf(base_inf)
self.window.destroy()

```

На рисунку 3.7 продемонстровано візуалізацію вибору нової робочої книги.

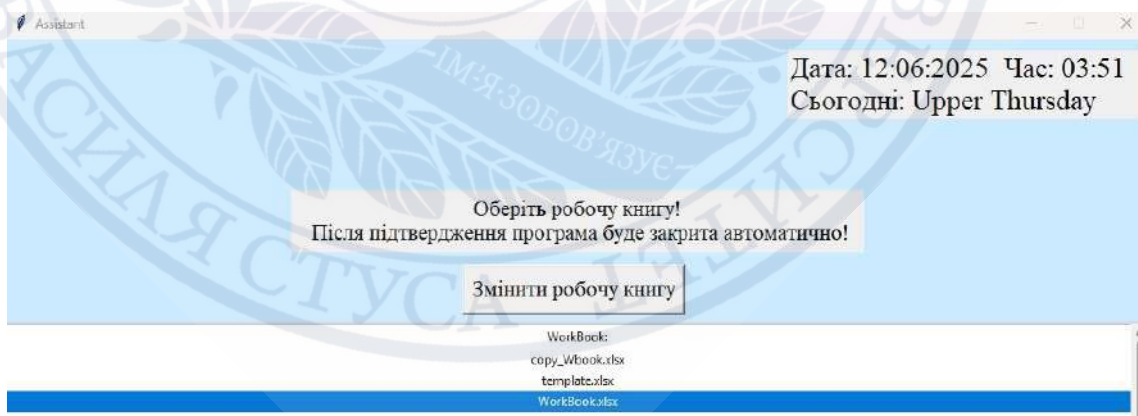


Рисунок 3.7 – Візуалізація вибору нової робочої книги

Внаслідок підтвердження вибору нової робочої книги, робота програми завершиться для оновлення її інформації.

На рисунку 3.8 продемонстровано сторінку нової робочої книги «technical_page», де відображається список класів з кількістю учнів та стартове положення у розкладів типів тижня.

	A	B	C	D	E	F	G
1	08:00	09:00	10:00	11:00	12:00	13:00	14:00
2	Інформат	Фізика	Математика				
3			2 29				
4	6A	16					
5	6Б	13					
6	6В	15					
7							
8							
9							
10							
11							
12							

Рисунок 3.8 – Сторінка активованої робочої книги «technical_page»

На рисунку 3.9 продемонстровано сторінку нової робочої книги «time_table», де можливо переглянути розклад та час початку кожного уроку.

	A	B	C	D	E	F	G
1							
2	Upper						
3							
4							
5							
6	Monday	#	Class	Subject		5	
7		08:00	1 6B	Математика			
8		09:00	2 6A	Фізика			
9		10:00	3 6B	Інформатика			
10		11:00	4 6Б	Інформатика			
11		12:00	5 6B	Фізика			
12							

Рисунок 3.9 – Сторінка активованої робочої книги «time_table»

3.2 Створення класу, функції візуалізації інтерфейсу

Створюємо клас «Assistant», що є головним елементом при розробці інтерактивного додатку: ініціалізуємо робочу книгу для подальшого зчитування інформації та запису її, створюємо робоче вікно та задаємо відповідні параметри з класу, викликаємо створення масиву «basic_inf» для збереження та постійного використання базової інформації. Наведемо відповідний фрагмент коду:

```
class Assistant:
    basic_inf = get_basic_inf_f()
    name_ak = basic_inf[0]
    work_book = load_workbook(f'./data/{name_ak}')
    h_main = 600
    w_main = 1100

    window = tk.Tk()
    window.title('Assistant')
    window.geometry(f"{w_main}x{h_main}+300+150")
    window.resizable(False, False)

    day_week_now = strftime('%A')
    hour_minute_now = strftime('%H:%M')
    week_s = ''

    continue_t_ln = []
    save_l_n = [[], []]
    value = []
    value_now = []
    value_r = []
    var_c = 0
    boll_1 = True
```

Функції «__init__» викликає ініціалізацію всіх елементів робочого вікна для подальшого їх використання (зміни, форматування, виведення або видалення з робочого вікна).

Функції «create_wg», «create_info» та «create_interactive» здійснюють налаштування та виведення створених об'єктів на робоче вікно:

```
def create_wg(self):
    self.label_info.place(relwidth=1, relheight=0.50, relx=0,
        rely=0)
```

```

        self.label_table.place(relwidth=1, relheight=0.55, relx=0,
                                rely=0.45)
        self.label_time.place(x=752, y=10, relwidth=0.31)

        self.table.pack(expand=tk.YES, fill=tk.BOTH)
        self.scroll_p.pack(side="right", fill="y")

    def create_info(self):
        self.label_info_1.place(relwidth=0.67, relheight=0.16,
                                x=10, y=10)
        self.label_info_2.place(relwidth=0.67, relheight=0.16,
                                x=10, rely=0.18)
        self.label_info_3.place(relwidth=0.3, relheight=0.09,
                                x=10, rely=0.35)

    def create_interactive(self, a):
        if a == 0:
            self.entry_1.place(relwidth=0.05, relheight=0.08,
                                x=350, rely=0.351)
            self.btn_1.place(relheight=0.08, x=415, rely=0.352)
            self.btn_2.place(relheight=0.08, x=560, rely=0.352)
        else:
            self.btn_3.place(relheight=0.08, x=345, rely=0.353)

```

Створюємо функцію «set_menu» для ініціалізації та виведення елементів меню на робоче вікно:

```

def set_menu(self):
    menubar = tk.Menu(self.window)
    self.window.config(menu=menubar)

    set_menu = tk.Menu(menubar, tearoff=0)
    set_menu_test_f = tk.Menu(menubar, tearoff=0)

    set_menu.add_command(label='Демонстраційний
файл', command=lambda: self.open_file_c())
    set_menu.add_command(label='Змінити файл',
command=lambda: self.change_work_book())
    set_menu.add_command(label='Створити заняття',
command=self.create_lesson)
    set_menu.add_command(label='Вихід',
command=self.window.destroy)
    set_menu.add_command(label='Робоча книга',
command=lambda: self.open_file(self.name_ak))
    menubar.add_cascade(label='Базові функції', menu=set_menu)

    set_menu_test_f.add_command(label='Створення заг. SUM',
command=self.create_z_sum)

```

```

        set_menu_test_f.add_command(label='Створення заг.
AVERAGE', command=self.create_z_aver)
        set_menu_test_f.add_command(label='Створення SUM',
command=self.create_sum)
        set_menu_test_f.add_command(label='Створення AVERAGE',
command=self.create_aver)
        menubar.add_cascade(label='Підсумки',
menu=set_menu_test_f)

```

На рисунку 3.10 та 3.11 продемонстровано можливість виведення списку для демонстрації доступних функцій.

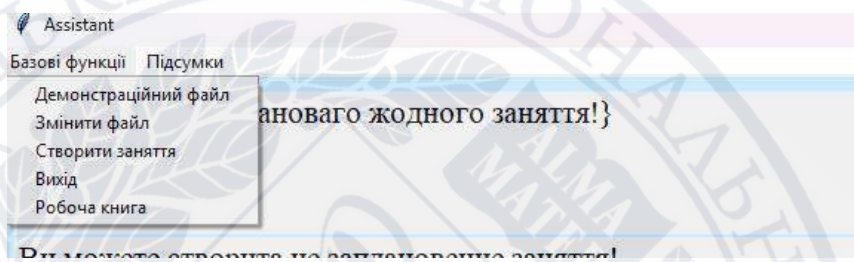


Рисунок 3.10 – Демонстрація роботи функції «set_menu», вкладки базові функції

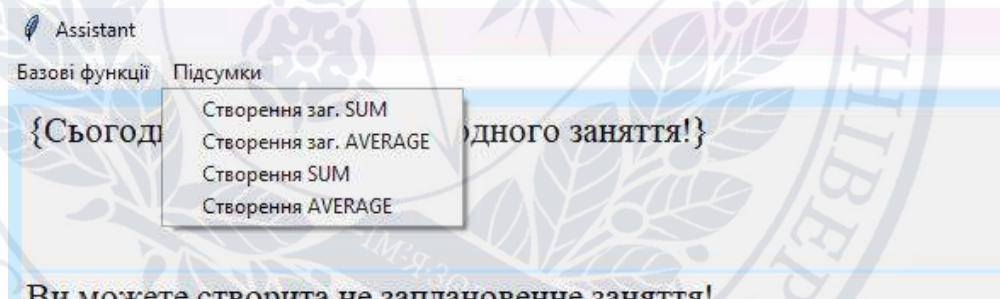


Рисунок 3.11 – Демонстрація роботи функції «set_menu», вкладки підсумки

На рисунку 3.12 та 3.13 демонстровано зовнішній вигляд робочого вікна інтерактивного додатку «Асистент».

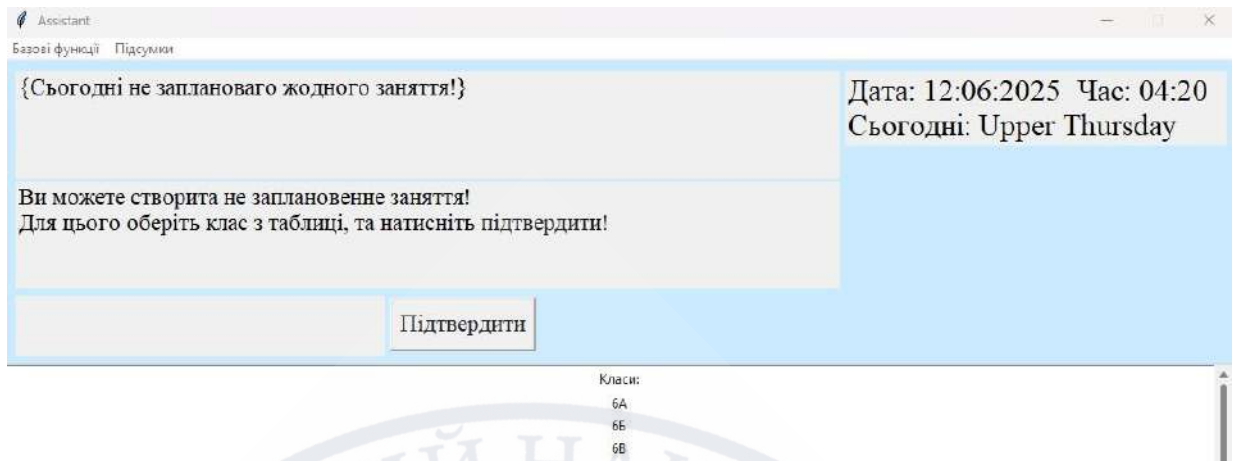


Рисунок 3.12 –функцій «create_lesson»

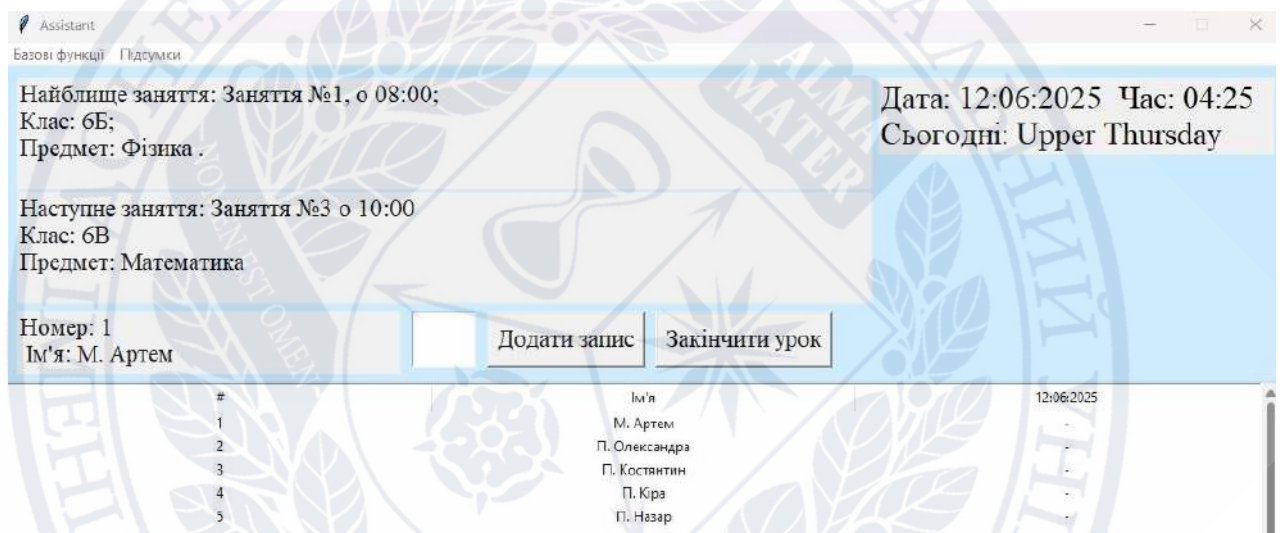


Рисунок 3.13 – Стандартний зовнішній вигляд при запланованому занятті

3.3 Функції що відповідають за роботу з часом та вторинні функції

Функція «now_week» відповідає з виявлення та оновлення інформації про тип тижня. Вона отримує номер тижня з модуль «Datetime» та перевіряє чи є він парним або непарним:

```
def now_week(self, structure_timetable):
    sheet_t = Assistant.work_book['technical_page']
    date_obj = datetime.datetime.now()
    date_obj = int(date_obj.strftime("%V")) % 2
    structure_timetable = structure_timetable.split(' ')
    if date_obj == 0:
```

```

        sheet_t.cell(row=1, column=4,
value=structure_timetable[1])
    else:
        sheet_t.cell(row=1, column=4,
value=structure_timetable[0])
    self.seve_wb()

```

Функція «day_week_l_n» відповідає за знаходження та повернення відповідної комірки у розкладі для подальшої взаємодії:

```

def day_week_l_n(self, i, day_week_now, l_row):
    sheet_time_t = Assistant.work_book['time_table']
    l_week = []
    i_and_j = [i, 0]

    i1 = i + 4
    j = 0
    while i1 < l_row:
        if sheet_time_t.cell(row=i1, column=1).value ==
day_week_now:
            j = sheet_time_t.cell(row=i1, column=6).value
            i_and_j = [i1, j]
            break
        i1 = i1 + 1
    return i_and_j

```

Функція «time_now_d» відповідає за створення та підтримку роботи годинника:

```

def time_now_d(self):
    time = strftime('Дата: %d:%m:%Y Час: %H:%M')
    time = time + f"\nСьогодні: {Assistant.week_s}
{Assistant.day_week_now}"
    self.label_time.config(text=time)
    self.label_time.after(60000, self.time_now_d)

```

Головним завдання функції «now_lesson_and_next» є орієнтація в часі, тобто вона знаходить заняття, що незабаром відбудеться. Якщо таких занять не знайдено, то виводиться відповідне повідомлення та виконується запуск функції «create_lesson», що запускає створення незапланованого заняття (ця функція продемонстрована на рисунку 3.13).

В коді наведеному далі відбувається перевірка на тип тижня:

```

if Assistant.week_s == structure_timetable[0]:
    i = sheet_t.cell(row=3, column=3).value
else:
    if Assistant.week_s == structure_timetable[1]:
        i = sheet_t.cell(row=3, column=2).value
    else:
        i = 2

```

Далі здійснюється перевірка чи запланований урок найближчим часом:

```

if j == 0 :
    list_c_s_st.append("Сьогодні не заплановано жодного
заняття!")
    Assistant.var_c = 1
elif len(Assistant.save_l_n[1]) == 4:
    while t < j:
        if sheet_time_t.cell(row=i + t + 1,
column=3).value == Assistant.save_l_n[1][1]:
            tum = 1
            break
            t = t + 1
    else:
        while t < j:
            nom_lesson = int(sheet_time_t.cell(row=i + t + 1,
column=3).value)
            minute_lesson = cal_minute(list_time[nom_lesson -
1])
            min = 10
            try:
                minute_lesson_s =
cal_minute(list_time[nom_lesson])
            except:
                minute_lesson_s =
cal_minute(list_time[nom_lesson - 1])
                min = - (minute_lesson_s -
cal_minute(list_time[nom_lesson - 2]) - 10)

            if minute_lesson - 10 < minute_now and minute_now
< minute_lesson_s - min:
                tum = 1
                break
                t = t + 1
        if minute_now > minute_lesson + 50:
            list_c_s_st.append("Сьогодні були вже проведені
усі заняття!")
            Assistant.var_c = 1

```

Перевірка чи проведені вже всі заняття:

```

if Assistant.var_c != 1:
    if tum != 1:
        t = 0
        n = 1
        while n < 3:
            list_now_lesson = []
            j = 2
            while j < 6:
                if n == 2:
                    if sheet_time_t.cell(row=i + n + t,
column=j).value == None:
                        list_now_lesson.append("Кінець")
                        break
                    else:
                        list_now_lesson.append(sheet_time_t.cell(row=i + n + t,
column=j).value)
                else:
                    list_now_lesson.append(sheet_time_t.cell(row=i + n + t,
column=j).value)
                j = j + 1
            list_c_s_st.append(list_now_lesson)
            n = n + 1
        return list_c_s_st

```

Вторинні функції «len_row», «read_list», «change_label_info», «get_value_stud», «take_obj», «seve_wb», «table_info» та «text_colum» є короткими функціями, що оптимізують отримання часто використовуваних даних.

Функція «find_space_new_mark» орієнтується в мітках занять для знаходження місця для нових оцінок (балів, відміток):

```

def find_space_new_mark(self, class_name, name_sbj,
value_stud):
    sheet_class = Assistant.work_book[class_name]
    i = value_stud + 5
    while True:
        if sheet_class.cell(row=i, column=2).value ==
name_sbj:
            j = 5
            while True:
                if sheet_class.cell(row=i, column=j).value ==
"F" or sheet_class.cell(row=i, column=j).value == None:
                    break

```

```

        j = j + 1
    break
    i = i + value_stud + 4
    i_and_j = [i, j]
    return i_and_j

```

Функція «add_new_mark_st» додає відповідні оцінки до журналу для заповнення заняття:

```

def add_new_mark_st(self):
    value_e = self.entry_1.get()
    try:
        value_e = int(value_e)
    except ValueError:
        None
    value_n = Assistant.value_now
    Assistant.value.append(value_e)
    value = Assistant.value

    if len(value) == 4:
        value = [value[0], value[1], value[3]]
        Assistant.value = value

    sheet_class = Assistant.work_book[value_n[0]]
    value_stud = self.get_value_stud(value_n[0])
    i_and_j = self.find_space_new_mark(value_n[0], value_n[1],
value_stud)
    i = i_and_j[0]
    j = i_and_j[1]

    sheet_class.cell(row=i, column=j, value="F")
    sheet_class.cell(row=i+1, column=j,
value=strftime('%d:%m:%Y'))
    t = 1
    while t <= value_stud:
        if sheet_class.cell(row=i+1+t, column=3).value ==
value[0]:
            sheet_class.cell(row=i + 1 + t, column=j,
value=value[2])
            break
        t = t + 1

    self.seve_wb()
    self.reload(1)

```

Функція «table_lable» заповнює таблицю інформацією:

```

def table_table(self, list_t_t, heddd):
    len_heddd = len(heddd)
    if len_heddd == 4 and heddd[2] == heddd[3]:
        heddd[2] = heddd[2] + ' '
        self.table['columns'] = heddd

    if len_heddd > 1:
        for header in heddd:
            self.table.heading(header, text=header,
                                anchor="center")
            self.table.column(header, anchor='center')
        else:
            self.table.heading(heddd, text=heddd[0],
                                anchor="center")
            self.table.column(heddd, anchor='center')
    for row in list_t_t:
        self.table.insert('', 'end', values=row)

    self.scroll_p = ttk.Scrollbar(self.label_table,
                                   command=self.table.yview())
    self.table.configure(yscrollcommand=self.scroll_p.set)

```

Функція «create_new_subject_class» створює предмет у відповідному класі коли такий є відсутнім:

```

def create_new_subject_class(self, name_subject, name_class,
row):
    value_stud = self.get_value_stud(name_class)
    sheet_class = Assistant.work_book[name_class]
    sheet_class.cell(row=row + 3, column=2, value=name_subject)
    sheet_class.cell(row=row + 4, column=2, value='Active')
    sheet_class.cell(row=row + 4, column=3, value='#')
    sheet_class.cell(row=row + 4, column=4,
value="Ім'я/Date:")

    sheet_class.cell(row=row + 3, column=4, value="stun:")
    list_st = self.read_list(3, 4, name_class, value_stud)

    i = self.len_row(name_class) + 1
    v_st = len(list_st) + i
    n_st = 0

    while i < v_st:

```

```

sheet_class.cell(row=i, column=3, value=n_st + 1)
sheet_class.cell(row=i, column=4, value=list_st[n_st])
n_st = n_st + 1
i = i + 1
self.seve_wb()

```

Функція «lit_colum» створює список з буквеного позначення колонок, що відповідає переліку проведених занять та повертає його:

```

def lit_colum(self, i):
    sheet_class = Assistant.work_book[Assistant.value_now[0]]
    dubl = [get_column_letter(5)]
    bol_i = False
    k = 6
    while True:
        if sheet_class.cell(row=i, column=k).value != None:
            if bol_i == True:
                dubl.append(get_column_letter(k))
                bol_i = False
            if sheet_class.cell(row=i, column=k).value == "A"
or sheet_class.cell(row=i, column=k).value == "S":
                dubl.append(get_column_letter(k - 1))
                bol_i = True
        else:
            if bol_i == True:
                break
            dubl.append(get_column_letter(k - 1))
            break
        k = k + 1
    return dubl

```

Функція «cr_ex_f» відповідає за створення готової функції Excel (створює формули з окремих елементів для подальшого запису до комірок Excel):

```

def cr_ex_f(self, i, j, value_stud, dubl, str_p, a):
    sheet_class = Assistant.work_book[Assistant.value_now[0]]
    i = i + 2
    value_stud = value_stud + i
    bol_i = True
    while i < value_stud:
        if a == 1 :
            k = len(dubl) - 2

```

```

else:
    k = 0
    str_f = str_p
    while k < len(dubl):
        dubl[k]
        str_f = str_f + dubl[k] + str(i)
        if bol_i == True:
            str_f = str_f + ':'
            bol_i = False
        else:
            str_f = str_f + ','
            bol_i = True
        k = k + 1
    if a == 2 :
        str_f = str_f[0:len(str_f) - 1] + ')'
        j = 2
    else:
        str_f = str_f[0:len(str_f) - 1] + '),0)'
    sheet_class.cell(row=i, column=j, value=str_f)
    i = i + 1

```

Функція «create_or_update_Active» відповідає за оновлення функції «Active» після закінчення уроку для оновлення статистики активних учнів:

```

def create_or_update_Active(self, i, j, value_stud):
    class_name = Assistant.value_now[0]
    sbj_name = Assistant.value_now[1]
    sheet_class = Assistant.work_book[class_name]
    dubl = self.lit_colum(i)
    str_f = '=COUNT('
    self.cr_ex_f(i, j, value_stud, dubl, str_f, 2)
    self.seve_wb()

```

Рисунок 3.14 показує роботу оновленої функції «create_or_update_Active» за допомоги вторинних функцій «lit_colum» та «cr_ex_f».

МИН

✕

✓

fx

=СЧЁТ(E22:H22;J22:L22)

	A	B	C	D	E	F	G	H	I	J	K	L
21		Active	#	Im'я/Date:	22.05.2025	27.05.2025	29.05.2025	03.06.2025	AVERAGE	05.06.2025	10.06.2025	12.06.2025
22		J22:L22	1	Д. Аріна	8	8	1 m	5,666667	m	m	10	
23		5	2	З. Андрій	8 m		2	10	6,666667		11	2
24		4	3	З. Дмитро	2 m	m		10	6	6 m		11
25		6	4	З. Артем	10 m		5	6	7	10	5	6
26		5	5	І. Вероніка		12	2	12	8,666667	9		9
27		6	6	К. Маргарита	4		12	10	8,666667	2	4	4
28		4	7	К. Домініка	6	5		2	4,333333	4 m	m	
29		6	8	К. Максим	7	10	4	6	6,75	1	7 m	
30		5	9	М. Артем	5 m			11	8	2	12	4
31		4	10	П. Олександра		m	5	9	7	1	6	
32		4	11	П. Костянтин	m		12	4	8		5	10
33		3	12	П. Кіра	1	m		10	5,5	m		6
34		5	13	П. Назар	3	1	3	9	4 m		1 m	

technical page

time table

6A

6E

6B

+

Рисунок 3.14 – Демонстрація роботи функції «Active»

3.4 Функції, що відповідають за запуск та виведення інформації

Функція «start_pr» запускає робоче вікно та ініціалізує ще декілька функцій, що відповідають за виведення інформації на об'єкти робочого вікна.

```
def start_pr(self, a):
    self.create_wg()
    self.set_menu()
    self.now_week(Assistant.basic_inf[3])
    if a == 0:
        self.start_f()
    else:
        self.start_save()
    self.time_now_d()
    self.window.mainloop()
```

Функція «reload», що відповідає за оновлення усіх об'єктів робочого вікна. Вона використовується для оновлення інформації.

```
def reload(self, a):
    [child.destroy() for child in
self.window.winfo_children()]
    self.__init__()

    if a == 0:
        self.start_pr(0)
    elif a == 1:
        self.start_pr(1)
    else:
        None
```

Функція «text_table_input_1_n» витягує відповідну інформацію з журналу для заповнення таблиці. Використовує систему міток для кожного заняття та орієнтації в заняттях.

Далі в коді продемонстрована орієнтація у мітках:

```
while True:
    if sheet_class.cell(row=i, column=2).value == sbj:
        j = 5
        while True:
            if sheet_class.cell(row=i, column=j).value ==
"F" or sheet_class.cell(row=i, column=j).value == None:
                t = 1
                if sheet_class.cell(row=(i), column=j -
1).value == "A" or sheet_class.cell(row=(i), column=j - 1).value
== "S":
                    y = -1
```

Продемонструємо частину коду, що відповідає за заповнення та виведення елементів таблиці:

```
        if j != 5:
            time = sheet_class.cell(row=(i+1),
column=j-1+y).value
            while t <= value_stud:
                if j == 5:
                    mark_2 = sheet_class.cell(row=(i +
1 + t), column=j).value
                    if mark_2 == None:
                        mark_2 = '-'
                    pre_list_data = [t, list_st[t -
1], mark_2]
                else:
                    mark_1 = sheet_class.cell(row=(i +
1 + t), column=j - 1 + y).value
                    if mark_1 == None:
                        mark_1 = '-'
                    mark_2 = sheet_class.cell(row=(i +
1 + t), column=j).value
                    if mark_2 == None:
                        mark_2 = '-'
                    pre_list_data = [t, list_st[t -
1], mark_1, mark_2]
                list_data.append(pre_list_data)
                t = t + 1
            break
        j = j + 1
    break
```

```

else:
    if i > l_row_c:
        self.create_new_subject_class(sbj, class_name,
i-3)

        j = 1
        while j <= value_stud:
            pre_list_data = [j, list_st[j-1], "-"]
            list_data.append(pre_list_data)
            j = j + 1
        break
    i = i + value_stud + 4
list_data.append(time)

```

На рисунку 3.15 над рядком дат продемонстровані мітки.

Математика	stun:	T	T	T	T	A	T	T	T	
Active	#	Im'я/Date:	22:05:2025	27:05:2025	29:05:2025	03:06:2025	AVERAGE	05:06:2025	10:06:2025	12:06:2025
4	1	Д. Аріна	8	8	1 m	5,666667	m	m		10
5	2	З. Андрій	8 m		2	10	6,666667		11	2
4	3	З. Дмитро	2 m	m		10	6	6 m		11
6	4	З. Артем	10 m		5	6	7	10	5	6
5	5	З. Роман		10	2	10	8,666667	0		0

Рисунок 3.15 – Демонстрація системи міток для оцінки стану заняття

Мітки можуть відповідати таким станам:

- T – урок закінчено;
- F – урок не закінчено при запуску програми можливо продовжити редагувати це заняття;
- A, S – спеціальні мітки для позначення стовпця з підсумками.

Функції «start_f» та «start_save» відповідають за ініціалізацію деяких об'єктів та за виведення інформації на них. «start_f» функція першого запуску тобто використовується для виведення інформації з нових занять, тоді як «start_save» зберігає всю інформацію для збереження теперішнього стану програми.

Розглянемо код лише функції «start_f» тому, що код функцій подібний та немає сильних змін. Наведений код функції відповідає за виведення інформації про найближче заняття якщо воно існує. В іншому випадку код викликає ініціалізацію функції створення незапланованого уроку.

```

def start_f(self):
    self.create_info()
    if Assistant.boll_1 == True:
        list_c_s_st =
self.now_lesson_and_next(Assistant.basic_inf[3],
Assistant.basic_inf[2])
        Assistant.save_l_n = list_c_s_st
        Assistant.boll_1 = False
    else:
        list_c_s_st = Assistant.save_l_n

    if len(list_c_s_st) == 1:
        self.label_info_1.configure(text=list_c_s_st)
        self.create_interactive(1)
        if Assistant.var_c == 1:
            self.label_info_2.configure(text="Ви можете
створита не запланованне заняття! \nДля цього оберіть клас з
таблиці, та натисніть підтвердити!")
            self.table_info()
        else:
            self.label_info_2.configure(text="Оберіть предмет
з таблиці, та натисніть створити!")
            self.label_info_3.configure(text=f"Ви обрали клас:
{Assistant.value_r[0]}\nОбраний предмет: ")
            self.table_info()
        else:
            self.create_interactive(0)
            self.create_info()

            Assistant.value_now = [list_c_s_st[0][2],
list_c_s_st[0][3]]
            Assistant.value_r = list_c_s_st

            self.label_info_1.configure(text=f"Найблище заняття:
Заняття №{list_c_s_st[0][1]}, о {list_c_s_st[0][0]}; \nКлас:
{list_c_s_st[0][2]}; \nПредмет: {list_c_s_st[0][3]}")
            if len(list_c_s_st[1]) == 1:
                self.label_info_2.configure(text=f'Це останнє
заняття на сьогодні.')
            else:
                self.label_info_2.configure(text=f"Наступне
заняття: Заняття №{list_c_s_st[1][1]} о {list_c_s_st[1][0]}
\nКлас: {list_c_s_st[1][2]} \nПредмет: {list_c_s_st[1][3]}")

            list_t_t =
self.text_table_input_l_n(list_c_s_st[0][2], list_c_s_st[0][3])
            self.label_info_3.configure(text=f"Номер:
{list_t_t[0][0]} \n Ім'я: {list_t_t[0][1]}")
            Assistant.value = [list_t_t[0][0], list_t_t[0][1]]

```

```
l_list = len(list_t_t) - 1
heddd = self.text_colum(list_t_t)
self.table_lable(list_t_t[0:l_list], heddd)
```

На рисунку 3.16 демонструється роботи функції «start_f» (виведення інформація за поточне та майбутнє заняття і інформації про відмітки з минулого та теперішнього заняття).

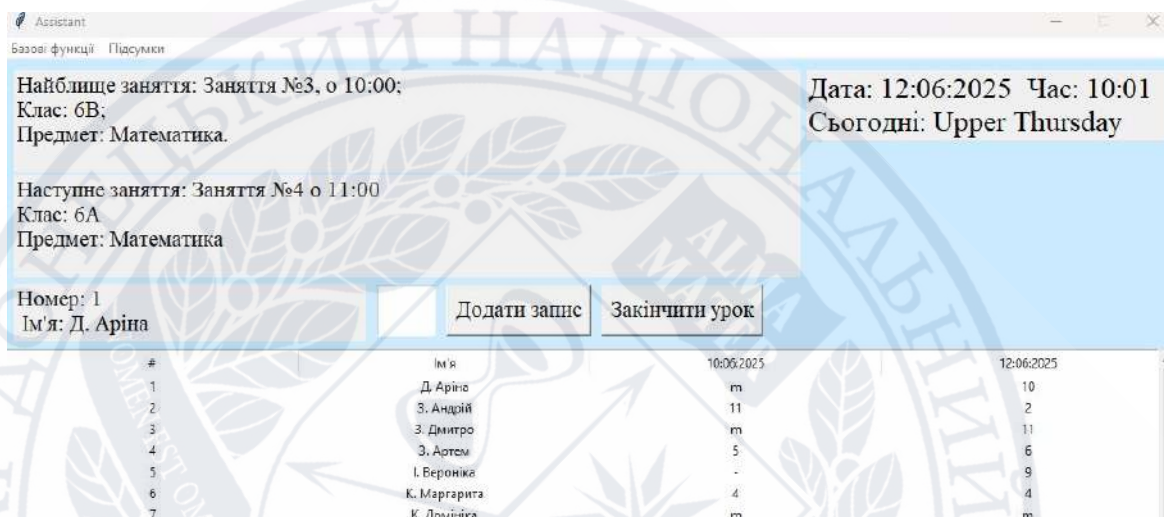


Рисунок 3.16 – Демонстрація роботи функції «start_f»

3.6 Функції реакції та тестування роботи програми

Функція «create_result» відповідає за підведення підсумків у журналі та виведення відповідної інформації про виконання дії.

```
def create_result(self, v):
    try:
        class_name = Assistant.value_now[0]
        sbj_name = Assistant.value_now[1]
    except IndexError :
        self.label_info_3.configure(text=f"Ви не обрали клас  
чи предмет!")
        return None
    value_stud = self.get_value_stud(class_name)
    i_and_j = self.find_space_new_mark(class_name, sbj_name,
    value_stud)
```

```

i = i_and_j[0]
j = i_and_j[1] + 1
sheet_class = Assistant.work_book[class_name]
text_a = ["AVERAGE all", "SUM all", "AVERAGE", "SUM"]
sheet_class.cell(row=i + 1, column=j, value=text_a[v])
if v == 0 or v == 2:
    text_a = 'A'
else:
    text_a = 'S'
sheet_class.cell(row=i, column=j, value=text_a)
dubl = self.lit_colum(i)
if v == 0:
    str_f = '=IFERROR(AVERAGE('
    a = 0
elif v == 1:
    str_f = '=IFERROR(SUM('
    a = 0
elif v == 2:
    str_f = '=IFERROR(AVERAGE('
    a = 1
elif v == 3:
    str_f = '=IFERROR(SUM('
    a = 1
self.cr_ex_f(i, j, value_stud, dubl, str_f, a)
text_a = ["Було підведено підсумок \nзар. AVERAGE", "Було
підведено підсумок \nзар. SUM", "Було підведено підсумок AVERAGE",
"Було підведено підсумок SUM"]
self.label_info_3.configure(text=text_a[v])
self.seve_wb()

```

Опис всіх розглянутих функцій у третьому розділі наведено в повному лістингу (Додаток А).

На рисунку 3.17 продемонстровано активацію функції «create_result».

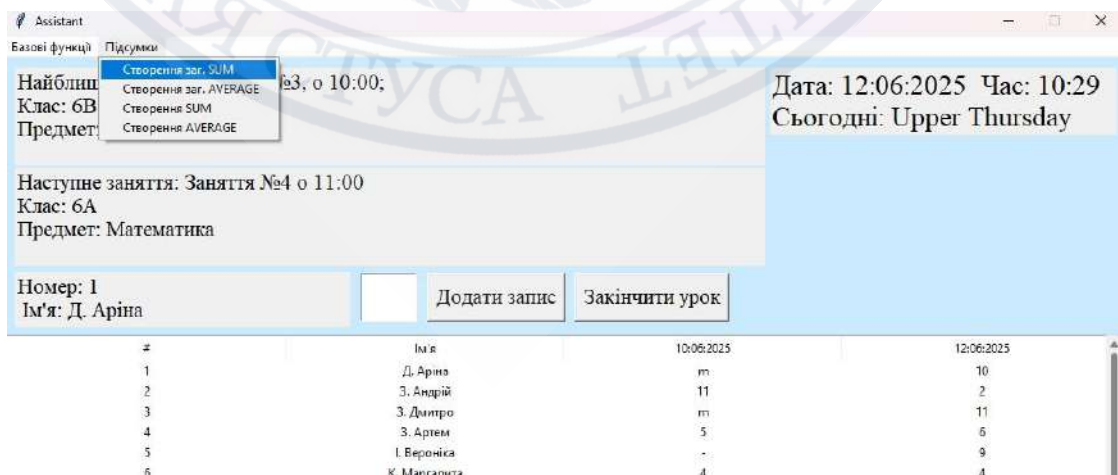


Рисунок 3.17 – Демонстрація активації підведення підсумку

На рисунку 3.18 продемонстровано результат роботи функції «create_result».

Active	#	Im'я/Date:	22:05:2025	27:05:2025	29:05:2025	03:05:2025	AVERAGE	05:06:2025	10:06:2025	12:06:2025	SUM all
4	1	Д. Аріна	8	8	1	m	5,665667	m	m	10	0
5	2	З. Андрій	8	m	2	10	6,665667		11	2	33
4	3	З. Дмитро	2	m	m	10	6	6	m	11	29
6	4	З. Артем	10	m	5	6	7	10	5	6	42
5	5	І. Вероніка		12	2	12	8,665667	9		9	44
6	6	К. Маргарита	4		12	10	8,665667	2	4	4	36
4	7	К. Домініка	6	5		2	4,333333	4	m	m	17
6	8	К. Максим	7	10	4	6	6,75	1	7	m	35
5	9	М. Артем	5	m		11	8	2	12	4	34
4	10	П. Олександра		m	5	9	7	1	6		21
4	11	П. Костянтин	m	12	4		8		5	10	31
3	12	П. Кіра	1		m	10	5,5		m	6	17
5	13	П. Назар	3	1	3	9	4	m	1	m	17
5	14	П. Єлизавета		6	8	3	5,665667	m	9	6	32

Рисунок 3.18 – Демонстрація результату підведення підсумку у журналу

Функції «open_file» та «open_file_c» відповідають за відкриття файлу на робочому столі. «open_file» відкриває головну робочу книгу, що блокує роботу програми, а «open_file_c» відкриває копію робочої книги для перегляду журналу та роботи з програмою.

Продемонструємо код цих функцій:

```
def open_file_c(self):
    Assistant.work_book.save(f'./data/copy_Wbook.xlsx')
    os.startfile("data\\copy_Wbook.xlsx")

def open_file(self, name_file):
    os.startfile("data\\" + name_file)
```

Функції «click», «create_lesson» та «end_leson»:

- «click» дозволяє обирати елемент з таблиці;
- «create_lesson» ініціалізує створення нового заняття;
- «end_leson» закінчує заняття та переходить до наступного заняття

Продемонструємо код цих функцій:

```

def click(self, event):
    try:
        t = self.table.selection()
        if len(self.table.item(t) ["values"]) == 1:

            t2 = self.table.item(t) ["values"] [0]
            Assistant.value = [t2]
        else:
            t2 = self.table.item(t) ["values"] [0]
            t3 = self.table.item(t) ["values"] [1]
            Assistant.value = [t2, t3]
    except IndexError:
        None
    self.change_label_info()

def create_lesson(self):
    Assistant.var_c = 1
    Assistant.continue_t_ln = Assistant.save_l_n

    Assistant.save_l_n = ['Створення нового завдання']
    self.reload(0)

def end_leson(self):
    sheet_class = Assistant.work_book[Assistant.value_now[0]]
    value_stud = self.get_value_stud(Assistant.value_now[0])
    i_and_j = self.find_space_new_mark(Assistant.value_now[0],
    Assistant.value_now[1], value_stud)
    i = i_and_j[0]
    j = i_and_j[1]
    sheet_class.cell(row=i, column=j, value="T")
    if sheet_class.cell(row=i+1, column=j).value == None:
        sheet_class.cell(row=i+1, column=j,
value=strftime('%d:%m:%Y'))
    self.create_or_update_Active(i , j, value_stud)
    Assistant.boll_1 = True
    if Assistant.continue_t_ln != []:
        Assistant.save_l_n = [Assistant.continue_t_ln[1],
Assistant.continue_t_ln[0]]
        Assistant.continue_t_ln = []
    if Assistant.save_l_n[1] == ["Кінець"]:
        Assistant.var_c = 1
        Assistant.save_l_n = ["Сьогодні були вже проведені усі
заняння!"]
        Assistant.boll_1 = False
    self.reload(0)

```

На рисунку 3.19 продемонстрований запуск інтерактивного додатку «Асистент».

Assistant
Базові функції Підсумки

Найближче заняття: Заняття №4, о 11:00;
Клас: 6А;
Предмет: Математика.

Дата: 12:06:2025 Час: 10:57
Сьогодні: Upper Thursday

Наступне заняття: Заняття №6 о 13:00
Клас: 6В
Предмет: Інформатика

Номер: 1
Ім'я: А. Аліна

#	Ім'я	12:06:2025
1	А. Аліна	-
2	Б. Артем	-
3	Б. Богдан	-
4	Б. Владислав	-
5	Б. Олексій	-
6	Б. Олександра	-
7	В. Марія	-
8	В. Ахмед	-
9	Д. Арієн	-
10	Д. Аліна	-

Рисунок 3.19 – Запуску програми

На рисунку 3.20 продемонстрована робота функції додавання нових міток до журналу.

Assistant
Базові функції Підсумки

Найближче заняття: Заняття №4, о 11:00;
Клас: 6А;
Предмет: Математика.

Дата: 12:06:2025 Час: 10:58
Сьогодні: Upper Thursday

Наступне заняття: Заняття №6 о 13:00
Клас: 6В
Предмет: Інформатика

Номер: 2
Ім'я: Б. Артем

#	Ім'я	12:06:2025
1	А. Аліна	10
2	Б. Артем	-
3	Б. Богдан	11
4	Б. Владислав	-
5	Б. Олексій	12
6	Б. Олександра	9
7	В. Марія	8
8	В. Ахмед	-

Рисунок 3.20 – Додавання нових міток

На рисунку 3.21 зображений перехід до наступного заняття після виконання функції «end_leson».

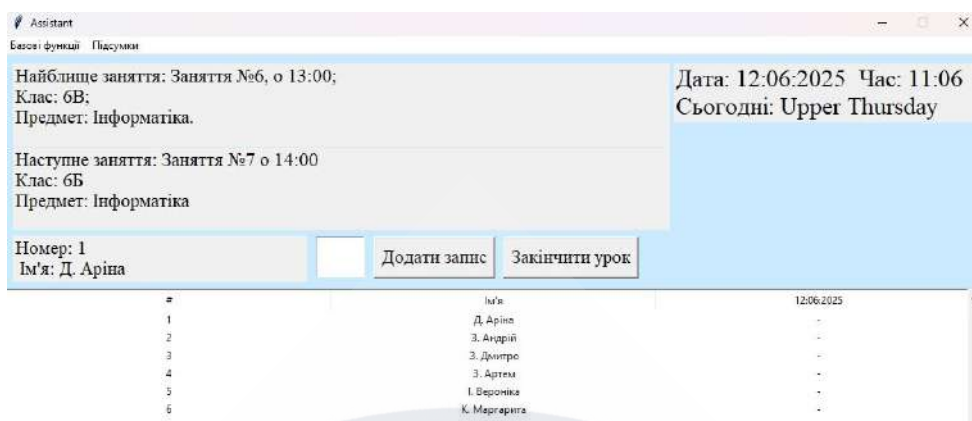


Рисунок 3.21– Перехід до нового заняття в наслідок функції «end_leson».

На рисунках 3.22 продемонстрована функція створення не запланованого заняття.

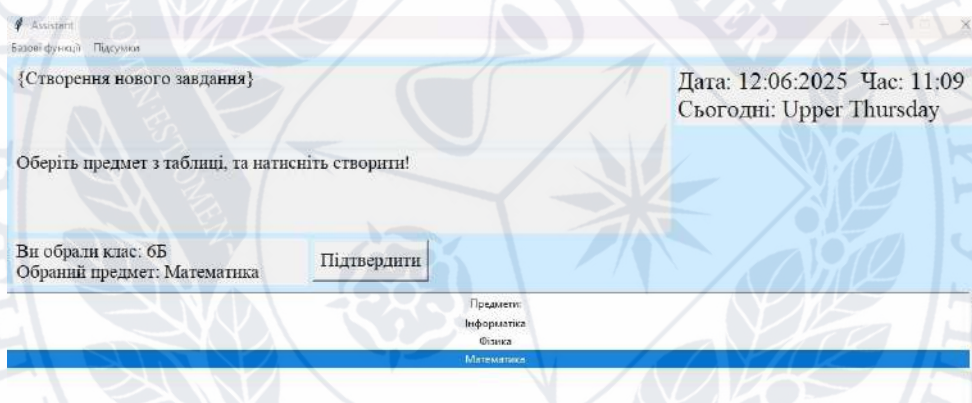


Рисунок 3.22 – Інтерфейс вибору класу та предмету

На рисунку 3.23 продемонстровано створене незаплановане заняття.

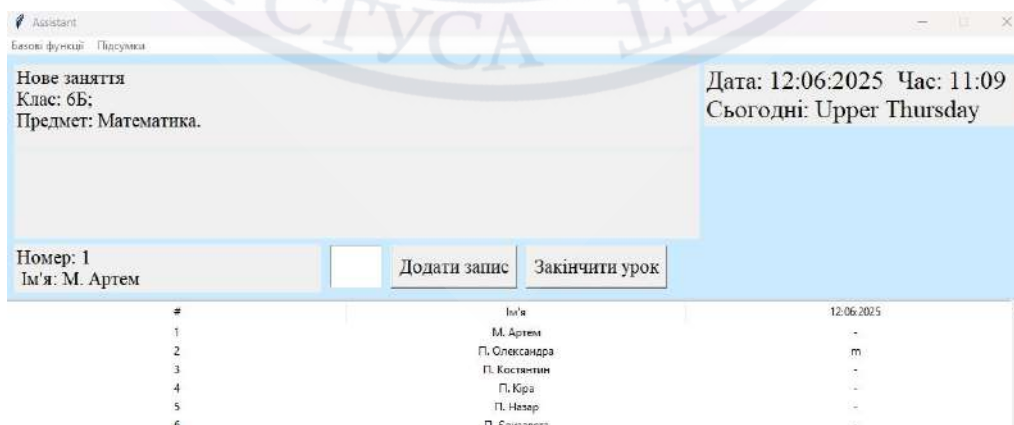


Рисунок 3.23 – Інтерфейс нового заняття

На рисунку 3.24 наведено результати проведення декількох занять та підведення локальних підсумків.

	B	C	D	E	F	G	H	I
17		15	Р. Ангеліна					
18								
19								
20	Математика	stun:		T	T	T	F	A
21	Active	#	Ім'я/Date:	22:05:202	27:05:202	29:05:202	03:06:202	AVERAGE
22		3	1 Д. Аріна	8	8	1 m		5,66667
23		3	2 З. Андрій	8 m		2	10	6,66667
24		2	3 З. Дмитро	2 m	m		10	6
25		3	4 З. Артем	10 m		5	6	7
26		3	5 І. Вероніка		12	2	12	8,66667
27		3	6 К. Маргарита	4		12	10	8,66667
28		3	7 К. Домініка	6	5		2	4,33333
29		4	8 К. Максим	7	10	4	6	6,75
30		2	9 М. Артем	5 m			11	8

Рисунок 3.24 – Журнал предмету

У цьому розділі розглянуто всі основні та частина другорядних функцій, що необхідні для реалізації інтерактивного додатку «Асистент». Ефективність роботи програми підтверджується коректністю та якістю виконання поставлених задач. До того ж цей додаток є простим у використанні та має стабільну систему, що дозволяє йому запускатись на більшості персональних девайсів.

ВИСНОВКИ

У бакалаврській роботі проаналізовано актуальні проблеми, з якими стикаються педагогічні працівники в сучасних умовах. Зокрема, виокремлено такі ключові виклики, як недостатнє матеріально-технічне забезпечення, відсутність постійного робочого кабінету та відсутність стабільного доступу до мережі Інтернет внаслідок впливу різних чинників. Ці фактори суттєво ускладнюють освітній процес, знижують ефективність роботи вчителя та негативно впливають на якість навчання учнів.

Для вирішення цих проблем запропоновано інноваційне рішення – розробка інтерактивного додатку. Головною перевагою цього додатку є його здатність працювати без постійного підключення до мережі Інтернет та необхідності встановлення Excel, що є критично важливим в умовах обмеженого доступу до глобальної мережі в період військового стану. Крім того, під час його розробки особлива увага приділялася низькій ресурсозатратності, що забезпечує можливість його використання на різних пристроях.

Додаток реалізований за допомогою мови програмування Python з використанням бібліотек Tkinter та Openpyxl. На основі аналізу переваг і недоліків цієї мови програмування, визначено, що вона найкраще підходить для реалізації запланованого функціоналу інтерактивного додатку «Асистент». Врахувавши можливості електронної книги Microsoft Office Excel, а саме можливість подальшої обробки зібраних даних, демонстрацію функціоналу розробленого додатку ефективно представлено через інтерфейс Excel.

У роботі проаналізовано та систематизовано список функцій, які були використані під час реалізації інтерактивного додатку, а саме функції для візуального відображення програми, функції що відповідають за роботу із часом, функції що відповідають за обробку інформації, а також функції, що реалізують події ініціалізовані користувачем.

Розроблений додаток є багатофункціональним інструментом, що по суті є аналогом електронного журналу. Він дозволяє педагогу в режимі реального часу відслідковувати ще не проведені заняття, додавати заняття, що не були

заплановані за розкладом, керувати списком учнів а також виставляти та редагувати оцінки. Важливою функцією є можливість автоматичного розрахунку підсумкових та тематичних оцінок, що значно економить час вчителя. Додатково, передбачено можливість відслідковувати учнів за рівнем активності, що допомагає педагогу краще відстежувати прогрес та залученість кожного учня.

Особливу увагу в роботі було приділено питанню конфіденційності результатів оцінювання учнів та особливості їх оцінювання як методу контролю успішності навчального процесу. Впровадження електронних систем вимагає надійних механізмів захисту даних, і в межах розробленого додатку були враховані принципи забезпечення конфіденційності та цілісності інформації.

Таким чином, створений інтерактивний додаток «Асистент» є ефективним інструментом, здатним значно полегшити роботу педагогічних працівників, оптимізувати вторинні процеси, що опосередковано впливають на завантаженість працівників освіти, а також надає можливість працювати за будь-яких умов що значно покращує ефективність та якість освітнього процесу.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Bondarenko H. Ukrainian education in wartime: challenges and problems. Вісник ХНУ імені В. Н. Каразіна. 2022. № 62. С. 141-158.
2. Javed S., Javed S., Khan A. Effect of Education on Quality of Life and Well Being. The International Journal of Indian Psychology. 2016. № 3. P. 1-10
3. Marowka A. On parallel software engineering education using python. Education and Information Technologies. 2018. №23. P. 357-372
4. Srinath K.R. Python –The Fastest Growing Programming Language. International Research Journal of Engineering and Technology. 2017. № 4. P. 354–357.
5. The Python Standard Library. URL: <https://docs.python.org/uk/3.13/library/index.html> (дата звернення: 20. 05. 2025)
6. Tkinter — Python interface to Tcl/Tk.
URL: <https://docs.python.org/uk/3.13/library/tkinter.html#architecture> (дата звернення: 20. 05. 2025)
7. Openpyxl documentation. URL: <https://openpyxl.readthedocs.io/en/stable/> (дата звернення: 20. 05. 2025)
8. Openpyxl.utils.cell module documentation.
URL: <https://openpyxl.readthedocs.io/en/3.1/api/openpyxl.utils.cell.html> (дата звернення: 20. 05. 2025)
9. Бак С., Ковтонюк Г., Крупський Я. Об'єктно-орієнтований підхід до створення графічного інтерфейсу користувача в Python із використанням модуля Tkinter. Теорія і методика професійної освіти. 2025. № 1. С. 169-175
10. Бак С.М., Ковтонюк Г.М. Особливості створення графічного інтерфейсу користувача під час вивчення програмування мовою Python Майбутніми Вчителями математики. Сучасні інформаційні технології та інноваційні методики навчання у підготовці фахівців: методологія, теорія, досвід, проблеми. 2021. № 60. С. 143–157.
11. Брижко В. М., Пилипчук В. Г. Приватність, конфіденційність та безпека персональних даних. Інформація і право. 2020. №1 (32). С. 33-46.

12. Openpyxl.utils.cell module documentation.
URL: <https://docs.python.org/3/library/os.html> (дата звернення: 02. 06. 2025)
13. Використання цифрових технологій у проєктуванні навчального середовища початкової школи. Розвиток сучасної науки та освіти : реалії, проблеми якості, інновації. Матеріали IV Міжнародна науково-практична інтернет-конференція (29-31 травня 2023, Запоріжжя). Запоріжжя : ТДАТУ, 2023. 426-429 с.
14. Голяд І., Кільдерова Л., Голяд Р. Етичні аспекти моніторингу та оцінювання навчальних досягнень здобувачів освіти з особливими освітніми потребами. The driving force of science and trends in its development: VI International Scientific and Theoretical Conference. (Coventry, 21 June. 2024) Coventry, 2024. P. 112-117.
15. Голяд І. С., Динько В. А., Тропіна М. А. Особливості моніторингу навчальних досягнень учнів з трудового навчання та технології у закладах загальної середньої освіти. Молодь і ринок. 2022. № 5 (203). С. 81-86.
16. Гусак Л.П., Радзіховська Л.М. Використання табличного процесора MS EXCEL у викладання дисциплін професійного спрямування в економічних ЗВО. Сучасні інформаційні технології та інноваційні методики навчання в підготовці фахівців: методологія, теорія, досвід, проблеми. 2019. №54. 136-139 с.
17. Гривко А. В. Проблеми оцінювання навчальних досягнень учнів в умовах дистанційного навчання в період пандемії. XXIII International Scientific and Practical Conference: Theoretical and Practical Foundations of Social Process Management(San Francisco, 29-30 June). San Francisco, 2020. P. 182-184.
18. Грудинін Б. О. Готовність майбутніх фахівців до використання інновацій у педагогічній діяльності. Збірник наукових праць Кам'янець-Подільського національного університету імені Івана Огієнка. Серія : Педагогічна. – 2014. – Вип. 20. 78-81 с.
19. Дяденчук А. Ф. Інформаційні технології як засіб вдосконалення науково-дослідницької діяльності майбутніх інженерів. Удосконалення освітньо виховного процесу в закладі вищої освіти. Збірник науково методичних праць /

Таврійський державний агротехнологічний університет імені Дмитра Моторного. Мелітополь: ТДАТУ, 2022. Вип. 25. 248-255 с.

20. Електронний журнал обліку успішності слухачів (курсантів, студентів) як засіб раціоналізації навчально-виховного процесу. О. В. Діденко, Д. А. Купрієнко Інформаційні технології і засоби навчання. 2015. Т. 47, вип. 3. 110-123 с.

21. Закон України «Про інноваційну діяльність» URL: <http://zakon4.rada.gov.ua/laws/show/40-15>. (останнє звернення: 20. 05. 2025)

22. Кирикович В. Д., Козяр В. О., Гіроль А. М. Незалежне оцінювання знань здобувачів вищої освіти. Стан та перспективи впровадження в Національному університеті водного господарства та природокористування. Оновлення змісту, форм та методів навчання і виховання в закладах освіти. 2017. № 16. С. 18–20.

23. Ковтун О. В., Сидоренко С. І. Незалежне оцінювання якості освітніх послуг здобувачами вищої освіти: досвід та перспективи. Вісник Національного авіаційного університету. Серія: Педагогіка. Психологія. 2018. № 1(12). С. 52–60.

24. Козлов В. В., Суперсон І. І., Ткаченко О. А. Моделювання розрахунків у сфері цінних паперів засобами MS EXCEL. Науковий вісник Національної академії статистики, обліку та аудиту. 2014. № 2. 58-68 с.

25. Литвинова С., Мельник О., Сухих А. Електронний журнал як інструмент цифрової освітньої комунікації: результати всеукраїнського опитування. Інформаційні технології і засоби навчання. 2024. №2. С. 141-161

26. Литвинчук А.О., Гапон В. В., Пронь Н. Б., Барабаш О. А., Читаєва К. Г. Аналіз фінансового забезпечення закладів вищої освіти в умовах збройної агресії в Україні. Освітня аналітика України. 2024. № 6. С. 63-80.

27. Лукашенко Л.В. Фінансування закладів загальної середньої освіти України: виклики крізь призму війни. 2025 № 7 URL: <https://doi.org/10.5281/zenodo.14874948> (останнє звернення: 25. 05. 2025)

28. Меньяйленко О. С., Монастирна Г.В. Сучасні інформаційні засоби в педагогічних технологіях . Вісник Луганського національного університету ім. Т. Шевченка. Педагогічні науки. 2011. №21 (232). С. 134 – 142.
29. Мізюк В., Дмитрієва М., Абросімов Є., Поволяшко К. Про один із способів використання інструментів MS Excel для автоматизації ресурсзатратних задач. Наукові інновації та передові технології. 2023. № 6 (20). 518-527 с.
30. Ніксон Ф. Роль керівництва підприємництва в забезпеченні якості та надійності: пер. з англ. / Ф. Ніксон. – М. : Видавництво стандартів, 1990. 210 с.
31. Олексенко К. Розвиток освітнього простору майбутніх учителів до проектування освітньо-розвивального середовища здобувачів початкової освіти Гуманітарний вісник Запорізької державної інженерної академії. 2019. 126-128 с.
32. Освіта в умовах змін, інновацій та викликів. Міжнародна науково-методична конференція «Університетська наука і освіта: традиції та інновації». (13-14 травня 2021, Харків) Харків. 77-78 с.
33. Осередчук О. А. Зовнішній та внутрішній моніторинг якості освіти. Наукові записки. Серія: Педагогічні науки. 2001. С. 112-116.
34. Соловйов В. П., Г. І. Кореняко, В. М. Головатюк Інноваційний розвиток регіонів: питання теорії та практики : Монографія. Фенікс, 2008. 224 с.
35. Тарасова Т. Б., Баранов Б. М. Електронний щоденник як засіб реалізації сучасних підходів до контролю навчальної діяльності учнів. Автоматизація та комп'ютерно-інтегровані технології у виробництві та освіті: стан, досягнення, перспективи розвитку: Всеукраїнська науково-практична інтернет-конференція. (Черкаси, 13-19 березня 2017). Черкаси, 2017. С. 121-123.
36. Тоцька О. Л., Титаренко І. О. Фінансування освіти України з бюджетів різних рівнів в умовах війни. Освітня аналітика України. 2023. №3 (24). С. 34-42.
37. Формування професіональної ІТ-компетентності як основа реформування сучасної української школи. Taptautinė mokslinė – praktinė konferencija (2018 m. gegužės 11–12 d., Marijampolė), Kaunas. 2019. 126–27 с.

38. Шмирова О. В. Використання сучасних інформаційних технологій при викладанні іноземних мов. Наукові записки Національного університету Острозька академія. 2009. Вип. 11. С. 535-542.

39. Нові знання. URL: <https://nz.ua/>

40. Вовк О. В., Фефелова З. Є. Важливість прототипування в розробці веб-ресурсів. Матеріали Молодіжної школи-семінару IX Міжнар. Науково-технічна конференція (14-28 травня 2024 р Харків). Харків. 175-177 с.



ДОДАТКИ



ДОДАТОК А

Повний лістинг програми «Асистент»

```

from os.path import split
from time import strftime
import datetime
from openpyxl import load_workbook
from openpyxl.workbook import Workbook
from openpyxl.styles import Font, PatternFill, Alignment
import tkinter as tk
from tkinter import ttk
import os
from openpyxl.utils.cell import get_column_letter

#time6 = datetime.time(7, 35)
#time = time6.strftime('%H:%M')

def get_basic_inf_f():
    file = open('basic_elements.txt', 'r', encoding='utf-8')
    file_list = file.readlines()
    file.close()
    basic_list = []
    for i in file_list:
        j = len(i) - 1
        if i[-1] == '\n':
            i = i[0:j]
        basic_list.append(i)
    return basic_list

def write_base_inf(list):
    file = open('basic_elements.txt', 'w', encoding='utf-8')
    for index in list:
        file.write(index + '\n')
    file.close()

def cal_minute(date):
    time_minute = 0
    date = date.split(':')
    if date[0][0] == '0':
        if date[1][0] == '0':
            time_minute = 60 * int(date[0][1]) + int(date[1][1])
        else:
            time_minute = 60 * int(date[0][1]) + int(date[1])
    else:
        if date[1][0] == '0':
            time_minute = 60 * int(date[0]) + int(date[1][1])
        else:
            time_minute = 60 * int(date[0]) + int(date[1])

```

```

    return time_minute

class Assistant:

    basic_inf = get_basic_inf_f()
    name_ak = basic_inf[0]
    work_book = load_workbook(f'./data/{name_ak}.xlsx')
    h_main = 600
    w_main = 1100

    window = tk.Tk()
    window.title('Assistant')
    window.geometry(f"{w_main}x{h_main}+300+150")
    window.resizable(False, False)

    day_week_now = strftime('%A')
    hour_minute_now = strftime('%H:%M')
    #hour_minute_now = time
    week_s = ''

    continue_t_ln = []
    save_l_n = [[], []]
    value = []
    value_now = []
    value_r = []
    var_c = 0
    boll_1 = True

    def __init__(self):
        self.label_info = tk.Label(Assistant.window, height='10',
width='10', background='#caeaff')
        self.label_table = tk.Label(Assistant.window, height='10',
width='10', background='#cedfea')
        self.label_time = ttk.Label(Assistant.window, font=("Times
New Roman", 20))

        self.s = ttk.Style()
        self.s.configure('Lable.TextL', font = ("Times New
Roman", 16))

        self.table = ttk.Treeview(self.label_table,
show='headings')
        self.scroll_p = ttk.Scrollbar(self.table,
command=self.table.yview())

        self.label_info_1 = tk.Label(Assistant.window,
font=("Times New Roman", 16), justify=tk.LEFT, anchor='nw')
        self.label_info_2 = tk.Label(Assistant.window,
font=("Times New Roman", 16), justify=tk.LEFT, anchor='nw')
        self.label_info_3 = tk.Label(Assistant.window,
font=("Times New Roman", 16), justify=tk.LEFT, anchor='nw')

```

```

self.entry_1 = tk.Entry(Assistant.window)

self.btn_1 = tk.Button(Assistant.window, text="Додати
запис", font=("Times New Roman",16), command=self.add_new_mark_st)
self.btn_2 = tk.Button(Assistant.window, text="Закінчити
урок", font=("Times New Roman",16), command=self.end_leson)

self.btn_3 = tk.Button(Assistant.window,
text="Підтвердити", font=("Times New Roman", 16),
command=self.take_obj)

self.table.bind("<ButtonRelease-1>", self.click) #<Button-
1>

def create_wg(self):
    self.label_info.place(relwidth=1, relheight=0.50, relx=0,
rely=0)
    self.label_table.place(relwidth=1, relheight=0.55, relx=0,
rely=0.45)
    self.label_time.place(x=752, y=10, relwidth=0.31)

    self.table.pack(expand=tk.YES, fill=tk.BOTH)
    self.scroll_p.pack(side="right", fill="y")

def create_info(self):
    self.label_info_1.place(relwidth=0.67, relheight=0.16,
x=10, y=10)
    self.label_info_2.place(relwidth=0.67, relheight=0.16,
x=10, rely=0.18)
    self.label_info_3.place(relwidth=0.3, relheight=0.09,
x=10, rely=0.35)

def create_interactive(self, a):
    if a == 0:
        self.entry_1.place(relwidth=0.05, relheight=0.08,
x=350, rely=0.351)
        self.btn_1.place(relheight=0.08, x=415, rely=0.352)
        self.btn_2.place(relheight=0.08, x=560, rely=0.352)
    else:
        self.btn_3.place(relheight=0.08, x=345, rely=0.353)

def set_menu(self):
    menubar = tk.Menu(self.window)
    self.window.config(menu=menubar)

    set_menu = tk.Menu(menubar, tearoff=0)
    set_menu_test_f = tk.Menu(menubar, tearoff=0)

    set_menu.add_command(label='Демонстраційний
файл',command=lambda:self.open_file_c())
    set_menu.add_command(label='Змінити файл',
command=lambda:self.change_work_book())

```

```

        set_menu.add_command(label='Створити заняття',
command=self.create_lesson)
        set_menu.add_command(label='Вихід',
command=self.window.destroy)
        set_menu.add_command(label='Робоча книга',
command=lambda:self.open_file(self.name_ak))
        menubar.add_cascade(label='Базові функції', menu=set_menu)

        set_menu_test_f.add_command(label='Створення заг. SUM',
command=lambda:self.create_result(1))
        set_menu_test_f.add_command(label='Створення заг.
AVERAGE', command=lambda:self.create_result(0))
        set_menu_test_f.add_command(label='Створення SUM',
command=lambda:self.create_result(3))
        set_menu_test_f.add_command(label='Створення AVERAGE',
command=lambda:self.create_result(2))
        menubar.add_cascade(label='Підсумки',
menu=set_menu_test_f)

def now_week(self, structure_timetable):
    sheet_t = Assistant.work_book['technical_page']
    date_obj = datetime.datetime.now()
    date_obj = int(date_obj.strftime("%V")) % 2
    structure_timetable = structure_timetable.split(' ')
    if date_obj == 0:
        sheet_t.cell(row=3, column=4,
value=structure_timetable[1])
    else:
        sheet_t.cell(row=3, column=4,
value=structure_timetable[0])
    self.seve_wb()

def day_week_l_n(self, i, day_week_now, l_row):
    sheet_time_t = Assistant.work_book['time_table']
    l_week = []
    i_and_j = [i, 0]

    i1 = i + 4
    j = 0
    while i1 < l_row:
        if sheet_time_t.cell(row=i1, column=1).value ==
day_week_now:
            j = sheet_time_t.cell(row=i1, column=6).value
            i_and_j = [i1, j]
            break
        i1 = i1 + 1
    return i_and_j

def now_lesson_and_next(self, structure_timetable, list_time):
    sheet_time_t = Assistant.work_book['time_table']
    sheet_t = Assistant.work_book['technical_page']

```

```

Assistant.week_s = sheet_t.cell(row=3, column=4).value

list_time = list_time.split(' ')
structure_timetable = structure_timetable.split(' ')
l_row = self.len_row('time_table')
tum = 0
t = 0
list_c_s_st = []
minute_now = cal_minute(Assistant.hour_minute_now)

if Assistant.week_s == structure_timetable[0]:
    i = sheet_t.cell(row=3, column=3).value
else:
    if Assistant.week_s == structure_timetable[1]:
        i = sheet_t.cell(row=3, column=2).value
    else:
        i = 2
i = int(i)
i_and_j = self.day_week_l_n(i, Assistant.day_week_now,
l_row)
i = i_and_j[0]
j = i_and_j[1]

if j == 0 :
    list_c_s_st.append("Сьогодні не заплановано жодного
заняття!")
    Assistant.var_c = 1
elif len(Assistant.save_l_n[1]) == 4:
    while t < j:
        if sheet_time_t.cell(row=i + t + 1,
column=3).value == Assistant.save_l_n[1][1]:
            tum = 1
            break
        t = t + 1
    else:
        while t < j:
            nom_lesson = int(sheet_time_t.cell(row=i + t + 1,
column=3).value)
            minute_lesson = cal_minute(list_time[nom_lesson -
1])

            min = 10
            try:
                minute_lesson_s =
cal_minute(list_time[nom_lesson])
            except:
                minute_lesson_s =
cal_minute(list_time[nom_lesson - 1])
                min = - (minute_lesson_s -
cal_minute(list_time[nom_lesson - 2]) - 10)

```

```

        if minute_lesson - 10 < minute_now and minute_now
< minute_lesson_s - min:
            tum = 1
            break
        t = t + 1
    if minute_now > minute_lesson + 50:
        list_c_s_st.append("Сьогодні були вже проведені
усі заняття!")
        Assistant.var_c = 1

    if Assistant.var_c != 1:
        if tum != 1:
            t = 0
            n = 1
            while n < 3:
                list_now_lesson = []
                j = 2
                while j < 6:
                    if n == 2:
                        if sheet_time_t.cell(row=i + n + t,
column=j).value == None:
                            list_now_lesson.append("Кінець")
                            break
                        else:
                            list_now_lesson.append(sheet_time_t.cell(row=i + n + t,
column=j).value)
                    else:
                        list_now_lesson.append(sheet_time_t.cell(row=i + n + t,
column=j).value)
                    j = j + 1
                list_c_s_st.append(list_now_lesson)
                n = n + 1
            return list_c_s_st

    def time_now_d(self):
        time = strftime('Дата: %d:%m:%Y   Час: %H:%M')
        time = time + f"\nСьогодні: {Assistant.week_s}
{Assistant.day_week_now}"
        self.label_time.config(text=time)
        self.label_time.after(60000, self.time_now_d)

    def text_table_input_l_n(self, class_name, sbj ):
        sheet_t = Assistant.work_book['technical_page']
        sheet_class = Assistant.work_book[class_name]
        l_row_c = self.len_row(class_name)
        time = strftime('%d:%m:%Y')
        list_data = []
        value_stud = self.get_value_stud(class_name)
        list_st = self.read_list(3, 4, class_name, value_stud)
        y = 0

```

```

i = value_stud + 5
while True:
    if sheet_class.cell(row=i, column=2).value == subj:
        j = 5
        while True:
            if sheet_class.cell(row=i, column=j).value ==
"F" or sheet_class.cell(row=i, column=j).value == None:
                t = 1
                if sheet_class.cell(row=(i), column=j -
1).value == "A" or sheet_class.cell(row=(i), column=j - 1).value
== "S":
                    y = -1
                    if j != 5:
                        time = sheet_class.cell(row=(i+1),
column=j-1+y).value
                        while t <= value_stud:
                            if j == 5:
                                mark_2 = sheet_class.cell(row=(i +
1 + t), column=j).value
                                if mark_2 == None:
                                    mark_2 = '-'
                                pre_list_data = [t, list_st[t -
1], mark_2]
                            else:
                                mark_1 = sheet_class.cell(row=(i +
1 + t), column=j - 1 + y).value
                                if mark_1 == None:
                                    mark_1 = '-'
                                mark_2 = sheet_class.cell(row=(i +
1 + t), column=j).value
                                if mark_2 == None:
                                    mark_2 = '-'
                                pre_list_data = [t, list_st[t -
1], mark_1, mark_2]
                                list_data.append(pre_list_data)
                                t = t + 1
                                break
                            j = j + 1
                        break
                    else:
                        if i > 1_row_c:
                            self.create_new_subject_class(subj, class_name,
i-3)
                            j = 1
                            while j <= value_stud:
                                pre_list_data = [j, list_st[j-1], "-"]
                                list_data.append(pre_list_data)
                                j = j + 1
                            break
                        i = i + value_stud + 4
                        list_data.append(time)
                        return list_data

```

```

def start_f(self):
    self.create_info()
    if Assistant.boll_1 == True:
        list_c_s_st =
self.now_lesson_and_next(Assistant.basic_inf[3],
Assistant.basic_inf[2])
        Assistant.save_l_n = list_c_s_st
        Assistant.boll_1 = False
    else:
        list_c_s_st = Assistant.save_l_n

    if len(list_c_s_st) == 1:
        self.label_info_1.configure(text=list_c_s_st)
        self.create_interactive(1)
        if Assistant.var_c == 1:
            self.label_info_2.configure(text="Ви можете
створита не запланованне заняття! \nДля цього оберіть клас з
таблиці, та натисніть підтвердити!")
            self.table_info()
        else:
            self.label_info_2.configure(text="Оберіть предмет
з таблиці, та натисніть створити!")
            self.label_info_3.configure(text=f"Ви обрали клас:
{Assistant.value_r[0]}\nОбраний предмет: ")
            self.table_info()
        else:
            self.create_interactive(0)
            self.create_info()

            Assistant.value_now = [list_c_s_st[0][2],
list_c_s_st[0][3]]
            Assistant.value_r = list_c_s_st

            self.label_info_1.configure(text=f"Найблище заняття:
Заняття №{list_c_s_st[0][1]}, о {list_c_s_st[0][0]}; \nКлас:
{list_c_s_st[0][2]}; \nПредмет: {list_c_s_st[0][3]}")
            if len(list_c_s_st[1]) == 1:
                self.label_info_2.configure(text=f'Це останнє
заняття на сьогодні.')
            else:
                self.label_info_2.configure(text=f"Наступне
заняття: Заняття №{list_c_s_st[1][1]} о {list_c_s_st[1][0]}
\nКлас: {list_c_s_st[1][2]} \nПредмет: {list_c_s_st[1][3]}")

            list_t_t =
self.text_table_input_l_n(list_c_s_st[0][2], list_c_s_st[0][3])
            self.label_info_3.configure(text=f"Номер:
{list_t_t[0][0]} \n Ім'я: {list_t_t[0][1]}")
            Assistant.value = [list_t_t[0][0], list_t_t[0][1]]

            l_list = len(list_t_t) - 1

```

```

heddd = self.text_colum(list_t_t)
self.table_lable(list_t_t[0:l_list], heddd)

def start_save(self):
    self.create_info()
    self.create_interactive(0)
    list_c_s_st = Assistant.save_l_n

    if len(list_c_s_st[0]) != 4:
        Assistant.value_now = Assistant.value_r
        self.label_info_1.configure(text=f"Нове заняття\nКлас:
{Assistant.value_now[0]}; \nПредмет: {Assistant.value_now[1]}")

        list_t_t =
self.text_table_input_l_n(Assistant.value_now[0],
Assistant.value_now[1])
        Assistant.value = [list_t_t[0][0], list_t_t[0][1]]
        self.label_info_3.configure(text=f"Номер:
{Assistant.value[0]} \n Ім'я: {Assistant.value[1]}")

        l_list = len(list_t_t) - 1
        heddd = self.text_colum(list_t_t)
        self.table_lable(list_t_t[0:l_list], heddd)

    else:
        self.label_info_1.configure(
            text=f"Найблище заняття: Заняття
№{list_c_s_st[0][1]}, о {list_c_s_st[0][0]}; \nКлас:
{list_c_s_st[0][2]}; \nПредмет: {list_c_s_st[0][3]}")
        if len(list_c_s_st[1]) == 1:
            self.label_info_2.configure(text=f'Це останне
заняття на сьогодні.')
        else:
            self.label_info_2.configure(
                text=f"Наступне заняття: Заняття
№{list_c_s_st[1][1]} о {list_c_s_st[1][0]} \nКлас:
{list_c_s_st[1][2]} \nПредмет: {list_c_s_st[1][3]}")

            list_t_t =
self.text_table_input_l_n(list_c_s_st[0][2], list_c_s_st[0][3])
            self.label_info_3.configure(text=f"Номер:
{Assistant.value[0]} \n Ім'я: {Assistant.value[1]}")

            l_list = len(list_t_t) - 1
            heddd = self.text_colum(list_t_t)
            self.table_lable(list_t_t[0:l_list], heddd)

def start_pr(self, a):
    self.create_wg()
    self.set_menu()

```

```

self.now_week(Assistant.basic_inf[3])
if a == 0:
    self.start_f()
else:
    self.start_save()
self.time_now_d()
self.window.mainloop()

def reload(self, a):
    [child.destroy() for child in
self.window.winfo_children()]
    self.__init__()

    if a == 0:
        self.start_pr(0)
    elif a == 1:
        self.start_pr(1)
    else:
        None

def add_new_mark_st(self):
    value_e = self.entry_1.get()
    try:
        value_e = int(value_e)
    except ValueError:
        None
    value_n = Assistant.value_now
    Assistant.value.append(value_e)

    value = Assistant.value

    if len(value) == 4:
        value = [value[0], value[1], value[3]]
        Assistant.value = value

    sheet_class = Assistant.work_book[value_n[0]]
    value_stud = self.get_value_stud(value_n[0])
    i_and_j = self.find_space_new_mark(value_n[0], value_n[1],
value_stud)
    i = i_and_j[0]
    j = i_and_j[1]

    sheet_class.cell(row=i, column=j, value="F")
    sheet_class.cell(row=i+1, column=j,
value=strftime('%d:%m:%Y'))
    t = 1
    while t <= value_stud:
        if sheet_class.cell(row=i+1+t, column=3).value ==
value[0]:
            sheet_class.cell(row=i + 1 + t, column=j,
value=value[2])

```

```

        break
    t = t + 1

    self.seve_wb()
    self.reload(1)

def len_row(self, name_sheet):
    sheet = Assistant.work_book[name_sheet]
    return len(list(sheet.rows))

def read_list(self, row, column, name_sheet, value_obj):
    sheet_class = Assistant.work_book[name_sheet]
    list = []
    i = 0
    while i < value_obj:
        list.append(sheet_class.cell(row=row + i,
column=column).value)
        i = i + 1
    return list

def change_label_info(self):
    if Assistant.var_c == 1:
        try:
            self.label_info_3.configure(text=f"Ви обрали клас:
{Assistant.value[0]}")
        except IndexError:
            None
        elif Assistant.var_c == 2:
            self.label_info_3.configure(text=f"Ви обрали клас:
{Assistant.value_r[0]}\nОбраний предмет: {Assistant.value[0]}")
        else:
            self.label_info_3.configure(text=f"Номер:
{Assistant.value[0]} \n Ім'я: {Assistant.value[1]}")

def get_value_stud(self, class_name):
    sheet_t = Assistant.work_book["technical_page"]
    l_row_t = self.len_row("technical_page")
    i = 2
    while i <= l_row_t:
        if sheet_t.cell(row=i, column=1).value == class_name:
            break
        i = i + 1

    value_stud = sheet_t.cell(row=i, column=2).value
    return value_stud

def find_space_new_mark(self, class_name, name_sbj,
value_stud):
    sheet_class = Assistant.work_book[class_name]
    i = value_stud + 5
    while True:
        if sheet_class.cell(row=i, column=2).value ==
name_sbj:

```

```

        j = 5
        while True:
            if sheet_class.cell(row=i, column=j).value ==
"F" or sheet_class.cell(row=i, column=j).value == None:
                break
            j = j + 1
        break
        i = i + value_stud + 4
    i_and_j = [i, j]
    return i_and_j

def seve_wb(self):
    Assistant.work_book.save(f'./data/{Assistant.name_ak}')

def table_table(self, list_t_t, heddd):
    len_heddd = len(heddd)
    if len_heddd == 4 and heddd[2] == heddd[3]:
        heddd[2] = heddd[2] + ' '
        self.table['columns'] = heddd

    if len_heddd > 1:
        for header in heddd:
            self.table.heading(header, text=header,
anchor="center")
            self.table.column(header, anchor='center')
        else:
            self.table.heading(heddd, text=heddd[0],
anchor="center")
            self.table.column(heddd, anchor='center')
        for row in list_t_t:
            self.table.insert('', 'end', values=row)

    self.scroll_p = ttk.Scrollbar(self.label_table,
command=self.table.yview())
    self.table.configure(yscrollcommand=self.scroll_p.set)

def create_new_subject_class(self, name_subject, name_class,
row):
    value_stud = self.get_value_stud(name_class)
    sheet_class = Assistant.work_book[name_class]
    sheet_class.cell(row=row + 3, column=2, value=name_subject)
    sheet_class.cell(row=row + 4, column=2, value='Active')
    sheet_class.cell(row=row + 4, column=3, value='#')
    sheet_class.cell(row=row + 4, column=4,
value="Имя/Date:")
    sheet_class.cell(row=row + 3, column=4, value="стун:")
    list_st = self.read_list(3, 4, name_class, value_stud)

    i = self.len_row(name_class) + 1
    v_st = len(list_st) + i
    n_st = 0

    while i < v_st:

```

```

        sheet_class.cell(row=i, column=3, value=n_st + 1)
        sheet_class.cell(row=i, column=4, value=list_st[n_st])
        n_st = n_st + 1
        i = i + 1
    self.seve_wb()

def table_info(self):
    if Assistant.var_c == 1:
        heddd = ['Класи:']
        list_t = Assistant.basic_inf[5]
        list_t = list_t.split("|")
        self.table_lable(list_t, heddd)
    elif Assistant.var_c == 2:
        heddd = ['Предмети:']
        list_t = Assistant.basic_inf[6]
        list_t = list_t.split("|")
        self.table_lable(list_t, heddd)

def text_colum(self, list_t_t):
    time2 = strftime('%d:%m:%Y')
    l_list = len(list_t_t) - 1
    h_list = len(list_t_t[0])
    if h_list == 4:
        time1 = list_t_t[-1]
        heddd = ["#", "Ім'я", time1, time2]
    else:
        heddd = ["#", "Ім'я", time2]
    return heddd

def take_obj(self):
    Assistant.value_r.append(Assistant.value[0])

    if Assistant.var_c == 2:
        Assistant.var_c = 0
        Assistant.save_l_n.append(["Кінець"])
        self.reload(1)
    else:
        Assistant.var_c = Assistant.var_c + 1
        Assistant.value_r = []

        Assistant.value_r.append(Assistant.value[0])
        self.reload(0)

def lit_colum(self, i):
    sheet_class = Assistant.work_book[Assistant.value_now[0]]
    dubl = [get_column_letter(5)]

    bol_i = False
    k = 6
    while True:
        if sheet_class.cell(row=i, column=k).value != None:
            if bol_i == True:
                dubl.append(get_column_letter(k))

```

```

        bol_i = False
        if sheet_class.cell(row=i, column=k).value == "A"
or sheet_class.cell(row=i, column=k).value == "S":
            dubl.append(get_column_letter(k - 1))
            bol_i = True
    else:
        if bol_i == True:
            break
        dubl.append(get_column_letter(k - 1))
        break
    k = k + 1

```

```

return dubl

```

```

def cr_ex_f(self, i, j, value_stud, dubl, str_p, a):
    sheet_class = Assistant.work_book[Assistant.value_now[0]]
    i = i + 2
    value_stud = value_stud + i
    bol_i = True
    while i < value_stud:
        if a == 1 :
            k = len(dubl) - 2
        else:
            k = 0
        str_f = str_p
        while k < len(dubl):
            dubl[k]
            str_f = str_f + dubl[k] + str(i)
            if bol_i == True:
                str_f = str_f + ':'
                bol_i = False
            else:
                str_f = str_f + ','
                bol_i = True
            k = k + 1
        if a == 2 :
            str_f = str_f[0:len(str_f) - 1] + ')'
            j = 2
        else:
            str_f = str_f[0:len(str_f) - 1] + '),0)'
        sheet_class.cell(row=i, column=j, value=str_f)
        i = i + 1

```

```

def create_or_update_Active(self, i, j, value_stud):
    class_name = Assistant.value_now[0]
    sbj_name = Assistant.value_now[1]

    sheet_class = Assistant.work_book[class_name]

    dubl = self.lit_colum(i)

    str_f = '=COUNT('

```

```

self.cr_ex_f(i, j, value_stud, dubl, str_f, 2)

self.seve_wb()

def change_work_book(self):
    self.reload(2)
    Assistant.var_c = 1
    list_f = os.listdir("data")
    heddd = ['WorkBook:']

    self.set_menu
    self.label_info_1 = tk.Label(Assistant.window,
font=("Times New Roman", 16), justify=tk.CENTER, text = "Оберіть
робочу книгу!\n Після підтвердження програма буде закрита
автоматично!")
    self.btn_1 = tk.Button(Assistant.window, text="Змінити
робочу книгу", font=("Times New Roman", 16),
command=lambda:self.activate_book())

    self.create_wg()

    self.label_info_1.place(relwidth=0.5, relheight=0.10,
relx=0.25, rely=0.24)
    self.btn_1.place(relheight=0.08, relx=0.4, rely=0.36)
    self.table_lable(list_f, heddd)
    self.time_now_d()

    def activate_book(self):
        l_name = len(Assistant.value) - 6
        if Assistant.value[0][l_name:] == '.xlsx':
            work_book =
load_workbook(f'./data/{Assistant.value[0]}')
            s_list = work_book.sheetnames
            sheet_tp = work_book['technical_page']
            sheet_tt_cn = work_book['time_table']
            sheet_tt_cn.column_dimensions['E'].width = 25
            base_inf = [Assistant.value[0], '0']

            sheet_row = len(list(sheet_tt_cn.rows))
            add_inf = ''
            i = 1

            while True:
                if sheet_tp.cell(row=1, column=i).value != None:
                    add_inf = add_inf + str(sheet_tp.cell(row=1,
column=i).value) + " "
                else:
                    base_inf.append(add_inf[:-1])
                    break
                i = i + 1

            add_inf = add_inf.split(" ")

```

```

list_wt = Assistant.basic_inf[3]
list_wt = list_wt.split(" ")

if sheet_tt_cn.cell(row=2, column=1).value == None:
    self.label_info_1.configure(text="Не коректно
заповнена книга!")
    return
else:
    if sheet_tt_cn.cell(row=2, column=1).value ==
list_wt[1]:
        i = 3
        while i <= sheet_row:
            if sheet_tt_cn.cell(row=i, column=1).value
== list_wt[0]:
                sheet_tp.cell(row=3, column=3,
value=str(i))
                break
                i = i + 1

            sheet_tp.cell(row=3, column=2, value=2)

            i = 5
            while i <= sheet_row:
                if sheet_tt_cn.cell(row=i, column=3).value == "#":
                    j = 1
                    while sheet_tt_cn.cell(row=i + j,
column=3).value != None:
                        time = int(sheet_tt_cn.cell(row=i + j,
column=3).value)
                        sheet_tt_cn.cell(row=i + j, column=2,
value=add_inf[time - 1])
                        j = j + 1
                    i = i + 1

                i = 2
                add_inf = ''

            while i < len(s_list):
                sheet_tt_cn = work_book[s_list[i]]
                sheet_tt_cn.column_dimensions['D'].width = 30
                add_inf = add_inf + s_list[i] + "|"
                j = 3
                while sheet_tt_cn.cell(row=j, column=3).value !=
None:
                    j = j + 1
                sheet_tp.cell(row=i + 2, column=1,
value=s_list[i])
                sheet_tp.cell(row=i + 2, column=2, value=j - 3)
                i = i + 1

            base_inf.append(Assistant.basic_inf[3])
            base_inf.append(Assistant.basic_inf[4])
            base_inf.append(add_inf[:-1])

```

```

add_inf = ''
i = 1
while True:
    if sheet_tp.cell(row=2, column=i).value != None:
        add_inf = add_inf + str(sheet_tp.cell(row=2,
column=i).value) + "|"
    else:
        base_inf.append(add_inf[:-1])
        break
    i = i + 1

work_book.save(f'./data/{Assistant.value[0]}')
write_base_inf(base_inf)
self.window.destroy()

def create_result(self, v):
    try:
        class_name = Assistant.value_now[0]
        sbj_name = Assistant.value_now[1]
    except IndexError :
        self.label_info_3.configure(text=f"Ви не обрали клас
чи предмет!")
        return None
    value_stud = self.get_value_stud(class_name)
    i_and_j = self.find_space_new_mark(class_name, sbj_name,
value_stud)
    i = i_and_j[0]
    j = i_and_j[1] + 1
    sheet_class = Assistant.work_book[class_name]
    text_a = ["AVERAGE all", "SUM all", "AVERAGE", "SUM"]
    sheet_class.cell(row=i + 1, column=j, value=text_a[v])
    if v == 0 or v == 2:
        text_a = 'A'
    else:
        text_a = 'S'
    sheet_class.cell(row=i , column=j, value=text_a)
    dubl = self.lit_colum(i)
    if v == 0:
        str_f = '=IFERROR(AVERAGE('
        a = 0
    elif v == 1:
        str_f = '=IFERROR(SUM('
        a = 0
    elif v == 2:
        str_f = '=IFERROR(AVERAGE('
        a = 1
    elif v == 3:
        str_f = '=IFERROR(SUM('
        a = 1
    self.cr_ex_f(i, j, value_stud, dubl, str_f, a)

```

```

text_a = ["Було підведено підсумок \nзаг. AVERAGE", "Було
підведено підсумок \nзаг. SUM", "Було підведено підсумок AVERAGE",
"Було підведено підсумок SUM"]
self.label_info_3.configure(text=text_a[v])
self.seve_wb()

def click(self, event):
    try:
        t = self.table.selection()
        if len(self.table.item(t) ["values"]) == 1:

            t2 = self.table.item(t) ["values"] [0]
            Assistant.value = [t2]
        else:
            t2 = self.table.item(t) ["values"] [0]
            t3 = self.table.item(t) ["values"] [1]
            Assistant.value = [t2, t3]
    except IndexError:
        None
    self.change_label_info()

def create_lesson(self):
    Assistant.var_c = 1
    Assistant.continue_t_ln = Assistant.save_l_n

    Assistant.save_l_n = ['Створення нового завдання']
    self.reload(0)

def end_leson(self):
    sheet_class = Assistant.work_book[Assistant.value_now[0]]
    value_stud = self.get_value_stud(Assistant.value_now[0])
    i_and_j = self.find_space_new_mark(Assistant.value_now[0],
Assistant.value_now[1], value_stud)
    i = i_and_j[0]
    j = i_and_j[1]
    sheet_class.cell(row=i, column=j, value="T")
    if sheet_class.cell(row=i+1, column=j).value == None:
        sheet_class.cell(row=i+1, column=j,
value=strftime('%d:%m:%Y'))
    self.create_or_update_Active(i , j, value_stud)
    Assistant.boll_1 = True
    if Assistant.continue_t_ln != []:
        Assistant.save_l_n = [Assistant.continue_t_ln[1],
Assistant.continue_t_ln[0]]
        Assistant.continue_t_ln = []
    if Assistant.save_l_n[1] == ["Кінець"]:
        Assistant.var_c = 1
        Assistant.save_l_n = ["Сьогодні були вже проведені усі
заняння!"]
        Assistant.boll_1 = False
        self.reload(0)

def open_file_c(self):

```

```
Assistant.work_book.save(f'./data/copy_Wbook.xlsx')
os.startfile("data\\copy_Wbook.xlsx")

def open_file(self, name_file):
    os.startfile("data\\" + name_file)

program_s = Assistant()
program_s.start_pr(0)
```

